Positional Control Strategies for a Modular, Long-Reach, Truss-Type Manipulator

by

Robert James Salerno

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

APPROVED:

C. F. Reinholtz, Chairman

S. G. Dhande | H. H. Robertshaw | R. L. West

M. P. Deisenroth | R. G. Leonard

November, 1993
Blacksburg, Virginia

Positional Control Strategies for a Modular, Long-Reach, Truss-Type Manipulator

by

Robert James Salerno

Committee Chairman: Charles F. Reinholtz
Virginia Polytechnic Institute and State University
Department of Mechanical Engineering
November 1993

Abstract

This dissertation proposes a new type of modular, long-reach, truss-type manipulator. Variable Geometry Trusses (VGT's) are used to construct a reconfigurable manipulator system in which all primary members are loaded in pure tension or compression. Each module of the manipulator system is either a static truss link or one of several possible VGT actuators. This results in an extremely stiff and strong manipulator system with minimal overall weight. While many potential applications exist for this technology, the present work was largely motivated by the need for a robotic waste remediation system for underground radioactive waste storage tanks. This new manipulator system provides several advantages when used for this application. The reconfigurable nature of the proposed system allows the manipulator to be adapted on site to unforeseen conditions. Additionally, the kinematic redundancy of the manipulator ensures that solutions can be accomplished even in a highly obstructed workspace. The parallel structure of the truss

modules enables the manipulator to be withdrawn in the event of a structural failure. Finally, of particular importance to this task, the open framework of the modules provide a passageway for waste conveyance or additionally, could act as a shielded conduit for control and power cabling.

Kinematic analysis algorithms tailored to address the peculiarities of this new manipulator system have also been developed. In this work, the kinematic redundancy of the system is exploited to provide alternative solutions, to avoid numerical difficulties at singularities, or to avoid workspace obstacles. These issues are addressed through a combination of null space optimization procedures and order reduction methods. The null space optimization procedures are accomplished by extracting information from a full singular value decomposition of the jacobian matrix. This method is shown to converge quickly, even for systems with thirty or more degrees of freedom. This represents a significant increase over most of the current literature which typically addresses systems of eight or fewer degrees of freedom.

This dissertation presents the first application of null space optimization techniques for the positional control of a high degree-of-freedom parallel manipulators. This work also formalizes the concept of a canonical input specification set. The application of this concept results in greatly simplified analyses of many parallel manipulators. Although the manipulator system discussed was specifically developed for robotic handling of radioactive waste, the final resulting methodology is suited to a much broader class of problems, namely, under-constrained, redundant manipulator systems in general.

Acknowledgements

I would like to express my sincere appreciation to the chairman of my advisory committee, Prof. Charles F. Reinholtz. I am grateful not only for his many helpful suggestions and insights, but also for his positive outlook, quick wit and valued friendship. Studying under Prof. Reinholtz has emphasized what a pleasure it is to work with individuals who are enthusiastic about, and enjoy, their profession.

I must also extend my appreciation to Prof. Harry H. Robertshaw, Prof. Sanjay G. Dhande, and my friends, Paul Tidwell and Babu Padmanabhan. All of these individuals contributed greatly to this work. Special recognition is due to Prof. Robertshaw for initially suggesting that I pursue this research topic.

I wish to thank the other members of my advisory committee, Prof. Robert West, Prof. Robert Leonard, and Prof. Michael Deisenroth for their many helpful comments. I consider myself very lucky for having had the oppurtunity to learn from these individuals.

I am deeply grateful to my parents, James and Elizabeth Salerno. Their support and encouragement throughout my life, have enabled me to accomplish what I have. I am especially grateful to my father, whose tremendous enthusiasm for science and engineering has profoundly influenced my views on education and the engineering profession.

Lastly, I must express my most heartfelt gratitude to my wife, Dory. Throughout my career as a graduate student, her support, understanding and encouragement never waivered.

Contents

List of Figures

Chapter 1

Introduction

This dissertation will focus on the development of positional control algorithms for a new type of modular, reconfigurable, long-reach manipulator. Although this type of manipulator is applicable to many tasks, it has been conceived with the initial intent of using it for radioactive waste removal from the Westinghouse Hanford Waste Storage Facility in Washington State. Figure 1.1 presents a typical manipulator of this type in the process of servicing a waste storage tank. With the exception of this one figure, all depictions of long-reach truss manipulators contained within this dissertation were generated as PostScript files from the algorithms discussed. This chapter will provide some background information on the Hanford Waste Storage Facility, and will briefly introduce the proposed manipulator system.

1.1 Motivation for the Development of the Manipulator System

The Hanford waste storage facility is the site of 149 underground, radioactive waste storage tanks. These single-shell-tanks were constructed from 1943 to 1964

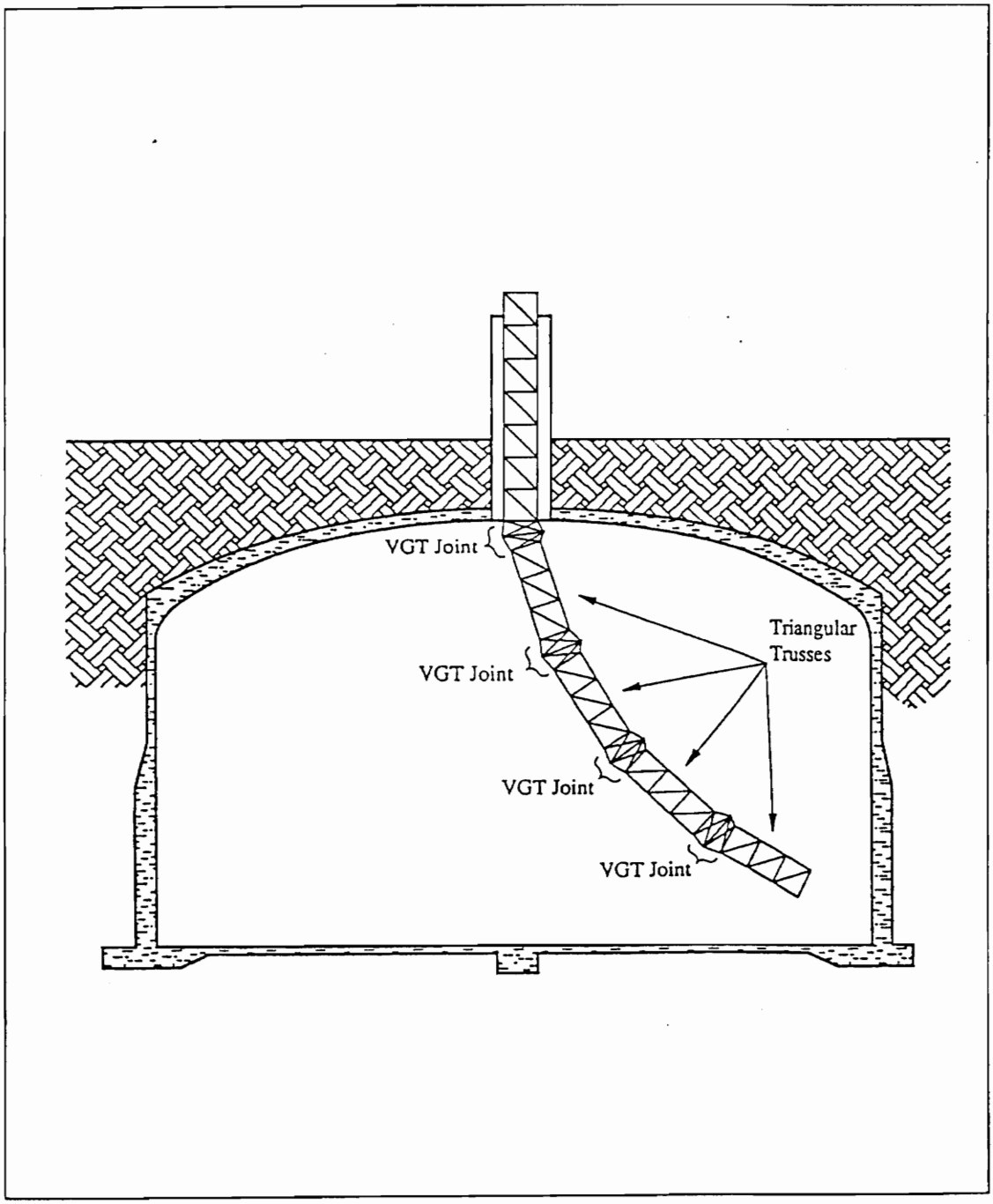


Figure 1.1: The Proposed Manipulator System

to facilitate long-term storage of chemicals and contaminated material generated during the production of plutonium for the nation's nuclear weapons program [24]. Because the integrity of these containment vessels is now seriously in question, it is imperative that the waste material be removed, processed, and interred in a more suitable manner. This task, characterized as Environmental Restoration and Waste Management, has been identified as a high priority by the United States Department of Energy (DOE) [66]. This research addresses only the issue of waste retrieval.

The scheduled clean-up operations of the Hanford waste tanks will require a rugged, reliable retrieval system that is sufficiently adaptive to adjust to the many types of waste materials and equipment present. The physical requirements for accomplishing the waste removal task are demanding. The manipulator must be constructed in a manner that provides for a long-reach (approximately 45 feet), while maintaining the positioning accuracy of a substantial payload(perhaps as large as 1000 lbs), with all access to the workspace being restricted to a small (42 inch diameter) portal [51]. These access portals are typical buried under 6-12 feet of earth [24]. Corporations, such as SPAR Aerospace Limited of Brampton, Ontario, have proposed the use of very large conventional manipulators to approach this problem [55]. To date, none of these conventional designs have proven successful.

The design is further complicated by the varying level of waste from tank to tank. Thus, in some tanks there may be sufficient room for the insertion of a complete manipulator system, while in others, this possibility does not exist.

1.2 The Proposed Manipulator System

The concept for this manipulator system was originally presented in a brief 1992 paper [51]. This document established manipulator performance requirements and

proposed a version of the manipulator system discussed here. At first glance, the proposed manipulator system shown in Fig. 1.1 appears similar to many conventional serial manipulators. It consists of relatively long, static links and compact, actuated joints. The major difference between the proposed system and others previously proposed is the extensive use of truss technology. In terms of mechanical design, the joints for such a manipulator present the single greatest obstacle to their construction. Clearly these joints will be subjected to enormous forces and torques while in service. Because of this, it is proposed that a new manipulator technology be utilized, namely implementing Variable-Geometry Trusses (VGT's) as joints. It will be shown later that hydraulically actuated VGT modules offer an ideal solution to the joint actuation problem posed by this massive manipulator. In addition to utilizing VGT's for the joints, the links will also be constructed from truss elements. A typical pair of active and static modules is shown in Fig. 1.2. Figures 1.3 and 1.4 present an electrically-actuated proto-type of a VGT module. This proto-type was constructed to demonstrate and verify the forward and inverse kinematic algorithms. Figure 1.5 presents a close-up view of the complex internal joint design necessary to ensure the proper relative motion between adjacent truss members. Note that this particular design has the electric motors mounted on the exterior of the truss structure. More streamlined designs, incorporating internally mounted motors are possible.

The following sections will introduce variable-geometry trusses in general and highlight some attributes of the proposed system.

1.2.1 Introduction to VGT Technology

Robotic manipulators in use today are generally serially connected devices which possess six or fewer degrees-of-freedom. As the name implies, pure serial manipulators are composed of a series of links connected to form an open chain. Each

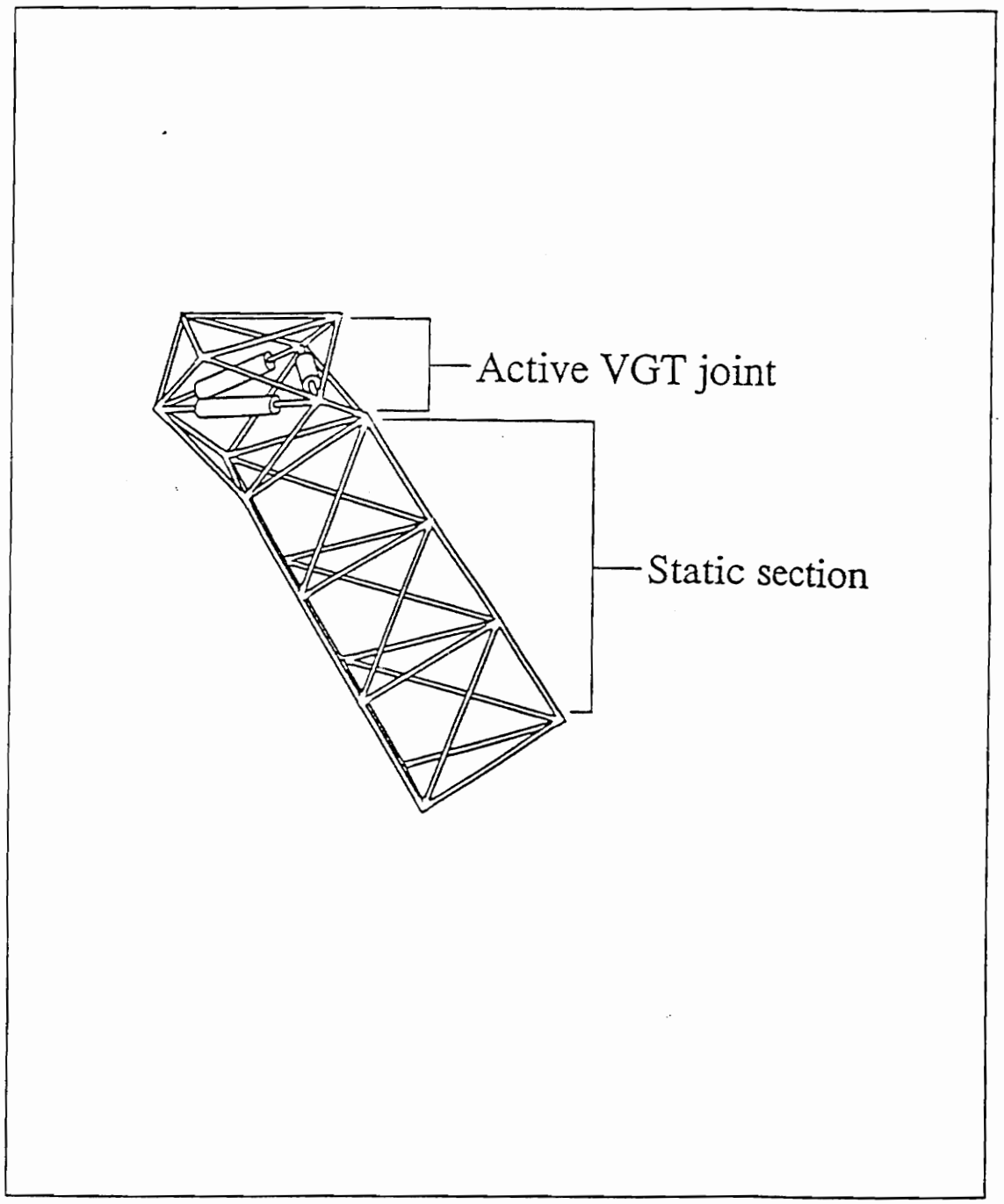


Figure 1.2: Typical Static and Active Truss Modules

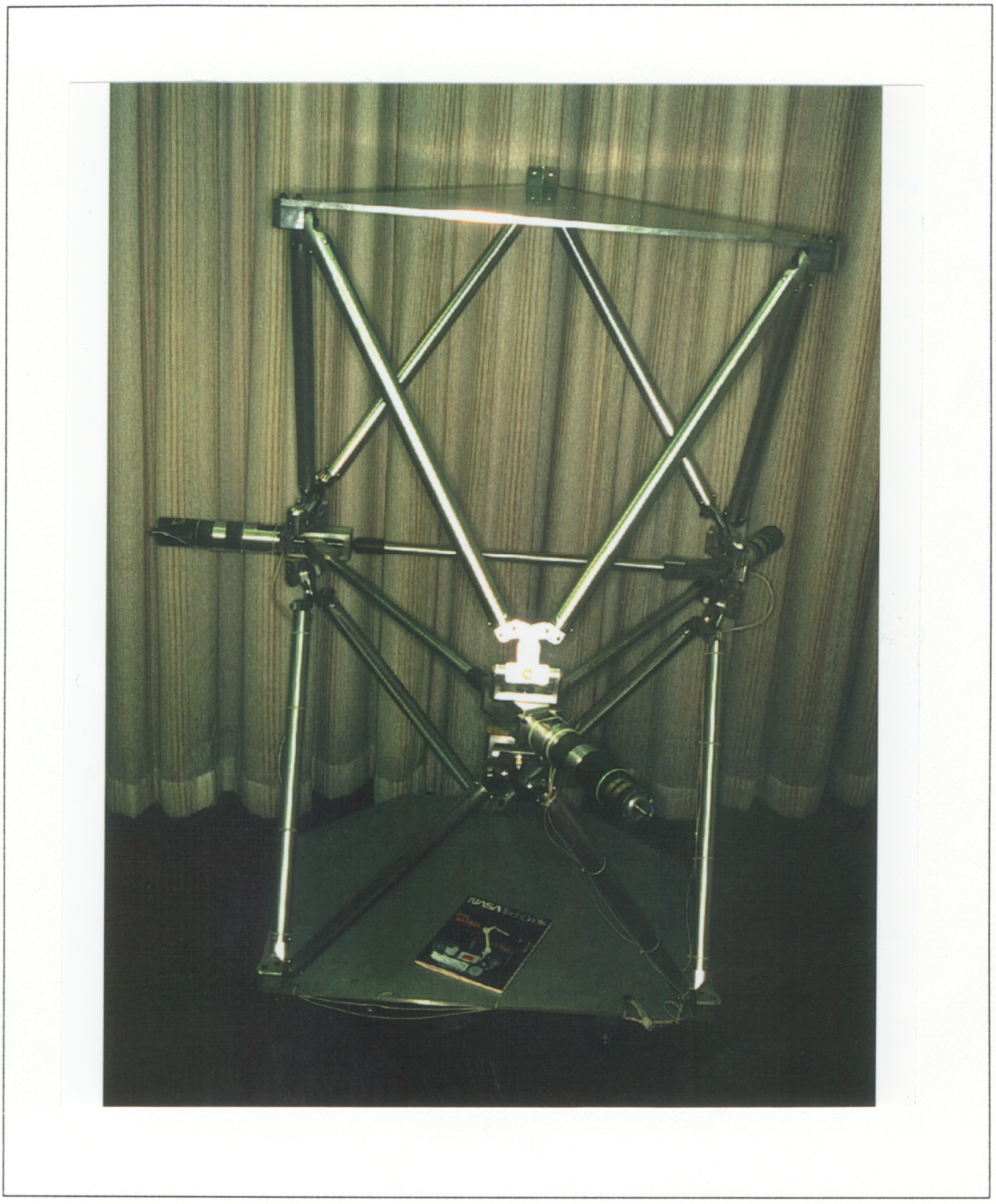


Figure 1.3: Proto-Type Variable Geometry Truss Module

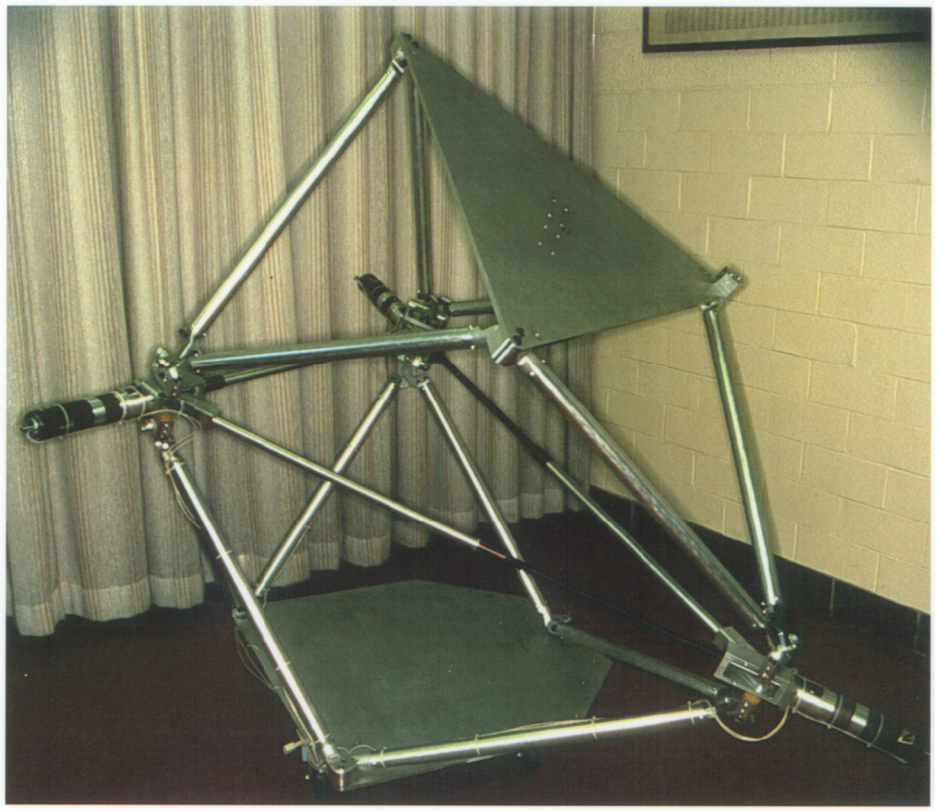


Figure 1.4: Proto-Type Variable Geometry Truss Module

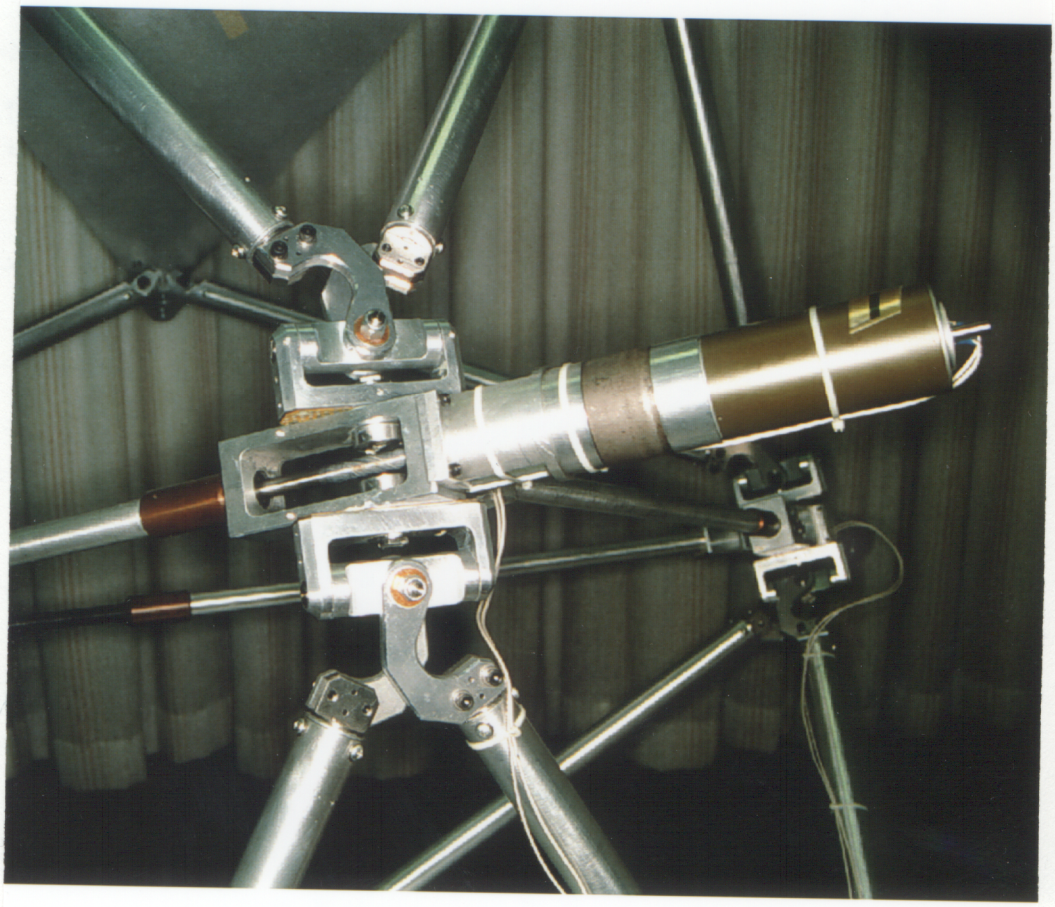


Figure 1.5: Internal Joint Close-Up

link of the manipulator is, in effect, a cantilever beam carrying the full load of all the links further out in the chain. The individual members of such a chain may be subjected to all possible types of loads; bending, torsional, axial, and shear. Of these possible loads, the design is usually limited by the effects of torsional and bending loads. Consequently, most purely serial manipulators are inherently compliant and have relatively poor load carrying capacities compared to their overall weight.

In contrast, parallel manipulators are composed of links connected to form one or several closed loops. This dissertation is concerned with a special subset of parallel manipulators referred to as Variable-Geometry Truss Manipulators. Simply stated, an ideal Variable-Geometry Truss (VGT) is a statically determinant truss which contains some number of variable length members. These extensible members allow the truss to change shape in a precise, controllable manner. Like a common static truss, if properly designed, the primary members of a VGT manipulator will be predominantly loaded as two-force (pure tension/compression) members. There are negligible bending loads imposed by the mass of the individual links themselves. This structural characteristic results in an extremely high stiffness-to-weight ratio, which enables some relatively light VGT manipulators to carry heavy payloads. These VGT modules are capable of supporting and manipulating several other modules, static or active, extended in a serial manner. Note that connecting these parallel structures in a serial manner imposes no bending or torsional loads on the individual truss members.

1.2.2 Other Benefits of Utilizing a Truss Structure

Most large-scale truss structures, are composed of a repeating pattern of basic truss cells. This inherent modular structure presents great advantages for the intended application. With this configuration, it is possible to change the geometry

of the manipulator by substituting different geometries of static or active modules, altering the order in which the modules are connected, or by changing the total number of modules present in the chain. In short, the manipulator may be reconfigured on site to adapt to unforeseen obstacles or other complications. Further, the modular structure also ensures that damaged or worn components may be quickly exchanged to avoid a prolonged maintenance downtime.

The truss structure is also advantageous, because, unlike conventional manipulators, it has an open framework which can be used for materials delivery, waste removal, or instrumentation and control cabling. Essentially, the exterior of the truss structure shields the more vulnerable components from abrasion and other hazards.

One of the more attractive features afforded by this open framework, is the possibility of designing an integrated waste conveyance system. There are two primary benefits to incorporating the waste conveyance system within the structure of the manipulator. First, this method would require only one penetration of the tank shell, thus preserving the structural integrity of the tank and minimizing the possibility of additional leakage. Secondly, the use of an auxiliary waste conveyance system, will always block a portion of the manipulator's workspace. To reach these blocked locations it will be necessary to relocate either the manipulator or the auxiliary conveyance system.

Figure 1.6 presents section views of two integrated waste conveyance concepts. The first is composed of a chain of screw conveyors coupled with universal joints. The screws can be designed such that they fit within a cylindrical shell and are supported there by fixed bearings. These cylindrical shells could be relatively thick, in which case they may also serve as the static modules, or they could be thin-walled and incorporated into the design of the static modules. A flexible boot is used to pass the waste through the active modules. As an alternative, a second,

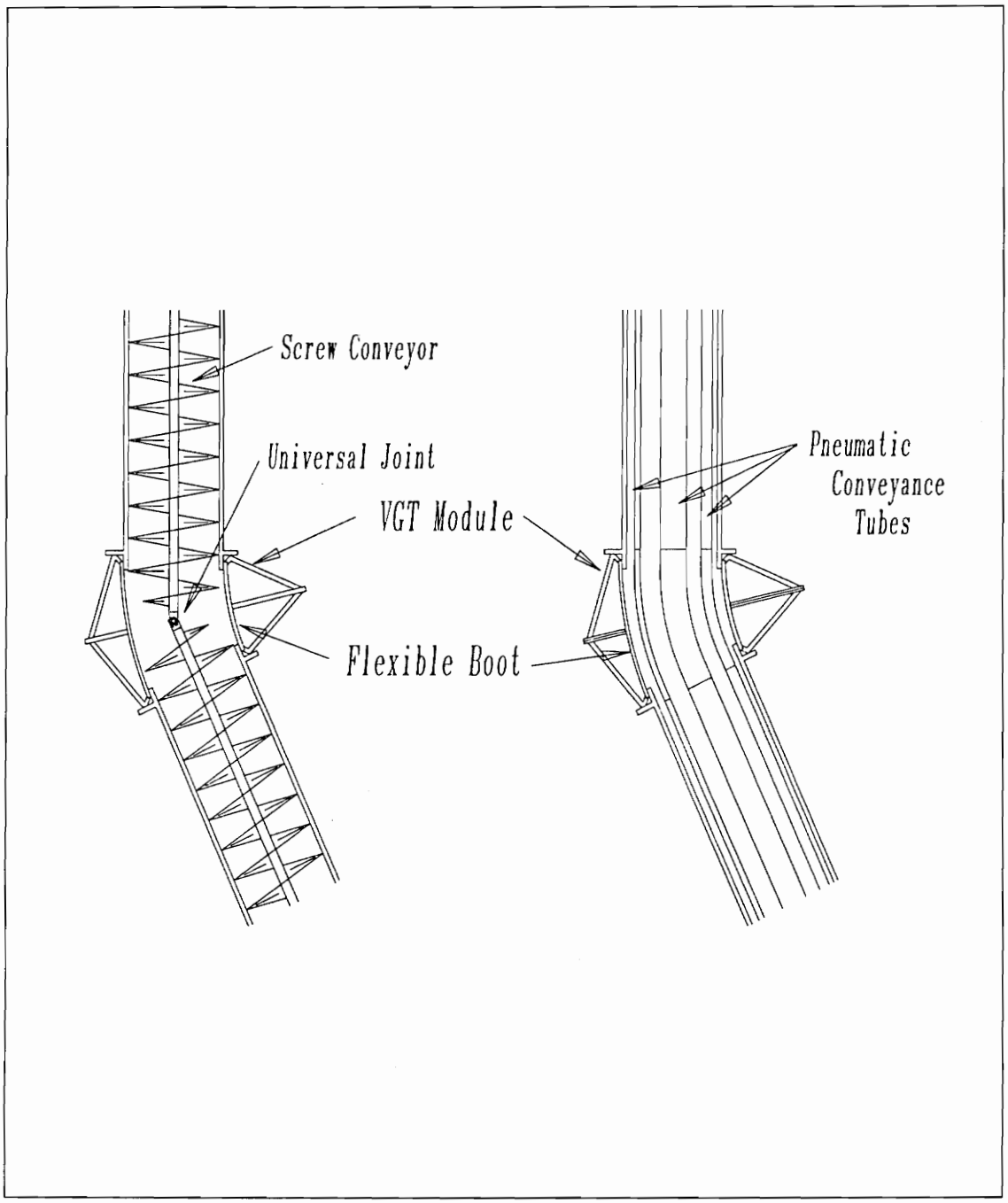


Figure 1.6: Two Possible Integrated Waste Conveyance Systems

a more conventional fluid conveyance system is also presented in Fig. 1.6.

Additionally, the parallel nature inherent in truss design ensures that the manipulator can be withdrawn in the event of a single structural failure.

1.2.3 Kinematic Complications

There are disadvantages associated with the use of the proposed manipulator. The positional control strategies, which are relatively simple for conventional manipulators, become much more complicated. Not only are they complicated by the kinematic properties of the VGT modules, but also by the very general, modular nature of the system. If the manipulator is to be truly reconfigurable, there must be provisions for solving the kinematics of a new configuration without prior knowledge of how it will be assembled. The remainder of this dissertation will focus on resolving these kinematic complications.

1.3 Outline of Contents

Chapter 2 will review existing literature and appropriate background information. In Chapter 3, kinematic analyses of the individual modules will be accomplished with emphasis on the information required for conducting similar analyses on multi-module systems. Efficient methods for analyzing the forward kinematics of the multi-module systems will be presented in Chapter 4. Chapter 5 will introduce some of the complications involved in solving the inverse positioning problem of a general modular manipulator. Also, the basic philosophy behind two different approaches to the inverse kinematic analysis will be briefly presented. The two methods discussed are null space optimization techniques and order reduction methods. The two subsequent chapters will discuss each of these analysis techniques in greater detail, highlighting both their advantages and limitations.

Several simplified examples will be presented along with the explanation of these methods. Chapter 8 will present some realistic examples of these techniques being used to solve the inverse positioning problem for the modular manipulator. Finally, conclusions will be presented in Chapter 9 along with recommendations for future research concerning the development of this manipulator system.

Chapter 2

Review of Literature

The material presented within this dissertation relies heavily on the fundamentals of kinematics, robotics, numerical analysis and geometric modeling. This literature review will not contain references to texts which contain these fundamental concepts. Instead, when appropriate, these texts will be referenced within the the body of the dissertation. This literature review will present publications which represent the past and current research on parallel, redundant and long-reach manipulators. To facilitate the organization of this material, the literature review will be divided into three sections; kinematics of parallel manipulators, variable geometry truss research, and redundant and long-reach manipulator research.

2.1 Kinematics of Parallel Manipulators

Much of the early research conducted in robotics centered around the use of serially-connected manipulators. No significant attention was given parallel manipulators until 1965, when Stewart [56] proposed the idea of a platform type manipulator. This initial proposal showed that very rigid six degree-of-freedom manipulators could be developed, but unfortunately, they had very restricted workspaces. Other than their practical implementation as flight and vehicle simulators, these devices

remained somewhat of a novelty for many years. Perhaps one reason for this was the lack of adequate computing facilities to quickly solve the iterative kinematic constraint equations. The next decade witnessed tremendous advances in the field of robotics, but practically all of the research was concerned solely with serial manipulators.

Further advances in parallel manipulators were put forth by Tesar and Cox [61] in 1981, Hunt [19] in 1983, Yang and Lee [65] in 1984, and Fichter [15] in 1985. Tesar and Cox performed the kinematic and dynamic analysis of a three degree-of-freedom parallel manipulator. Hunt was concerned with the general analysis techniques for approaching the kinematics of parallel devices. Yang and Lee conducted a feasibility study on several geometrical variations of Stewart platforms. Fichter concentrated on the analysis and design of a Stewart platform based manipulator.

Research concerned with the dynamic analysis of parallel manipulators has been conducted by Sklar and Tesar [54] and Lee and Chao [25]. The work by Sklar and Tesar was concerned with the dynamic analysis techniques for hybrid serial/parallel manipulators. This included several useful geometries for industrial manipulators. Lee and Chao focused on the kinematic and dynamic analysis of one particular geometry of spatial three degree-of-freedom manipulator. Sugimoto [58, 59] has developed computational methods for the dynamic analysis of many types of parallel manipulators.

The papers cited above are all concerned with the analysis of parallel manipulators. Although, none of these papers address the concept of a truss manipulator, many of the analysis techniques presented could be applied to Variable Geometry Trusses.

2.2 Variable Geometry Truss Research

Simple VGT's are actually quite common and have been employed in engineering applications for decades. Many common configurations of construction cranes, draw bridges, and similar devices can be classified as VGT's. However, most of these common applications utilize a VGT element as a single degree-of-freedom mechanism. Stewart's early work, although technically not a truss structure, illustrated that it was possible to construct a fully-parallel manipulator capable of positioning and orienting a platform in three dimensional space.

In the early eighties the National Aeronautics and Space Administration (NASA) became interested in the development of "deployable" space structures [14]. NASA had many applications in space which required very large, stiff structures. A natural choice for such structures is a static truss. However, because of transport problems, these large structures must either be assembled in space or be transported in a compact form for later automatic deployment. Typically, these deployable structures become static once locked into their extended position.

While investigating possible geometries for these deployable structures, Rhodes and Mikulas [46] discovered that one certain geometry of deployable truss (a double-octahedral) had properties that made it suitable to act as a three degree-of-freedom spatial manipulator. This was the first true VGT manipulator concept. Refinements were later made by Rhodes and Mikulas in conjunction with Sincarsin [52] that lead to the development of a working proof-of-concept model. This model not only solved many of the complex joint geometry problems, but also successfully demonstrated the potential usefulness of these devices. This insight made possible the work of Miura and Furuya [31], and Miura, et al. [30], who analyzed the kinematics of a double-octahedral VGT. This early analysis work was later extended by Gokhale and Reinholtz [45], and Padmanabhan, et al. [41]. The re-

search of Padmanabhan is particularly interesting in that it identified several tasks for which a symmetric double-octahedral truss was well suited. This work then presents closed-form solutions to these tasks. Further refinements of the physical manipulator design were achieved by Tidwell, et al. in 1990 [62, 63]. This work included the development of design-curves that enabled quick evaluation of the potential motion of a double octahedral VGT.

Simultaneous with the development of the double-octahedral truss model, Sincarsin and Hughes [53] also explored the characteristics of four other candidate geometries. Their evaluations concluded that the double-octahedral geometry was indeed the most favorable geometry. Of primary interest in this study was the issue of collapsibility. Therefore, it should not be assumed that this is the best geometry for all applications. Jain and Kramer [20] also investigated another possible geometry and completed the design of a tetrahedral/tetrahedral VGT.

Other research concerning the use of a VGT cell as a replacement for more conventional devices has been conducted by Nayfeh [36], Clark and Robertshaw [10] and Wynn and Robertshaw [64]. Nayfeh investigated the kinematics of a foldable, revolute jointed space crane composed of several, essentially planar, VGT cells. The analysis undertaken was limited to only one of the proposed cells. Clark investigated the use of these VGT modules for actively damping vibration in large truss structures. This study dramatically illustrates the superiority of VGT actuators over conventional proof-mass type actuators for vibration control. This work was extended to the control of vibrations in spatial structures by Wynn in 1990. This last work included an impressive experimental demonstration of vibrations being actively controlled by a spatial VGT manipulator.

Natori, Iwasaki and Kuwao [35] have also investigated the vibration characteristics of long-chain planar trusses.

2.3 Redundant and Long-Reach Manipulator Research

Prior to 1988, interest in the concept of using VGT's for robotic manipulation remained somewhat limited. One primary reason for this lack of interest was the fact that these devices possessed relatively limited workspaces. This drawback can be overcome by joining several VGT's together end-to-end. This produces a long-chain manipulator whose workspace can be quite substantial . Miura, Furuya, and Suzuki [30, 31] were among the first to propose this long-chain structure in 1985. However, their solutions were limited by the assumption that all cells mimic the same motion. For their selected double-octahedral geometry, this constrained the motion of the manipulator to only three degrees-of-freedom regardless of the number of actuators present. This illustrates perhaps the second primary reason for the lack of interest in VGT manipulators. In short, these long-chain manipulators posed problems that were unheard of from the standpoint of conventional robotics. Most conventional manipulators are limited to six degrees-of-freedom. Practical long-chain truss manipulators require from nine to sixty degrees-of-freedom. No algorithms had yet been developed to satisfactorily solve the kinematics of these under-constrained systems. Shape control concepts provided a solution to this problem. These methods were introduced for planar VGT manipulators by Salerno and Reinholtz [49] in 1988 and later refined to include an n degree-of-freedom spatial manipulator system in 1989 [48, 50]. A simultaneous research effort, which yielded similar results was accomplished by Chirikijan and Burdick [8, 9]. Naccarato and Hughes [33] have also developed similar algorithms. Of particular interest concerning this present application is the work done by Padmanabhan in 1989 [40]. This research concerned the design of a VGT-jointed, four degree-of-freedom, planar manipulator. Here, the VGT joints were substituted for

simple revolute joints. In many ways this work closely parallels the proposed waste removal manipulator.

The concept of using curves or surfaces to determine the shape of a structure was briefly addressed by Natori, Iwasaki and Kuwao [35]. In their research, shape control concepts were utilized to yield an "adaptive planar truss structure" capable of forming a variety of parabolic shapes for large space antennae.

In recent years, much attention has been focused on redundant manipulators. As industrial manipulators find applications beyond the structured environment of the factory floor, redundancy provides these machines with the ability to adapt to unforeseen conditions. Many researchers [39, 22, 23] have sought methods which "resolve" the redundancy. Typically these approaches sought to utilize the extra degrees-of-freedom to control parameters other than the end-effector position. This is accomplished by adding constraints to the system until it is no longer under-constrained. Often, solutions can then be found in closed-form. An example of this is specifying not only an end-effector position but also an elbow position to ensure obstacle avoidance. Other approaches introduced kinetic constraints in addition to geometric constraints. Limiting or specifying characteristics of the joint velocities is an example of this method. Although these methods are appropriate to many situations, they are eliminating many possible solutions in order to achieve fast processing times.

Another approach to resolving kinematic redundancy is to utilize the redundant degrees-of-freedom to optimize some objective function. This is the approach used by Arnautovic [1], Mayorga and Wong [29], Suh and Hollerbach [60], and Carignan [6]. In this case, the necessary conditions of the optimization procedure provide the required constraints to identify a unique solution. The challenge to this task, is identifying the optimum solution quickly. The techniques presented in this dissertation are an extension of these optimization concepts.

Other related works of interest are Colbaugh and Jamshidi [11], Kelmar and Khosla [22], and Blume, et al. [5]. Colbaugh and Jamshidi specifically addressed the use of robotic manipulators for hazardous waste handling. Kelmar and Khosla proposed a fully serial, reconfigurable, modular manipulator system. Blume presented research on another large-scale manipulator system. This manipulator has a 22 meter reach and a 1.4 metric ton payload.

Chapter 3

Kinematics of the Individual Truss Modules

To obtain a more detailed understanding of the complex motion characteristics associated with the long-reach VGT manipulators, it is first necessary to gain a familiarity with the fundamental elements used in their construction. The science of kinematics provides a methodical way of analyzing the position, velocity, acceleration, and all higher-order derivatives of motion of these complex truss configurations. One definition of kinematics that is particularly appropriate to the study of VGT manipulators is as follows: *kinematics is the study of constrained motion of interconnected rigid links.* In this definition the term *"rigid"* is used to describe the non-elastic behavior of the links and does not preclude the use of intentionally extensible links.

From a robotics viewpoint there are two distinct types of kinematic analyses, namely, forward kinematic analysis and inverse kinematic analysis. Forward kinematic analysis, sometimes referred to as direct kinematic analysis, is concerned with finding the position and orientation of any or all members of a device given only the geometric constraints of the individual links, the order and manner in which these links are assembled, and a set values for the control variables (typ-

ically either joint angles or extensible link lengths). Forward kinematic analysis could also be conducted to find the linear and angular velocities or accelerations of any member. This is often referred to as forward rate analysis as opposed to forward position analysis.

Inverse kinematic analysis of a given device is concerned with finding a set control variable values that yields a desired position and/or orientation of a set of members. As a practical matter, in most useful devices, it is not possible to control the position and orientation of all members simultaneously. Again, the inverse rate analysis may be solved for any higher order derivative of motion given a specified control variable input.

In the following sections, these concepts will be further developed for several different geometries of active truss modules. A comprehensive kinematic description of the static truss modules will also be developed.

3.1 Truss Terminology and Notational Convention

Many geometric configurations, both planar and spatial, are possible candidates for use as a VGT manipulator. This section will enumerate some of the possible geometries, introduce a convention for naming the links of a truss, and describe the notational conventions that will be used for analysis throughout this dissertation.

The geometric structure on which all statically determinant planar trusses are based is the triangle. The triangle can therefore be thought of as the basic unit structure, or cell, used to construct all planar trusses. Spatial trusses present several possible cell structures that could be used to form a statically determinant truss. Four simple geometries are shown in Fig. 3.1. The tetrahedron is the simplest geometry. More complicated geometries can be formed by connecting all vertices

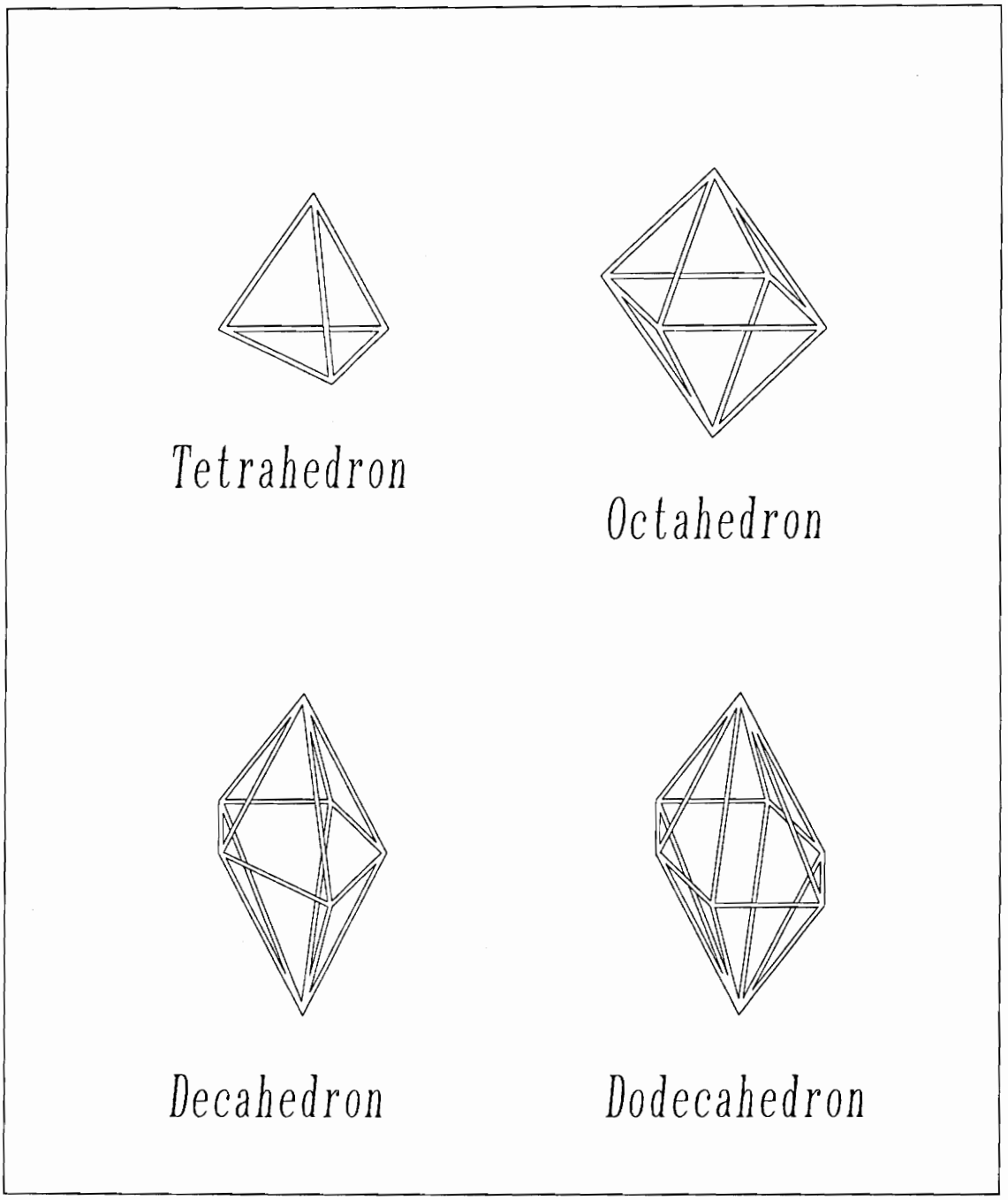


Figure 3.1: Four Simple Spatial Unit Cells

of a planar polygon to a pair of out-of-plane points as illustrated in Fig. 3.1. This type of geometric construction is referred to as di-pyramidal. Large-scale, statically determinant truss structures are commonly constructed with a replicating pattern of these basic unit cells.

A common static truss is shown in Fig. 3.2. This truss is composed of three types of members. Those members which are oriented transverse to the truss axis are called battens. Those members which lie parallel to the truss axis are referred to as longerons. The remaining members, which lie neither parallel nor perpendicular to the truss axis, are identified as cross-longerons. In applying this nomenclature to variable-geometry trusses, it is often difficult, if not impossible, to distinguish between longerons and cross-longerons. For some geometries, a single member could fit both descriptions depending on the length of the extensible members. For this reason, this dissertation will refer to both groups simply as longerons. The points at which the truss members are joined will be called nodes.

To simplify notation, vectors expressed in a frame $\mathcal{F}$ will be identified by a leading-superscript $\mathcal{F}$ (for example, ${}^{\mathcal{F}}\vec{V}$). The length of fixed length members will always be denoted by upper case L's (for example L_3 or L_S). The length of all extensible members will be identified by lower case l's (for example l_3 or l_s). Some analyses may contain both L_3 and l_3. These lengths are not, in general, equal. Also, all position vectors will be described using spatial homogeneous coordinates and thus will be represented with 4×1 matrices.

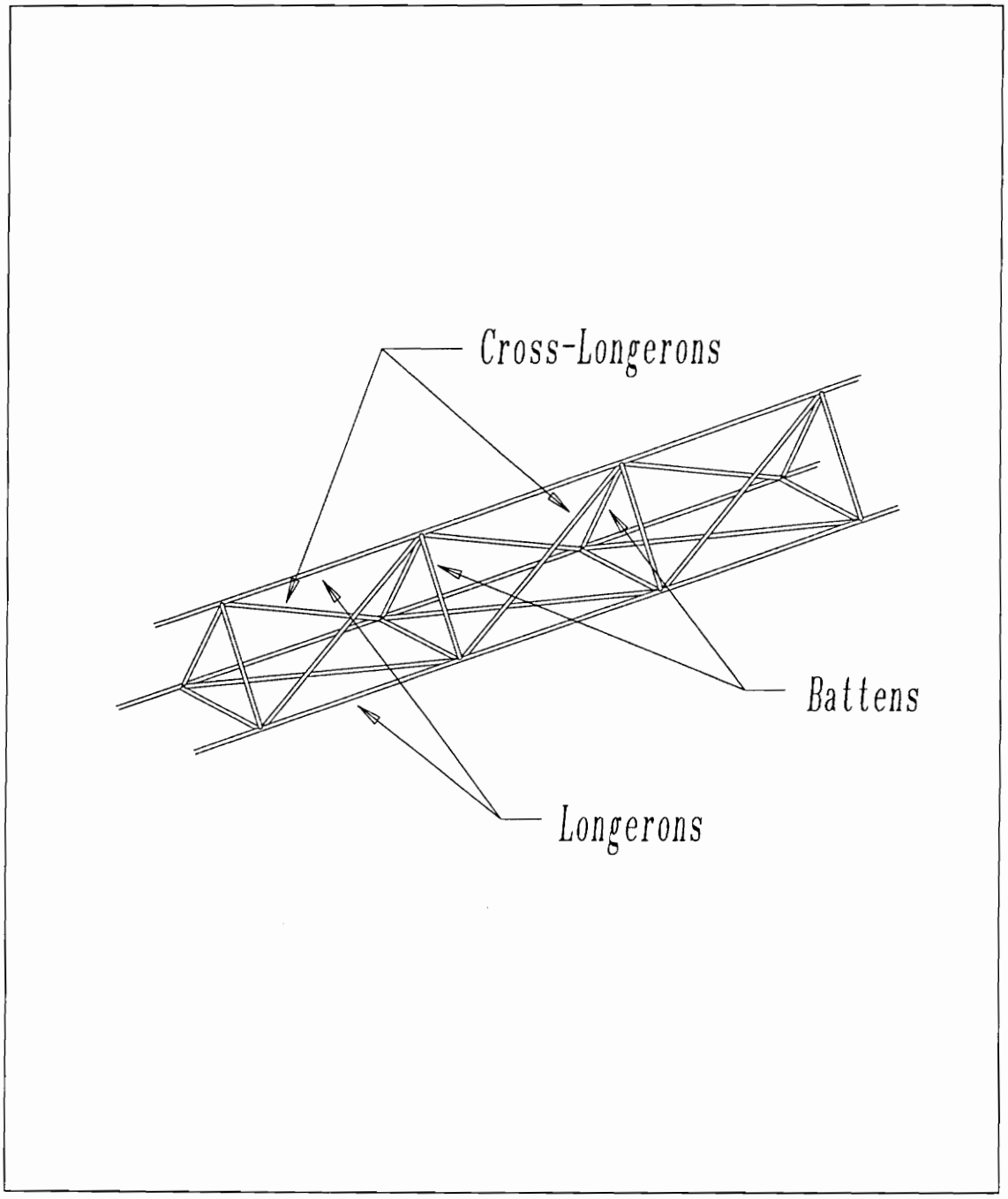


Figure 3.2: A Typical Spatial Truss

3.2 Conventional vs. Canonical Input Specifications for the Forward Kinematic Analysis

The goal of kinematic analysis, in general, is to develop motion input/output relationships for the devices studied. In the case of forward kinematics, to determine the system output given the system input. In terms of conventional forward kinematic analysis, the *input* is almost always interpreted as those parameters which are directly controlled. In other words, the same parameters that are adjusted by the physical actuators and measured by the instrumentation system. For most manipulators these inputs are either angular motions of a revolute joint or linear displacements of a prismatic joint. These seem to be natural parameters to specify as the input to the forward kinematics. These are, however, not the only possible input parameters. Some input specifications will be more convenient than others. The following simple examples will show that for some parallel manipulators these conventional input specifications may not necessarily be the most appropriate.

First, it must be understood that any set of independent parameters could be considered an input specification set, provided that the chosen parameters, when fixed, completely define the positional *state* of the manipulator. Meaning that no further motion of the manipulator is possible. This does not mean that only one possible position of the manipulator exists for the given values. It is possible that other branches may satisfy the specified values. Naturally, an n degree-of-freedom device will require that n parameters be specified.

As an example, consider the simple planar two link serial manipulator shown in Fig. 3.3-A. It is easy to see how the two joint angles θ_1 and θ_2 could be used to control the XY position of point P. The conventional input specification would be θ_1 and θ_2. If θ_1 and θ_2 are fixed, no motion is possible. Also note that, the set of θ_1

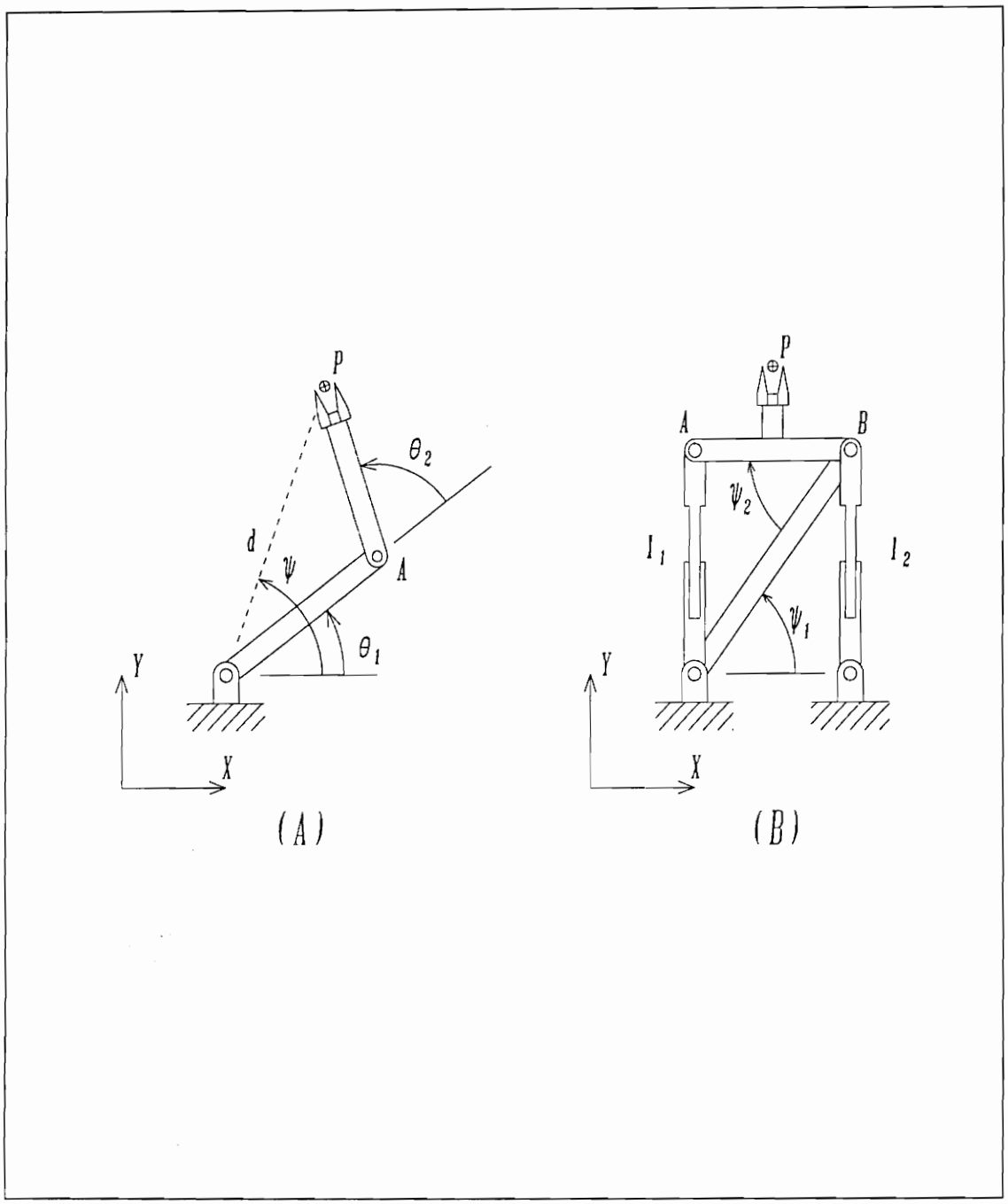


Figure 3.3: Possible input specifications for a serial and parallel manipulator

and θ_2 has the property that it uniquely defines the position of the manipulator. This uniqueness is desirable, since it dictates that the solution will be closed-form and linear. Other valid specification sets are, d and ψ, or the X coordinate of point A and θ_2. Although valid, neither is desirable because they do not uniquely define the position of the manipulator. Both result in two solutions to the forward kinematics. These solutions are depicted in Fig. 3.4. Since no single unique solution results, the forward solution will be non-linear.

For most serial manipulators the conventional input specification set is the simplest, or canonical, input specification set.

Now, contrast these results with those of the parallel manipulator shown in Fig. 3.3-B. It is easy to verify that the XY position of point P can be controlled, within a limited workspace, by changing l_1 and l_2. If the conventional input parameters, l_1 and l_2, are utilized, there are four solutions to the forward kinematics. These are illustrated in Fig. 3.5. For this parallel manipulator, the conventional input specification set results in a non-linear forward solution. Consider the following canonical input parameter set. If it is assumed that the input to the system is ψ_1 and ψ_2, then a unique closed-form solution to the forward kinematics problem is obtained. If it ever is required to iteratively solve the forward kinematics in search of a particular result, it is logical to chose the parameter specification set that results in linear solutions. Once the solution is found it is easy to calculate the corresponding unique values of l_1 and l_2.

Note that, if the conventional input specification set is used, the partial derivatives of the outputs with respect to the inputs ($\frac{\partial P_x}{\partial l_1}$, $\frac{\partial P_x}{\partial l_2}$, $\frac{\partial P_y}{\partial l_1}$, and $\frac{\partial P_y}{\partial l_2}$) cannot be expressed in closed form. The output parameters are not differentiable with respect to the inputs because of the non-linearities present in this forward problem. However, it is possible to use the alternate input specification set to find $\frac{\partial P_x}{\partial \psi_1}$, $\frac{\partial P_x}{\partial \psi_2}$, $\frac{\partial P_y}{\partial \psi_1}$, and $\frac{\partial P_y}{\partial \psi_2}$ in closed form. Similarly, $\frac{\partial \psi_1}{\partial l_1}$, $\frac{\partial \psi_2}{\partial l_2}$, $\frac{\partial \psi_1}{\partial l_1}$, and $\frac{\partial \psi_2}{\partial l_2}$ may also

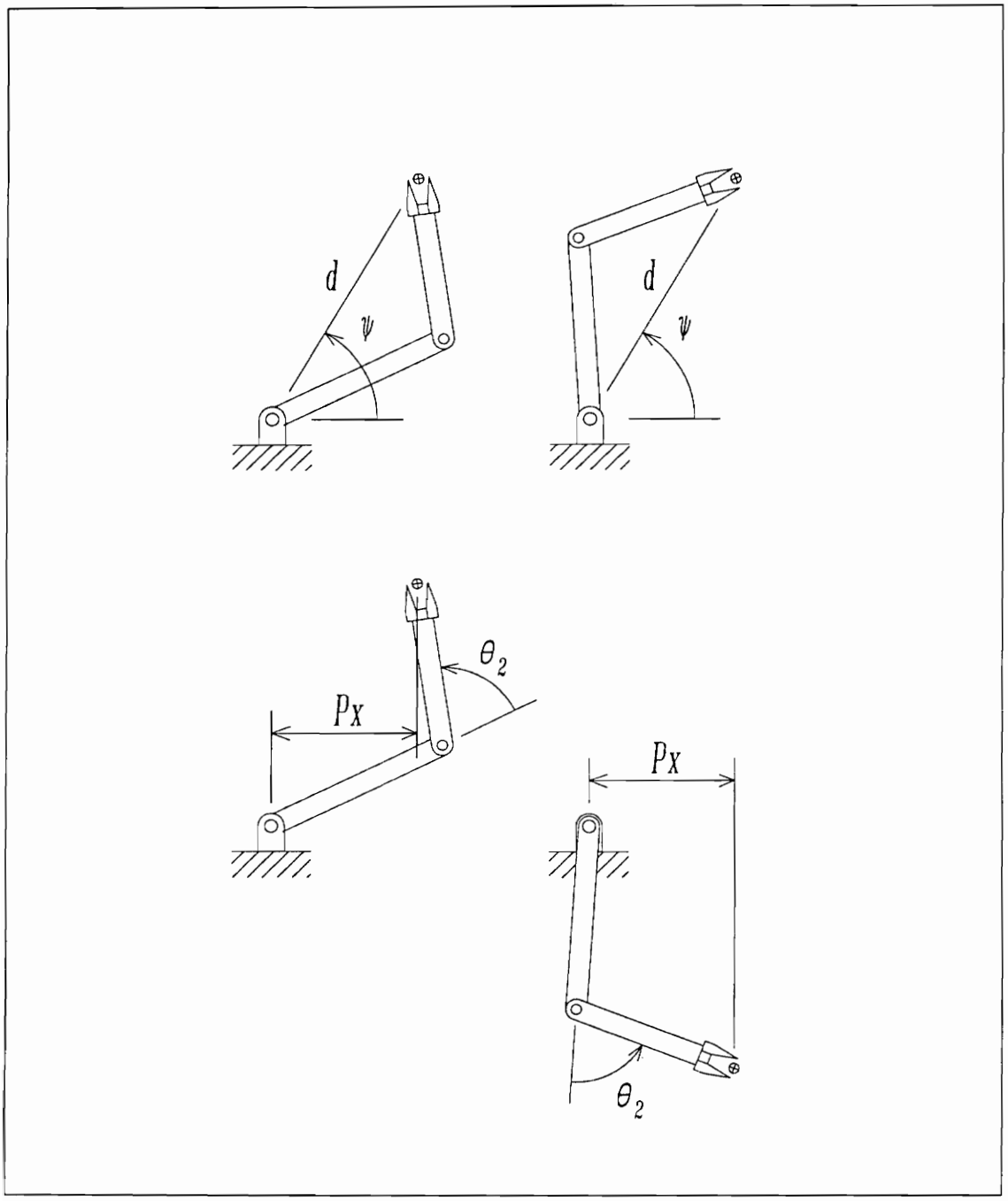


Figure 3.4: Solutions for Alternative Input Specification Sets

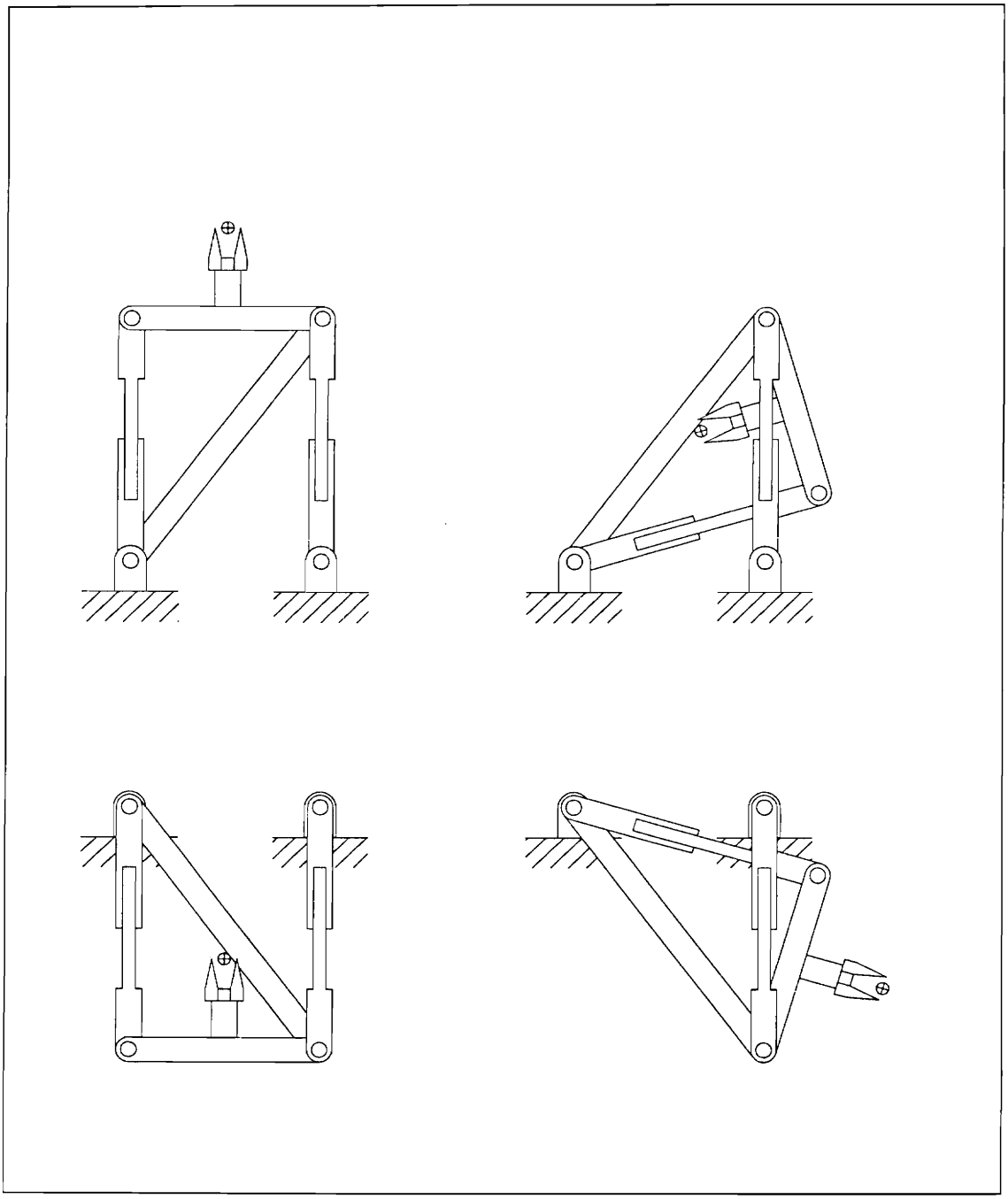


Figure 3.5: Four Solutions to the Conventional Forward Kinematic Analysis of a Simple Parallel Manipulator.

be expressed in closed-form. It is now trivial to use the chain rule to compute the partial derivatives of goal positions with respect to changes in link lengths in closed form.

Although the method of utilizing a canonical input specification will not be necessary for all of the parallel geometries, it will be employed as part of the solution method for the inverse solution of the double-octahedral geometries.

3.3 Analysis of a Tetrahedral Module

The tetrahedral unit cell of Fig. 3.1 represents the simplest possible spatial truss configuration. The basic unit cell consists of four non-coplanar nodes, each of which is joined to all other nodes by a side member. Thus, a tetrahedron has four triangular facets. Each facet shares one side member with all other facets. Imposing the restriction that all active modules must have two static triangular faces (this is necessary to provide an attachment surface) leaves only one remaining side member as a candidate for actuation. This essentially produces a single degree-of-freedom "hinge" or revolute actuator. The chosen geometry for the modular manipulator is illustrated in Fig. 3.6. It consists of five static length members and one variable length member joined together with spheric joints at the four non-coplanar nodes, n_1,n_2,n_3, and n_4. For this work, it is assumed that all of the static length members except $n_2 - n_3$ are of length L_1, $n_2 - n_3$ is of length L_2, and the length of the actuated member will be denoted by l. This configuration was chosen to produce isosceles fixed triangles to which static trusses, with the same end condition, may be attached. A more general case, with scalene triangles could be analyzed, however, this geometry tends to produce forces which shear the two static trusses in opposite directions. This occurs anytime the line between n_1 and n_4 is not perpendicular to the hinge line $n_2 - n_3$. This shearing action is

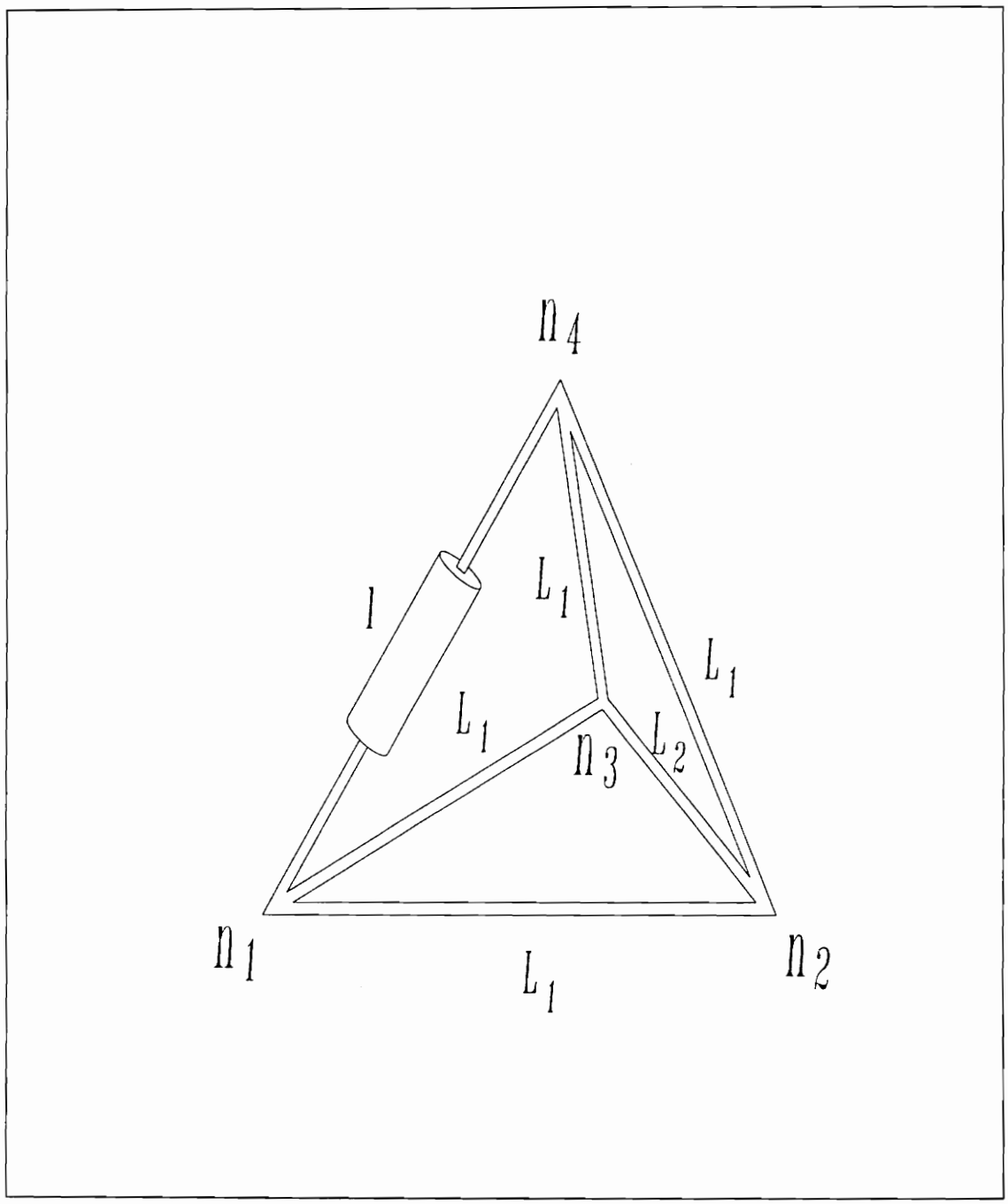


Figure 3.6: Active Tetrahedral Truss Module

detrimental to the specified objective of the manipulator system.

A configuration with equilateral triangles is addressed by the present analysis if $L_1 = L_2$.

3.3.1 Forward and Inverse Kinematic Analysis

The forward kinematic problem for the tetrahedral truss can be stated as follows: *given the fixed lengths, L_1 and L_2, and the input length, l, find the angle θ that describes the relative rotation between the two fixed triangles.*

Likewise, the corresponding inverse kinematic problem may be summarized as: *given the desired rotation between the two fixed triangles, find the extensible link length l that produces this rotation.*

Figure 3.7 schematically illustrates an active tetrahedral truss module with the parameters needed to compute the forward and inverse kinematics. First, the height, or altitude, of the static triangles must be calculated as follows:

$$h = \sqrt{L_1^2 - \frac{L_2^2}{4}} \tag{3.1}$$

Now the angle θ can be found from the law of cosines as:

$$\theta = \arccos(1 - \frac{l^2}{2h^2}) = \arccos(1 - \frac{l^2}{2L_1^2 - \frac{1}{2}L_2^2}) \tag{3.2}$$

Rearranging this same equation yields a solution to the inverse kinematic problem.

$$l = \sqrt{2h^2(1 - \cos\theta)} = \sqrt{(2L_1^2 - \frac{1}{2}L_2^2)(1 - \cos\theta)} \tag{3.3}$$

Note that this single degree-of-freedom device has both a closed form forward and inverse solution. The forward solution is non-linear, and hence results in two

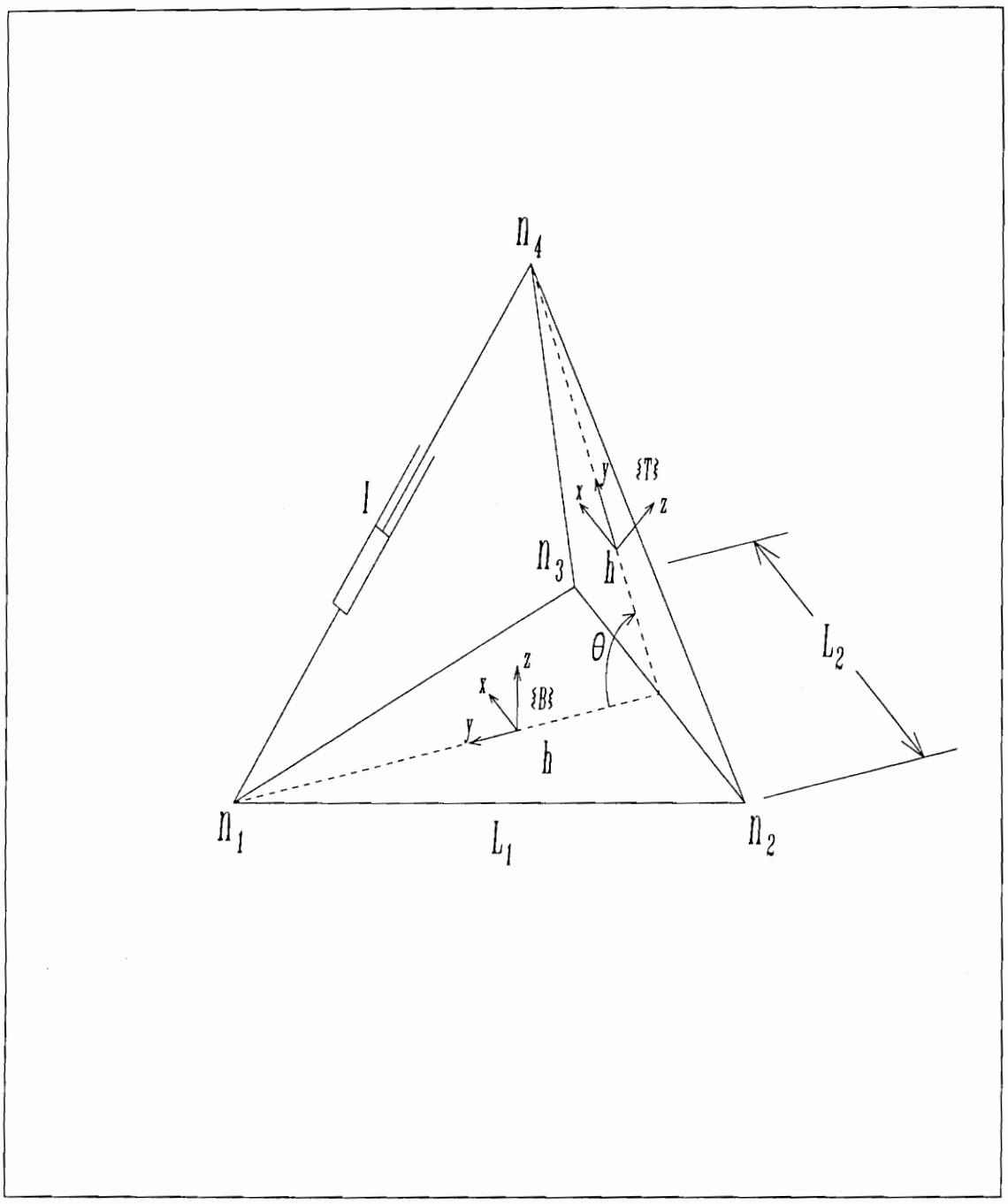


Figure 3.7: Schematic Tetrahedral Module with Analysis Parameters

possible solutions ($\pm\theta$). The inverse solution is linear and therefore results in a unique solution.

To efficiently utilize this result in solving a multi-module truss, requires that the input/output relationship of all geometries be specified in the same manner. Although this last step appears to unnecessarily complicate the tetrahedral solution, it is fundamental to the development of a general multi-module kinematics algorithm. To establish this input/output relationship, a base coordinate frame, $\mathcal{B}$, will always be attached at the centroid of one fixed triangle. The X and Y axes of frame $\mathcal{B}$ are constrained to lie within the plane defined by $\Delta n_1 n_2 n_3$, with the X axis parallel to $n_2 n_3$. A second coordinate frame, $\mathcal{T}$, is attached in a similar manner to $\Delta n_1 n_2 n_4$. The positional relationship between these two frames is completely described by a homogeneous transform, ${}^{\mathcal{B}}_{\mathcal{T}}T$. For this particular geometry, this is only a function of θ, and may be written as:

$$ {}^{\mathcal{B}}_{\mathcal{T}}T(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & -\frac{h}{3}(1-\cos\theta) \\ 0 & \sin\theta & \cos\theta & \frac{h}{3}\sin\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4} $$

3.4 Analysis of a Longeron-Actuated, Octahedral Module

The most general form of a longeron-actuated, octahedral-based VGT module is illustrated in Fig. 3.8. This specific geometry is often referred to as a Stewart platform manipulator. This active truss consists of six static length members and six variable length members which are joined together at six nodes $(n_1, n_2, \ldots, n_6)$ with spheric joints. The fixed-length members are joined together in sets of three to form two triangles. For convenience, it can be assumed without loss of generality

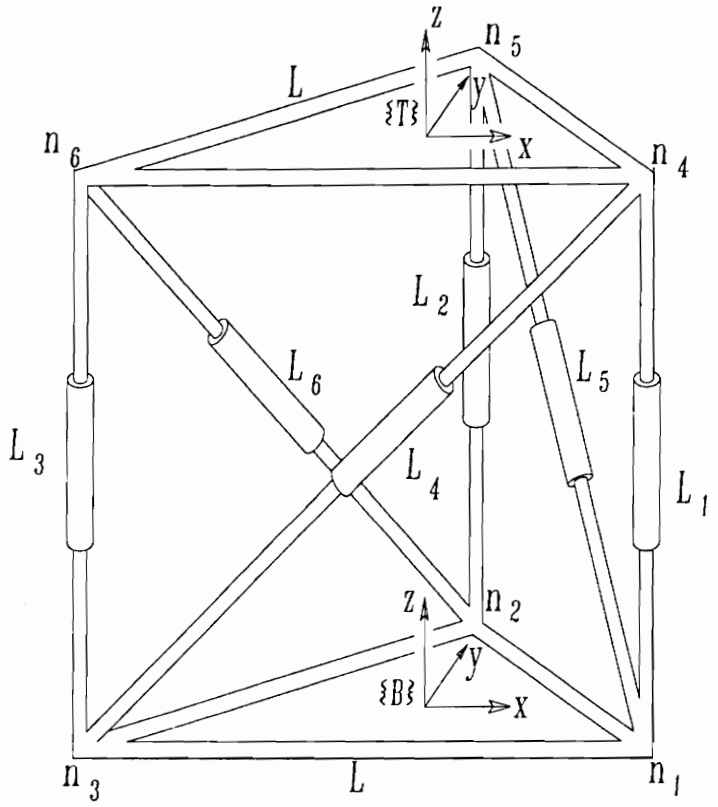


Figure 3.8: General, Longeron-Actuated, VGT Module (A Stewart Platform)

that the static-length members are of equal lengths, and thus form two equilateral triangles. These two equilateral triangles will form the interface between the static and active truss modules. Other geometries for these two triangles are possible. These cases can be analyzed using the same procedures outlined below with only slight notational modifications. One of the static triangles is used to define the base plane and has coordinate frame $\mathcal{B}$ at its centroid. The other triangle will form the top plane with frame $\mathcal{T}$ at its centroid. The length of the extensible members will be denoted by the control variables $l_1, l_2, \ldots, l_6$. In this analysis, all fixed length members are assumed to be of length L. By assigning different values to the six control variables, the position and orientation of frame $\mathcal{T}$ relative to frame $\mathcal{B}$ can be altered. This relative position and orientation is described with six parameters. Three will be reserved for the x, y, and z position of frame $\mathcal{T}$ in frame $\mathcal{B}$ as described by the vector ${}^{\mathcal{B}}\vec{P}_{\mathcal{T}}$. The three remaining parameters will describe the orientation of $\mathcal{T}$ relative to $\mathcal{B}$. This is done by utilizing the standard roll-pitch-yaw notation; where roll is a rotation about the X-axis of frame $\mathcal{B}$ by γ, pitch is a rotation about the Y-axis of frame $\mathcal{B}$ by β, and yaw is a rotation about the Z-axis of frame $\mathcal{B}$ by α. This rotation of frame $\mathcal{T}$ relative to $\mathcal{B}$ can be compactly expressed as ${}^{\mathcal{B}}_{\mathcal{T}}R_{\gamma\beta\alpha}$, a 3×3 matrix possessing three independent variables.

It should be noted that it is also possible to construct a longeron-actuated, octahedral module with fewer than six extensible links. This is the case where one or more actuated longeron members are substituted with static length members. For this degenerate Stewart platform case, the forward kinematics proceed as usual. However, the inverse problem is somewhat complicated since the specified goal position and orientation must first be constrained to lie within the subspace of the manipulator.

The following sections will present solutions to the forward and inverse kine-

matics of this manipulator.

3.4.1 Forward Kinematic Analysis

The forward kinematic analysis problem for the longeron-actuated, octahedral truss can be posed as follows:

Given the lengths of the longeron members (static or extensible), $l_1, l_2, \ldots, l_6$, *solve for the position and orientation of frame* $\mathcal{T}$ *relative to frame* $\mathcal{B}$.

In general, for this configuration of manipulator, this problem cannot be solved in closed form [17]. Instead, an iterative procedure has been developed that converges to a solution in a small number of iterations. This solution is generally more accurate than can be physically measured or controlled.

To assist in the forward kinematic analysis, it is helpful to employ the concept of kinematically equivalent devices. Two devices are said to be kinematically equivalent with respect to a point if both devices exhibit exactly the same motion characteristics for that point. Note that the two devices need not be structurally equivalent. This concept will simplify the mathematics involved and will aid in the visualization of the forward kinematic analysis.

Figure 3.9 depicts the kinematically equivalent device which will be substituted for the side triangle $\triangle n_3 n_1 n_4$. In this case, two spheric joints and two prismatic joints (Fig. 3.9-A) have been replaced by one cylindric joint and one prismatic joint (Fig. 3.9-B). The cylindric joint is positioned along its axis, $\hat{u}_1$, at the point b_1. An explicit expression for $\vec{b}_1$ can be found as follows:

$$\psi_1 = \arccos\left(\frac{L^2 + l_4^2 - l_1^2}{2Ll_4}\right), \tag{3.5}$$

therefore,

$$\vec{b}_1 = \vec{n}_3 + l_4 \cos\psi_1 \hat{u}_1. \tag{3.6}$$

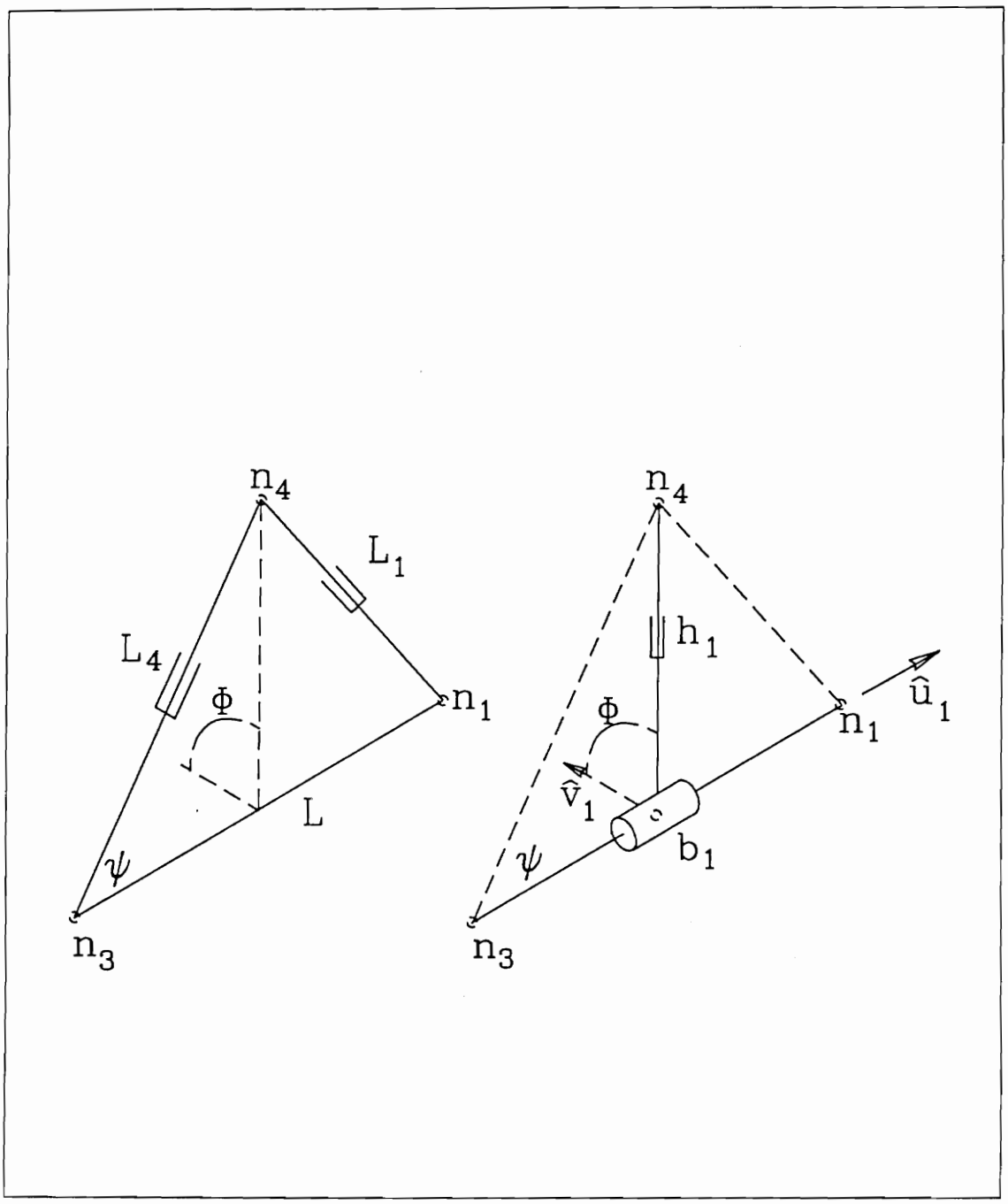


Figure 3.9: Kinematically Equivalent Devices Used for Analysis

Now h_1, the altitude of triangle $\triangle n_3 n_1 n_4$,can be found.

$$h_1 = l_4 \sin \psi_1. \tag{3.7}$$

This is equivalent to the extension of the prismatic member which is shown in Fig. 3.9-B. The unit vector $\hat{v}_1$ is defined to be normal to $\hat{u}_1$ and is constrained to lie in the plane of $\triangle n_1 n_2 n_3$. Now the point n_4 can be located as follows:

$$\vec{n}_4 = \vec{b}_1 + h_1 R_{[\hat{u}_1, \phi_1]} \hat{v}_1; \tag{3.8}$$

where $R_{[\hat{u}_1, \phi_1]} \hat{v}_1$ represents a rotation of $\hat{v}_1$ about $\hat{u}_1$ by an amount ϕ_1. Examination of the four previous equations reveals that $\vec{n}_4$ is a function dependent only on L, L_1, l_4, and ϕ_1. For the forward kinematics problem, $L, l_1, l_2, \ldots, l_6$ are known. Therefore, the vector $\vec{n}_4$ is only dependent on the unknown ϕ_1. Similar results can be obtained for the other two side triangles. These results are now summarized below:

$$\begin{aligned} \vec{n}_4 &= \vec{n}_3 + l_4 \cos \psi_1 \hat{u}_1 + l_4 \sin \psi_1 R_{[\hat{u}_1, \phi_1]} \hat{v}_1, \\ \vec{n}_5 &= \vec{n}_1 + l_5 \cos \psi_2 \hat{u}_2 + l_5 \sin \psi_2 R_{[\hat{u}_2, \phi_2]} \hat{v}_2, \\ \vec{n}_6 &= \vec{n}_2 + l_6 \cos \psi_3 \hat{u}_3 + l_6 \sin \psi_3 R_{[\hat{u}_3, \phi_3]} \hat{v}_3; \end{aligned} \tag{3.9}$$

where,

$$\psi_2 = \arccos \left(\frac{L^2 + l_5^2 - l_2^2}{2 L l_5} \right), \tag{3.10}$$

$$\psi_3 = \arccos \left(\frac{L^2 + l_6^2 - l_3^2}{2 L l_6} \right). \tag{3.11}$$

The distance between nodes of the top triangle can now be found for any assumed set of ϕ's as:

$$\begin{aligned} D_1 &= \|\vec{n}_4 - \vec{n}_5\| = \sqrt{(\vec{n}_4 - \vec{n}_5)\cdot(\vec{n}_4 - \vec{n}_5)}, \\ D_2 &= \|\vec{n}_5 - \vec{n}_6\| = \sqrt{(\vec{n}_5 - \vec{n}_6)\cdot(\vec{n}_5 - \vec{n}_6)}, \\ D_3 &= \|\vec{n}_6 - \vec{n}_4\| = \sqrt{(\vec{n}_6 - \vec{n}_4)\cdot(\vec{n}_6 - \vec{n}_4)}. \end{aligned} \tag{3.12}$$

It is now necessary to check for closure. In order to have a valid assembly, the nodes of the top triangle must be a distance L apart. Thus, the following three equations must be satisfied:

$$\begin{aligned} f_1(\phi_1, \phi_2) &= D_1 - L = 0, \\ f_2(\phi_2, \phi_3) &= D_2 - L = 0, \\ f_3(\phi_3, \phi_1) &= D_3 - L = 0. \end{aligned} \tag{3.13}$$

This can be done simply by applying a Newton-Raphson root-finding algorithm which centers around the following linearization:

$$\begin{bmatrix} \frac{\partial f_1}{\partial \phi_1} & \frac{\partial f_1}{\partial \phi_2} & 0 \\ 0 & \frac{\partial f_2}{\partial \phi_2} & \frac{\partial f_2}{\partial \phi_3} \\ \frac{\partial f_3}{\partial \phi_1} & 0 & \frac{\partial f_3}{\partial \phi_3} \end{bmatrix} \begin{bmatrix} \Delta\phi_1 \\ \Delta\phi_2 \\ \Delta\phi_3 \end{bmatrix} = \begin{bmatrix} -f_1 \\ -f_2 \\ -f_3 \end{bmatrix}. \tag{3.14}$$

For any given iteration, $\Delta\phi_1, \Delta\phi_2$, and $\Delta\phi_3$ can be found, and the current "guessed" position can be modified by:

$$\begin{aligned} \phi_{1_{new}} &= \phi_{1_{old}} + \sigma(\Delta\phi_1), \\ \phi_{2_{new}} &= \phi_{2_{old}} + \sigma(\Delta\phi_2), \\ \phi_{3_{new}} &= \phi_{3_{old}} + \sigma(\Delta\phi_3). \end{aligned}$$

where σ is a damping factor added to avoid overshoot and speed convergence. In typical operation, with $\sigma = 0.90$, this procedure converges within an acceptable range after only four iterations. Other more efficient non-linear equation solving routines could also be implemented. Upon convergence of the routine outlined above, the position of nodes n_4, n_5, and n_6 are known. The origin of frame $\mathcal{T}$ can now be described in frame $\mathcal{B}$ as the centroid of these three nodes.

$$^{\mathcal{B}}\vec{P}_{\mathcal{T}} = \frac{(\vec{n}_4 + \vec{n}_5 + \vec{n}_6)}{3}. \tag{3.15}$$

The three unit vectors which form the coordinate axes of frame $\mathcal{T}$ can be described in frame $\mathcal{B}$ as follows:

$$^{\mathcal{B}}\hat{X}_{\mathcal{T}} = \frac{(\vec{n}_4 - \vec{n}_6)}{||\vec{n}_4 - \vec{n}_6||}, \tag{3.16}$$

$$^{\mathcal{B}}\hat{Z}_{\mathcal{T}} = \frac{(\vec{n}_4 - \vec{n}_6) \times (\vec{n}_5 - \vec{n}_6)}{||\,(\vec{n}_4 - \vec{n}_6) \times (\vec{n}_5 - \vec{n}_6)\,||}, \tag{3.17}$$

$$^{\mathcal{B}}\hat{Y}_{\mathcal{T}} = {}^{\mathcal{B}}\hat{Z}_{\mathcal{T}} \times {}^{\mathcal{B}}\hat{X}_{\mathcal{T}}. \tag{3.18}$$

Note that each of the above expressions are vector quantities and, as such, each has three components. Now a rotation matrix describing the orientation of frame $\mathcal{T}$ with respect to frame $\mathcal{B}$ is given by:

$$^{\mathcal{B}}_{\mathcal{T}}R_{\gamma,\beta,\alpha} = \begin{bmatrix} {}^{\mathcal{B}}\hat{X}_{\mathcal{T}x} & {}^{\mathcal{B}}\hat{Y}_{\mathcal{T}x} & {}^{\mathcal{B}}\hat{Z}_{\mathcal{T}x} \\ {}^{\mathcal{B}}\hat{X}_{\mathcal{T}y} & {}^{\mathcal{B}}\hat{Y}_{\mathcal{T}y} & {}^{\mathcal{B}}\hat{Z}_{\mathcal{T}y} \\ {}^{\mathcal{B}}\hat{X}_{\mathcal{T}z} & {}^{\mathcal{B}}\hat{Y}_{\mathcal{T}z} & {}^{\mathcal{B}}\hat{Z}_{\mathcal{T}z} \end{bmatrix}; \tag{3.19}$$

where each column of the matrix contains the x, y, and z components of each coordinate axis that composes frame $\mathcal{T}$. If desired, this expression can now be equated to the standard roll-pitch-yaw representation of the rotation matrix as presented below:

$$
{}^{\mathcal{B}}_{\mathcal{T}}R_{\gamma,\beta,\alpha} = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}; \tag{3.20}
$$

where $c\beta = \cos\beta, s\beta = \sin\beta$, etc.

The roll-pitch-yaw description can now be obtained uniquely from one of the three following sets of relations [12, pages 41–42].

•Case 1. if $\left\| {}^{\mathcal{B}}\hat{X}_{\mathcal{T}z} \right\| \neq 1$,

$$
\begin{aligned}
\gamma &= \text{ATAN2}\left({}^{\mathcal{B}}\hat{Y}_{\mathcal{T}z}, {}^{\mathcal{B}}\hat{Z}_{\mathcal{T}z}\right) \\
\beta &= \text{ATAN2}\left(-{}^{\mathcal{B}}\hat{X}_{\mathcal{T}z}, \sqrt{{}^{\mathcal{B}}\hat{X}^2_{\mathcal{T}x} + {}^{\mathcal{B}}\hat{X}^2_{\mathcal{T}y}}\right) \\
\alpha &= \text{ATAN2}\left({}^{\mathcal{B}}\hat{X}_{\mathcal{T}y}, {}^{\mathcal{B}}\hat{X}_{\mathcal{T}x}\right)
\end{aligned} \tag{3.21}
$$

•Case 2. if ${}^{\mathcal{B}}\hat{X}_{\mathcal{T}z} = +1$,

$$
\begin{aligned}
\gamma &= \text{ATAN2}\left({}^{\mathcal{B}}\hat{Y}_{\mathcal{T}x}, {}^{\mathcal{B}}\hat{Y}_{\mathcal{T}y}\right) \\
\beta &= 90^o \\
\alpha &= 0
\end{aligned} \tag{3.22}
$$

•Case 3. if ${}^{\mathcal{B}}\hat{X}_{\mathcal{T}z} = -1$,

$$
\begin{aligned}
\gamma &= -\text{ATAN2}\left({}^{\mathcal{B}}\hat{Y}_{\mathcal{T}x}, {}^{\mathcal{B}}\hat{Y}_{\mathcal{T}y}\right) \\
\beta &= -90^o \\
\alpha &= 0
\end{aligned} \tag{3.23}
$$

The results of the forward kinematic solution just presented are non-linear and hence no unique solution exists. In fact, for this particular truss, sixteen distinct assemblies, or branches, exist [2, 3]. These solutions represent mathematically acceptable solutions, some of which the truss could physically obtain, but only after unpinning the joints and reassembling the truss within a new branch. From

an operational standpoint, all of these solutions are not practical. To result in a practical solution, this mathematical solution technique must somehow be tailored to omit all but the desired solution. This is most efficiently done by limiting the range of ϕ_1, ϕ_2, and ϕ_3 in the iterative procedure.

Historically, manipulators in which the kinematics must be solved iteratively have been avoided. This can primarily be attributed to convergence problems, and, to a lesser degree, the speed at which these algorithms could be executed. It is true that for large moves convergence of the proposed algorithms might present a problem. However, at the speed at which modern controllers can update a position, there is no need to move the system a great distance in one step. For small moves the values of ϕ_1, ϕ_2, and ϕ_3 are known relatively accurately to start with. Thus, convergence is not a problem unless singularities are present. Singularity points can be identified by forming a Jacobian matrix and setting its determinate equal to zero. However, this particular configuration can easily be designed such that the singularity points occur outside of the physical workspace, which is limited by the range of the extensible longerons.

3.4.2 Inverse Kinematic Analysis

The inverse kinematic analysis of the longeron-actuated, octahedral truss is both closed-form and linear. The inverse kinematic analysis problem can be stated as follows:

Given the position vector, $\vec{P}_1$, *and* α, β, γ, *solve for the set of longeron lengths* $\{l_1, l_2, \ldots, l_6\}$.

Since α, β, and γ are known, the rotation from $\mathcal{B}$ to $\mathcal{T}$ can be described by the rotation matrix

$$ {}^{\mathcal{B}}_{\mathcal{T}}R_{\gamma,\beta,\alpha} = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}. \tag{3.24} $$

A transform can now be defined which encompasses both the rotation and translation of $\mathcal{T}$ relative to $\mathcal{B}$. This transform is defined as:

$$ {}^{\mathcal{B}}_{\mathcal{T}}T = \begin{bmatrix} & [{}^{\mathcal{B}}_{\mathcal{T}}R] & & [{}^{\mathcal{B}}\vec{P}_1] \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.25} $$

Nodes n_4, n_5, and n_6, which are known in $\mathcal{T}$, can now be described in $\mathcal{B}$.

$$ \begin{aligned} {}^{\mathcal{B}}\vec{n}_4 &= {}^{\mathcal{B}}_{\mathcal{T}}T^{\mathcal{T}}\vec{n}_4, \\ {}^{\mathcal{B}}\vec{n}_5 &= {}^{\mathcal{B}}_{\mathcal{T}}T^{\mathcal{T}}\vec{n}_5, \\ {}^{\mathcal{B}}\vec{n}_6 &= {}^{\mathcal{B}}_{\mathcal{T}}T^{\mathcal{T}}\vec{n}_6. \end{aligned} \tag{3.26} $$

Finally, the length of the extensible members can be solved for directly as the euclidean distance between appropriate nodes.

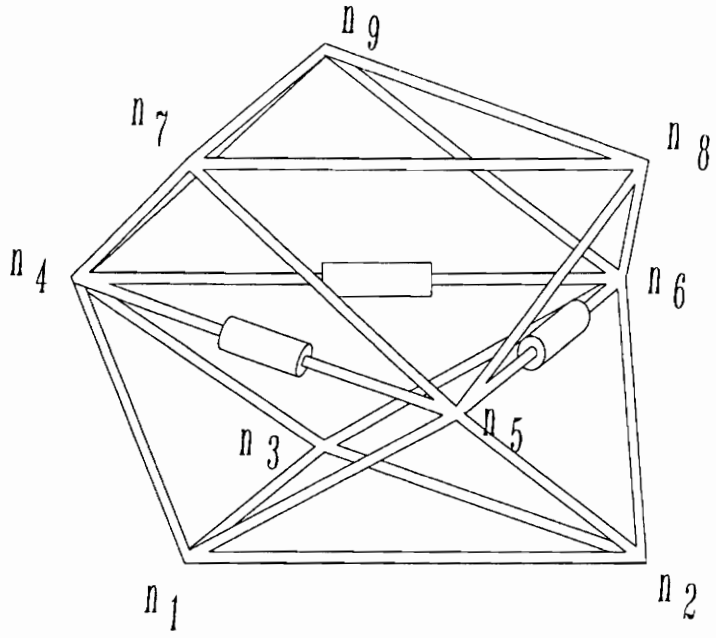
$$ \begin{aligned} l_1 &= \left\| {}^{\mathcal{B}}\vec{n}_4 - {}^{\mathcal{B}}\vec{n}_1 \right\| = \sqrt{({}^{\mathcal{B}}\vec{n}_4 - {}^{\mathcal{B}}\vec{n}_1)\cdot({}^{\mathcal{B}}\vec{n}_4 - {}^{\mathcal{B}}\vec{n}_1)}, \\ l_2 &= \left\| {}^{\mathcal{B}}\vec{n}_5 - {}^{\mathcal{B}}\vec{n}_2 \right\| = \sqrt{({}^{\mathcal{B}}\vec{n}_5 - {}^{\mathcal{B}}\vec{n}_2)\cdot({}^{\mathcal{B}}\vec{n}_5 - {}^{\mathcal{B}}\vec{n}_2)}, \\ l_3 &= \left\| {}^{\mathcal{B}}\vec{n}_6 - {}^{\mathcal{B}}\vec{n}_3 \right\| = \sqrt{({}^{\mathcal{B}}\vec{n}_6 - {}^{\mathcal{B}}\vec{n}_3)\cdot({}^{\mathcal{B}}\vec{n}_6 - {}^{\mathcal{B}}\vec{n}_3)}, \\ l_4 &= \left\| {}^{\mathcal{B}}\vec{n}_4 - {}^{\mathcal{B}}\vec{n}_3 \right\| = \sqrt{({}^{\mathcal{B}}\vec{n}_4 - {}^{\mathcal{B}}\vec{n}_3)\cdot({}^{\mathcal{B}}\vec{n}_4 - {}^{\mathcal{B}}\vec{n}_3)}, \\ l_5 &= \left\| {}^{\mathcal{B}}\vec{n}_5 - {}^{\mathcal{B}}\vec{n}_1 \right\| = \sqrt{({}^{\mathcal{B}}\vec{n}_5 - {}^{\mathcal{B}}\vec{n}_1)\cdot({}^{\mathcal{B}}\vec{n}_5 - {}^{\mathcal{B}}\vec{n}_1)}, \\ l_6 &= \left\| {}^{\mathcal{B}}\vec{n}_6 - {}^{\mathcal{B}}\vec{n}_2 \right\| = \sqrt{({}^{\mathcal{B}}\vec{n}_6 - {}^{\mathcal{B}}\vec{n}_2)\cdot({}^{\mathcal{B}}\vec{n}_6 - {}^{\mathcal{B}}\vec{n}_2)}. \end{aligned} \tag{3.27} $$

Note that there can only be one solution for the euclidean distance between two points. This means that with the absence of any other constraints, a unique solution for any specified position and orientation always exists. In reality, of course, other constraints do exist. The primary constraint for this device is the range of its

variable length members. The inverse kinematic results must be checked to ensure that they are within the physical range of the extensible links. If not, the given goal position is outside of the manipulators workspace and cannot be reached. To address the case where a truss may contain k static length longerons, the inverse solution can proceed as usual. However, the final results (link lengths) must be checked to ensure that inverse analysis produces corresponding values acceptably close to the true static lengths. If not, the given goal position does not lie within the $(6-k)$-dimensional subspace of the manipulator.

3.5 Analysis of a General, Batten-Actuated, Double-Octahedral Module

Section 3.4 introduced an analysis procedure for a VGT geometry based on an octahedral unit cell structure. This section will examine an active truss geometry, again based on the octahedral unit cell structure. This structure consists simply of two octahedra sharing a common triangular facet. For this reason it is referred to as a double-octahedral geometry. Although many possible choices exist for how to actuate this truss structure, this work will address only trusses in which the midplane battens are actuated. This general class of VGT manipulator has been shown to posses a good compromise between high strength and useful workspace.[4] [62, 63] Figure 3.10 illustrates a batten-actuated VGT module. Note that the top and bottom batten triangles are defined by fixed length members. These two triangles will serve as the attachment points for the static truss sections. The nodes of the truss are numbered n_1, n_2, ..., n_9 as shown in Fig. 3.10. The fixed length members of the base and top planes will be of length L_{Base} and L_{Top} respectively. The length of the extensible links will again be denoted by the lower case variables l_1, l_2, and l_3. For this general analysis, it will be assumed that the fixed length



Link Identification

Link	Between Nodes
l_1	$n_4 - n_5$
l_2	$n_5 - n_6$
l_3	$n_6 - n_4$
L_1	$n_1 - n_4$
L_2	$n_2 - n_5$
L_3	$n_3 - n_6$
L_4	$n_3 - n_4$
L_5	$n_1 - n_5$
L_6	$n_2 - n_6$
L_7	$n_5 - n_7$
L_8	$n_6 - n_8$
L_9	$n_4 - n_9$
L_{10}	$n_4 - n_7$
L_{11}	$n_5 - n_8$
L_{12}	$n_6 - n_9$

Figure 3.10: Batten Actuated, Double-Octahedral VGT Module

longeron members can be of different lengths. These lengths will be identified as $L_1, L_2, \ldots, L_{12}$.

Since this truss configuration is again based on an octahedral unit cell, the kinematic analysis is similar to the longeron-actuated octahedron previously presented. Because of this inherent similarity, the following forward and inverse kinematic discussions will be somewhat abbreviated. For further clarifications, refer back to Section 3.4.

3.5.1 Forward Kinematic Analysis

The forward kinematic problem for this active module may be summarized as follows: *given the lengths of the fixed length members ($L_1, \ldots, L_{12}$), and the lengths of the extensible members (l_1, l_2, and l_3), find the position and orientation of frame $\mathcal{T}$ relative to frame $\mathcal{B}$.*

Again, the concept of a kinematically equivalent device will greatly simplify the analysis of the octahedron. The kinematic equivalent to the first octahedral cell of the, double-octahedral truss is shown in Fig. 3.11-A.

By analogy to the previous octahedral analysis, the positions of the midplane nodes(n_4, n_5, n_6),can be identified as a function of the unknowns ϕ_1, ϕ_2, and ϕ_3.

$$\begin{aligned} \vec{n}_4 &= \vec{n}_3 + L_4 \cos\psi_1 \hat{u}_1 + L_4 \sin\psi_1 R_{[\hat{u}_1,\phi_1]}\hat{v}_1, \\ \vec{n}_5 &= \vec{n}_1 + L_5 \cos\psi_2 \hat{u}_2 + L_5 \sin\psi_2 R_{[\hat{u}_2,\phi_2]}\hat{v}_2, \\ \vec{n}_6 &= \vec{n}_2 + L_6 \cos\psi_3 \hat{u}_3 + L_6 \sin\psi_3 R_{[\hat{u}_3,\phi_3]}\hat{v}_3; \end{aligned} \tag{3.28}$$

where,

$$\psi_1 = \arccos\left(\frac{L_{Base}^2 + L_4^2 - L_1^2}{2L_{Base}L_4}\right), \tag{3.29}$$

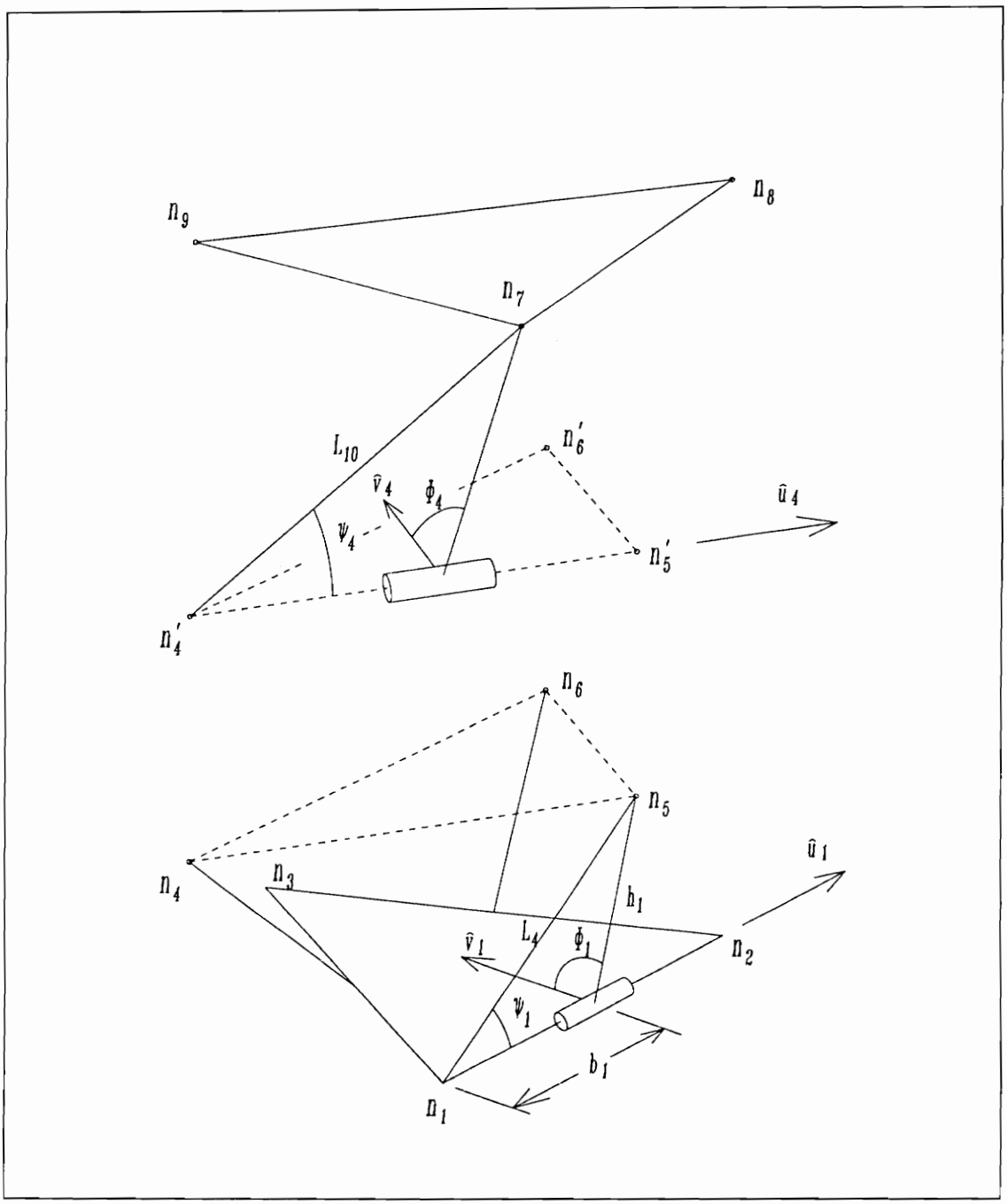


Figure 3.11: Kinematic Equivalent to the Batten-Actuated, Double-Octahedral VGT

$$\psi_2 = \arccos\left(\frac{L_{Base}^2 + L_5^2 - L_2^2}{2L_{Base}L_5}\right), \tag{3.30}$$

$$\psi_3 = \arccos\left(\frac{L_{Base}^2 + L_6^2 - L_3^2}{2L_{Base}L_6}\right). \tag{3.31}$$

The distance between nodes of the midplane triangle can now be found for any given ϕ_1, ϕ_2, and ϕ_3 as:

$$\begin{aligned} D_1 &= \|\vec{n}_4 - \vec{n}_5\| = \sqrt{(\vec{n}_4 - \vec{n}_5)\cdot(\vec{n}_4 - \vec{n}_5)}, \\ D_2 &= \|\vec{n}_5 - \vec{n}_6\| = \sqrt{(\vec{n}_5 - \vec{n}_6)\cdot(\vec{n}_5 - \vec{n}_6)}, \\ D_3 &= \|\vec{n}_6 - \vec{n}_4\| = \sqrt{(\vec{n}_6 - \vec{n}_4)\cdot(\vec{n}_6 - \vec{n}_4)}. \end{aligned} \tag{3.32}$$

It is now necessary to check for closure. In order to have a valid assembly, the nodes of the midplane triangle must be the specified distance apart. Thus, the following three equations must be satisfied:

$$\begin{aligned} f_1(\phi_1, \phi_2) &= D_1 - l_1 = 0, \\ f_2(\phi_2, \phi_3) &= D_2 - l_2 = 0, \\ f_3(\phi_3, \phi_1) &= D_3 - l_3 = 0. \end{aligned} \tag{3.33}$$

The Newton-Raphson root-finding algorithm may now be applied exactly as it was for the longeron-actuated truss. Upon convergence, a feasible set of ϕ_1, ϕ_2, and ϕ_3 is obtained. These must be checked to ensure the solution is in the proper branch.

At this stage of the analysis, the position of the three midplane nodes are known. As a practical consideration, all of the double-octahedral VGT cells constructed to date have contained a mid-plane joint offset. This deviation from ideal truss design is necessary to ensure an adequate range of motion for the truss

joint. In addition, incorporating this offset into the design avoids the complications involved in having five co-located spheric joints. If a mid-plane joint offset is present, this distance can easily be included in the analysis by forming three additional midplane nodes (n_4', n_5', and n_6') that are displaced the amount of the offset in a direction normal to the midplane. The solution for the second octahedron is obtained by essentially repeating the solution procedure above. One possible kinematic equivalent to the second octahedral cell is shown in Fig. 3.11-B. Note that $\hat{u}_4$, $\hat{u}_5$, and $\hat{u}_6$ are defined by the position of the midplane nodes. These unit vectors may be described in the base coordinate frame, $\mathcal{B}$, as:

$$\hat{u}_4 = \frac{\vec{n}_5 - \vec{n}_4}{l_1} \tag{3.34}$$

$$\hat{u}_5 = \frac{\vec{n}_6 - \vec{n}_5}{l_2} \tag{3.35}$$

$$\hat{u}_6 = \frac{\vec{n}_4 - \vec{n}_6}{l_3} \tag{3.36}$$

The unit vectors $\hat{v}_4$, $\hat{v}_5$, and $\hat{v}_6$ must also be defined such that they lie within the midplane, perpendicular to the corresponding actuated member. This is most easily accomplished by defining another unit vector, $\hat{w}$, that is normal to the midplane. Using the vector cross product,

$$\hat{w} = \frac{\hat{u}_5 \times (-\hat{u}_4)}{||\hat{u}_5 \times (-\hat{u}_4)||}. \tag{3.37}$$

Now,

$$\hat{v}_4 = \hat{w} \times \hat{u}_4 \tag{3.38}$$

$$\hat{v}_5 = \hat{w} \times \hat{u}_5 \tag{3.39}$$

$$\hat{v}_6 = \hat{w} \times \hat{u}_6 \tag{3.40}$$

The position of the top plane nodes may now be identified solely as a function of the unknowns ϕ_4, ϕ_5, and ϕ_6.

$$\begin{aligned} \vec{n}_7 &= \vec{n}'_4 + L_{10}\cos\psi_4\hat{u}_4 + L_{10}\sin\psi_4 R_{[\hat{u}_4,\phi_4]}\hat{v}_4, \\ \vec{n}_8 &= \vec{n}'_5 + L_{11}\cos\psi_5\hat{u}_5 + L_{11}\sin\psi_5 R_{[\hat{u}_5,\phi_5]}\hat{v}_5, \\ \vec{n}_9 &= \vec{n}'_6 + L_{12}\cos\psi_6\hat{u}_6 + L_{12}\sin\psi_6 R_{[\hat{u}_6,\phi_6]}\hat{v}_6; \end{aligned} \tag{3.41}$$

where,

$$\psi_4 = \arccos\left(\frac{l_1^2 + L_{10}^2 - L_7^2}{2l_1L_{10}}\right), \tag{3.42}$$

$$\psi_5 = \arccos\left(\frac{l_2^2 + L_{11}^2 - L_8^2}{2l_2L_{11}}\right), \tag{3.43}$$

$$\psi_6 = \arccos\left(\frac{l_3^2 + L_{12}^2 - L_9^2}{2l_3L_{12}}\right). \tag{3.44}$$

The distance between nodes of the top plane triangle can now be found for any given ϕ_4, ϕ_5, and ϕ_6 as:

$$\begin{aligned} D_4 &= \|\vec{n}_{10} - \vec{n}_{11}\| = \sqrt{(\vec{n}_{10} - \vec{n}_{11})\cdot(\vec{n}_{10} - \vec{n}_{11})}, \\ D_5 &= \|\vec{n}_{11} - \vec{n}_{12}\| = \sqrt{(\vec{n}_{11} - \vec{n}_{12})\cdot(\vec{n}_{11} - \vec{n}_{12})}, \\ D_6 &= \|\vec{n}_{12} - \vec{n}_{10}\| = \sqrt{(\vec{n}_{12} - \vec{n}_{10})\cdot(\vec{n}_{12} - \vec{n}_{10})}. \end{aligned} \tag{3.45}$$

Again, it is necessary to check for closure. In order to have a valid assembly, the nodes of the top plane triangle must be the specified distance apart. Thus, the following three equations must be satisfied:

$$\begin{aligned} f_1(\phi_1, \phi_2) &= D_1 - L_{Top} = 0, \\ f_2(\phi_2, \phi_3) &= D_2 - L_{Top} = 0, \\ f_3(\phi_3, \phi_1) &= D_3 - L_{Top} = 0. \end{aligned} \tag{3.46}$$

The Newton-Raphson root-finding algorithm may again be applied. Upon convergence, a feasible set of ϕ_4, ϕ_5, and ϕ_6 is obtained. These must be checked to ensure the solution is in the proper branch.

The origin of frame $\mathcal{T}$ can now be described in frame $\mathcal{B}$ as the centroid of the top plane triangle.

$${}^{\mathcal{B}}\vec{P}_{\mathcal{T}} = \frac{(\vec{n}_{10} + \vec{n}_{11} + \vec{n}_{12})}{3}. \tag{3.47}$$

The three unit vectors which form the coordinate axes of frame $\mathcal{T}$ can be described in frame $\mathcal{B}$ as follows:

$${}^{\mathcal{B}}\hat{X}_{\mathcal{T}} = \frac{(\vec{n}_{10} - \vec{n}_{12})}{||\vec{n}_{10} - \vec{n}_{12}||}, \tag{3.48}$$

$${}^{\mathcal{B}}\hat{Z}_{\mathcal{T}} = \frac{(\vec{n}_{10} - \vec{n}_{12}) \times (\vec{n}_{11} - \vec{n}_{12})}{||\,(\vec{n}_{10} - \vec{n}_{12}) \times (\vec{n}_{11} - \vec{n}_{12})\,||}, \tag{3.49}$$

$${}^{\mathcal{B}}\hat{Y}_{\mathcal{T}} = {}^{\mathcal{B}}\hat{Z}_{\mathcal{T}} \times {}^{\mathcal{B}}\hat{X}_{\mathcal{T}}. \tag{3.50}$$

Now, the transformation matrix that describes frame $\mathcal{T}$ with respect to frame $\mathcal{B}$ is given by:

$${}^{\mathcal{B}}_{\mathcal{T}}T_{(l_1,l_2,l_3)} = \begin{bmatrix} {}^{\mathcal{B}}\hat{X}_{\mathcal{T}} & {}^{\mathcal{B}}\hat{Y}_{\mathcal{T}} & {}^{\mathcal{B}}\hat{Z}_{\mathcal{T}} & {}^{\mathcal{B}}\vec{P}_{\mathcal{T}} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.51}$$

3.5.2 Inverse Kinematic Analysis

The inverse kinematic problem for the general batten-actuated VGT module may be stated as follows: *given the desired end condition of the active module, find the extensible link lengths,* l_1, l_2, *and* l_3.

This poses a slightly different problem than the other geometries discussed thus far. Note that, if a general six degree-of-freedom position and orientation is specified, a solution is only possible if the goal lies within the three degree-of-freedom subspace of the manipulator. Another possibility is to specify only three parameters. For example, the X, Y, and Z position of frame $\mathcal{T}$ relative to $\mathcal{B}$, ${}^{\mathcal{B}}\vec{P}_{\mathcal{T}}$, could be specified. Or the angular displacement about the $\hat{X}_{\mathcal{B}}$ and $\hat{Y}_{\mathcal{B}}$ axes may be specified in conjunction with a third positional parameter; either the X, Y, or Z position of $\mathcal{T}$ relative to $\mathcal{B}$. As a subtlety of this particular geometric configuration, a single truss module cannot effect any angular displacements about the $\hat{Z}_{\mathcal{B}}$ axis. For this reason, an advantageous way of specifying a goal orientation is to specify only a unit vector normal to the top plane,${}^{\mathcal{B}}\hat{Q}$.

To solve the inverse kinematic problem for any of the above goal specifications, the forward solution must be iteratively solved until convergence is achieved. It would appear that a logical method for accomplishing this is to follow the algorithmic steps enumerated below.

1. Assume a set of l's.
2. Use the forward kinematics solution to calculate the resulting end condition.
3. Compare this to the desired end position. If converged proceed to step 7.
4. Evaluate the effect of each l on each end parameter (e.g. $\frac{\partial x}{\partial l_1}$). This again requires solving the forward kinematics problem three times.
5. Compute a set of Δl's.
6. Modify the assumed set of l's and return to step 2.

7. Verify solution feasibility.

This procedure will indeed converge to an acceptable solution. However, the forward kinematics problem must be solved a total of four times for one iteration. The real problem here is that the forward kinematics problem is itself iterative. In the case of the general double octahedral truss, solving both the first and second octahedra require iterative loops. Thus, the above outlined procedure requires that eight iterative loops be processed with each pass.

Although this method does perform well in many situations, it can be dramatically improved by imposing some constraints on the design of the truss module. If, for example, the static length links are all made the same length, a faster inverse kinematic analysis method can be employed. This design constraint may prove to be too restrictive for many applications. As a more general alternative, the restriction that will be exploited for this work is that of symmetry. Namely, that the fixed length members of the second octahedral cell must be the same length as their reflected members in the first octahedral cell. In this case, the plane of symmetry is the midplane.

3.5.3 Forward Kinematic Solutions with Symmetry Imposed

The midplane of the symmetric double-octahedral truss described above may be identified as outlined previously for the general double-octahedral. This requires an iterative procedure to find the mid-plane nodes. After the midplane is identified, the three base nodes may be projected through the midplane to form the three nodes of the top plane. See Fig. 3.12. This projection process is closed form and will be briefly discussed below.

In general, a projection, or mirroring, process can be carried out in closed form

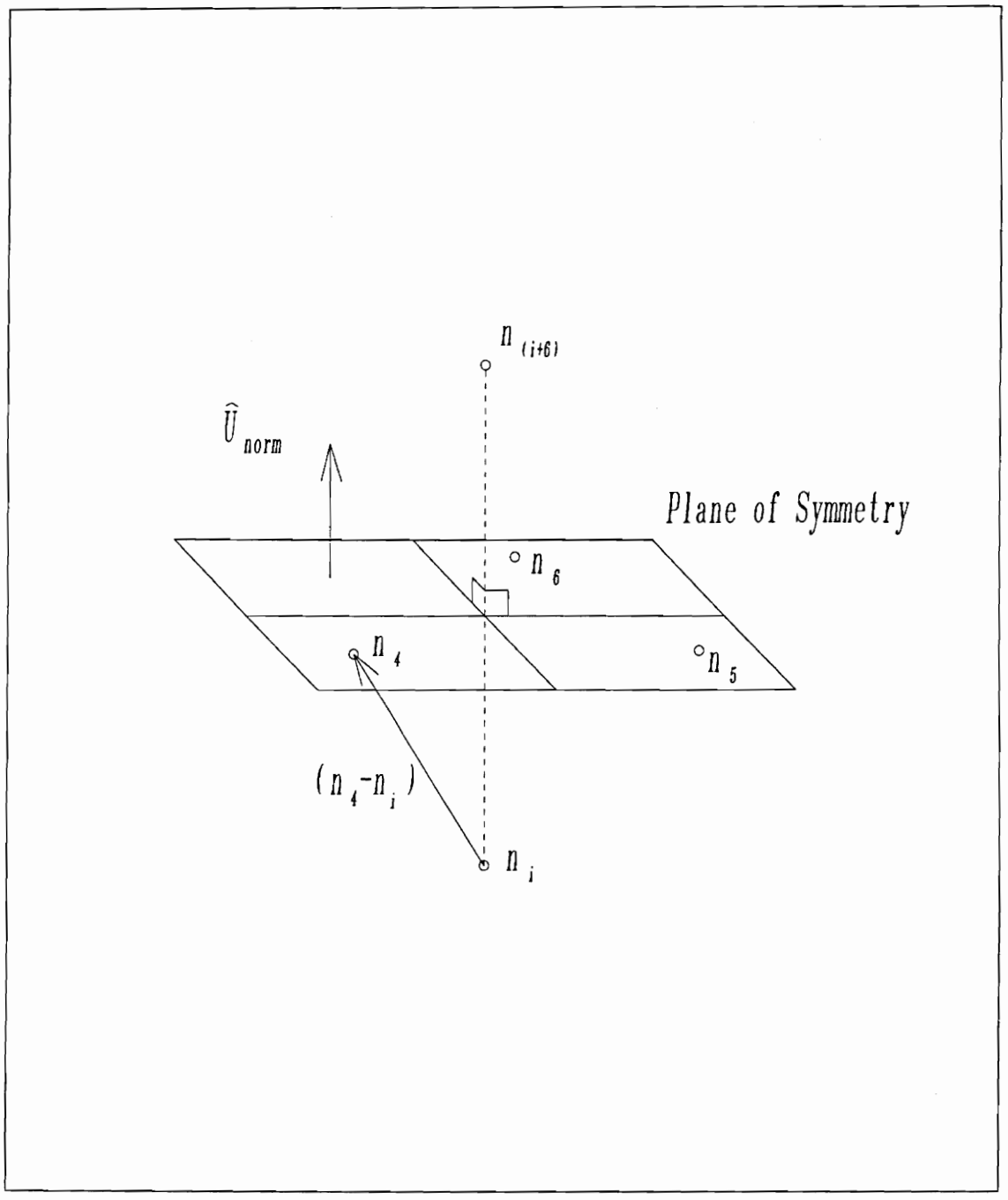


Figure 3.12: Method of projection through the plane of symmetry

if the plane of symmetry is defined and the coordinates of the point to be mirrored are known. For this particular case, the plane of symmetry is defined by nodes n_4, n_5, and n_6, or, if appropriate, the midpoints of $n_4 - n_4'$, $n_5 - n_5'$, and $n_6 - n_6'$. Thus, a unit vector normal to this plane can be defined using the vector cross product.

$$\hat{U}_{\text{norm}} = \frac{(\vec{n}_5 - \vec{n}_4) \times (\vec{n}_6 - \vec{n}_4)}{||(\vec{n}_5 - \vec{n}_4) \times (\vec{n}_6 - \vec{n}_4)||} \tag{3.52}$$

Since the base nodes must be projected perpendicular to the plane of symmetry, their mirror images must lie along a vector which passes through the base node in question and points in the direction of $\hat{U}_{\text{norm}}$. Additionally, the mirrored node must lie the same distance away from the plane of symmetry. This distance may be calculated using the vector dot product as follows:

$$d_i = (\vec{P}_{\text{sym}} - \vec{n}_i) \cdot \hat{U}_{\text{norm}} \qquad i = 1 \ldots 3 \tag{3.53}$$

where, $\vec{P}_{\text{sym}}$ represents a vector locating any point in the plane of symmetry (n_4, for example). Now, the coordinates of the top plane nodes may be identified as:

$$\vec{n}_{i+6} = \vec{n}_i + (2d_i + L_{jo})\hat{U}_{\text{norm}} \qquad i = 1 \ldots 3 \tag{3.54}$$

where L_{jo} is the length of the midplane joint offset, if any is present.

The forward kinematic procedure now requires only one iterative loop.

3.5.4 Inverse Kinematic Solutions with Symmetry Imposed

This section will discuss two general methods for arriving at a solution to the inverse problem: iterative and closed-form. The ultimate goal of this work is to develop algorithms that will be useful in solving a multi-module manipulator. This section will show that both iterative and closed-form inverse solutions to a single

module are possible. The closed-form solution will be utilized in one of the multi-module solution techniques to be discussed later. However, the iterative version provides many insights and will be utilized extensively for other formulations of the multi-module inverse problem.

Iterative Solutions

It was stated earlier that the forward kinematic solution of the symmetric truss requires only one iterative loop. This implies that using the inverse solution outlined for the general double-octahedral geometry will require only four iterative processes per pass. One iteration to determine the present position and orientation given l_1, l_2, and l_3, and three iterations to evaluate the effects of varying l_1, l_2, and l_3. This computational burden can be further reduced. Since the goal of the inverse kinematic problem is to identify the required extensible link lengths, it is not necessary to know the l's until a solution is found. As outlined, the first step in the inverse solution process was to assume values for the l's. This represents the conventional input specification. For this parallel, three degree-of-freedom device a canonical specification set exists that results in a linear forward solution. If values are assumed for the angles ϕ_1, ϕ_2, and ϕ_3, the positional state of the truss is completely and uniquely defined. The following procedural outline may now be used:

1. Assume a set of ϕ's.
2. Calculate in closed-form the midplane nodes.
3. Project the base nodes through the midplane to form the top plane.
4. Compare this to the desired end position. If converged proceed to step 8.
5. Evaluate the effect of each ϕ on each end parameter (e.g. $\frac{\partial x}{\partial \phi_1}$). These partials may now be evaluated in closed-form.

6. Compute a set of $\Delta\phi$'s.
7. Modify the assumed set of ϕ's and return to step 2.
8. Use the final set of ϕ's to compute in closed-form the actual l's required.

The inverse kinematics process now requires just the one main iterative loop. This was accomplished by iteratively solving the closed-form canonical input specification forward kinematic problem.

Closed-Form Solutions

This section summarizes the inverse kinematic research conducted by Padmanabhan, et al.[41, 42], who identified closed-form inverse kinematic solutions for the symmetric, batten-actuated, double-octahedral truss. Since this is a three degree-of-freedom device, the inverse goal specification can only possess three degrees-of-freedom. For this reason the inverse kinematic problem has been divided into two classes of problems. One class of problems is concerned with identifying the link lengths necessary to produce a given (X, Y, Z) position of the end-effector. This will be termed the inverse positioning problem. The other class of problems is concerned with identifying the link lengths required to produce a given extension, and angular rotation about two axes. This is termed the inverse gimbal problem. Both of theses tasks have been successfully solved in closed-form. Figure 3.13 illustrates the two inverse goal concepts. Note that this figure also presents the more intuitive gimbal model of the VGT. This module behaves much like a conventional Hooke's coupling, however, it has the ability to alter the parameter shown as d in Fig. 3.13. While these closed-form solutions are extremely useful for solving individual modules, they do not address the problem posed by the modular manipulator where additional degrees-of-freedom would be chained to the three degree-of-freedom module.

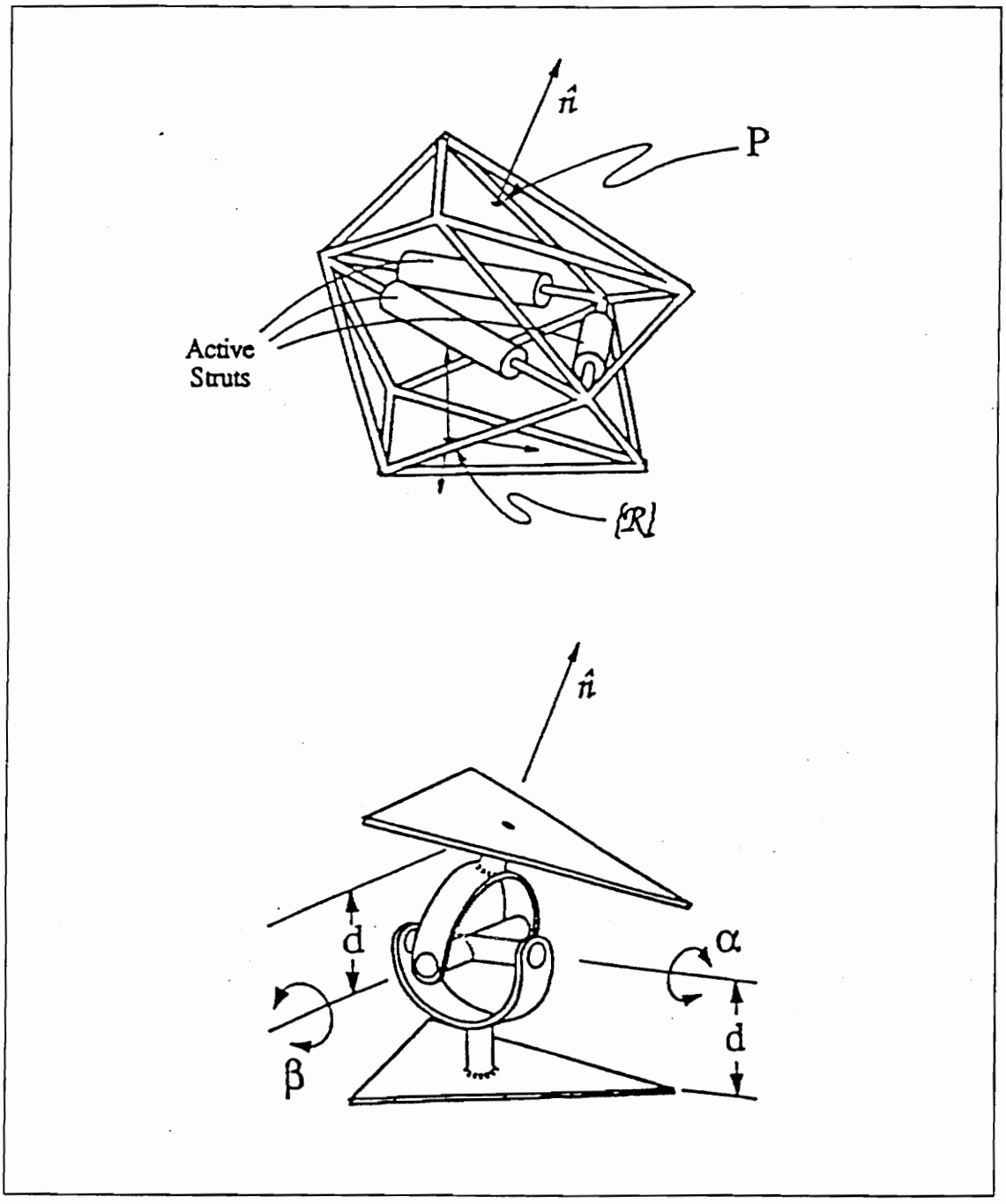


Figure 3.13: Two Possible Goal Specifications for Closed-Form Inverse Analysis

3.6 Description of the Static Truss Modules

The static truss sections will be treated in the same manner as the active truss sections. Since there will be no change in geometry of the static sections, a full kinematic analysis is not required. However, each static section must have a known transformation which describes the relative position and orientation of the faces which are designed to accommodate attachment of other truss elements. This transformation can be calculated from the given geometry or simply measured on the physical truss module. In general, the static sections are in the form of triangular prisms of various lengths and cross-sectional dimensions. A typical assortment of static truss sections is shown in Fig 3.14. Note that, some static sections may be tapered and the two end faces may not be parallel. In all cases the critical information to be obtained is the transformation from $\mathcal{B}$ to $\mathcal{T}$.

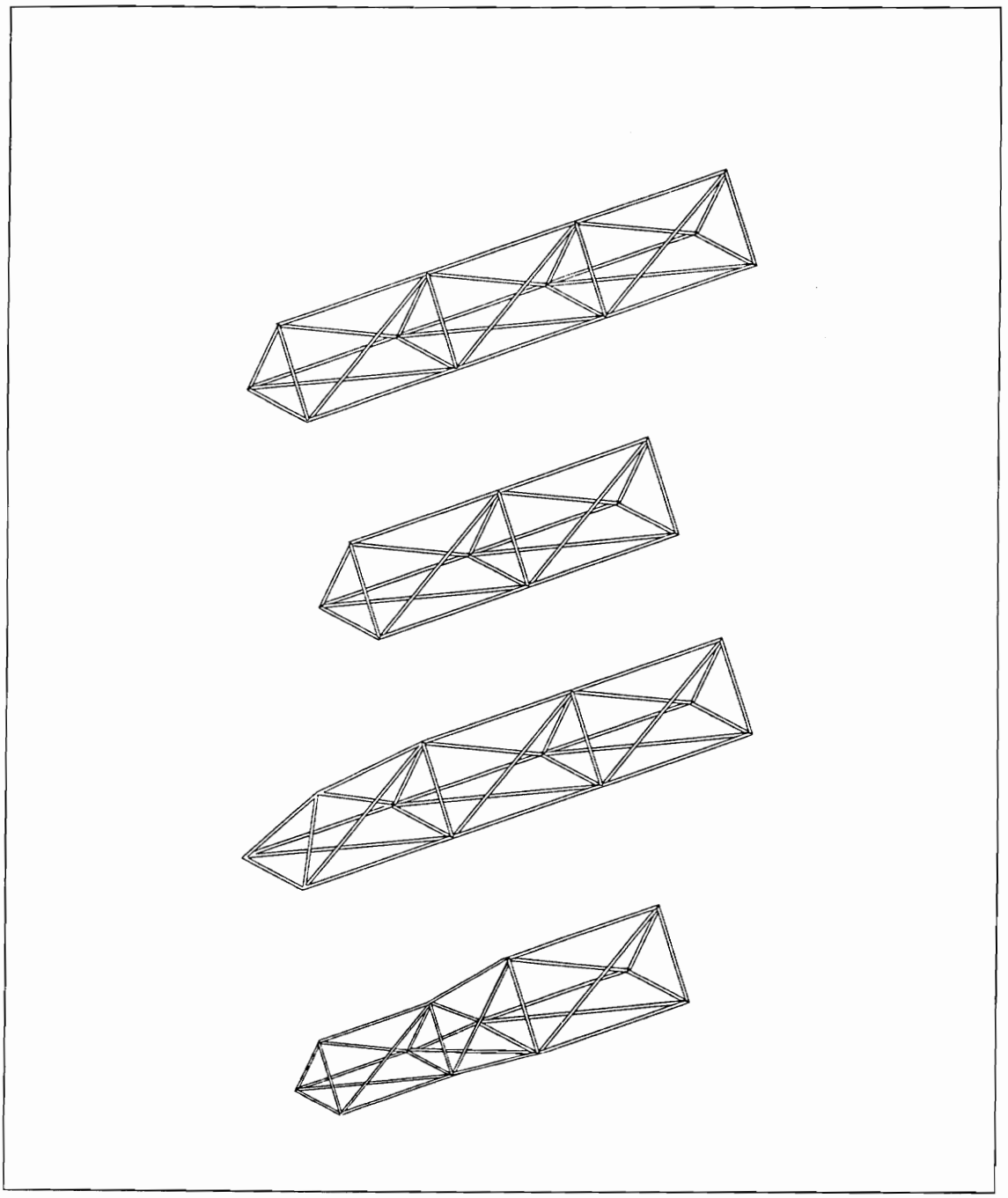


Figure 3.14: Possible Static Truss Sections

Chapter 4

Forward Kinematic Analysis of a Multi-Module, Truss-Type Manipulator

The forward kinematic analyses presented in Chapter 3 provide computational methods for calculating the position and orientation of any truss member with respect to the base frame of its corresponding module. A similar analysis is possible for any number of modules joined together in a predetermined fashion. A primary goal of this chapter will be to express the position and orientation of all members relative to a fixed global reference frame. This single global coordinate system description is imperative for determining the position and orientation of the end of the manipulator. Additionally, the information provided may also be utilized in constructing graphical output or a finite element model.

The forward kinematics algorithms that will be developed in this chapter will be used as the basis for one method of solving the inverse kinematic problem. For this approach, the forward kinematic problem must be solved repetitively to obtain a solution to the inverse problem. The efficiency of these forward solutions is therefore critical. For this reason, the later sections of this chapter will focus on

methods of improving the efficiency of the forward kinematic solution. This gain in efficiency will be obtained by using a canonical input specification set. Effectively, transforming the forward kinematic problem into a closely related linear problem whose solution can be obtained more readily. Once this related problem is solved, the resulting solution may be transformed back into the desired form.

The following sections will outline procedures for the solution of the forward kinematic problem. To serve as an example, the nine degree-of-freedom, long-reach manipulator shown in Fig. 4.1 will be used. This specific manipulator utilizes three batten-actuated, double-octahedral VGT's as the active modules. This geometry was chosen as an example for two reasons. First, the double-octahedral geometry seems to be ideally suited as a joint for a general application long-reach manipulator. The double-octahedral's primary mobility is that of two rotations, so it behaves much like a gimbal or conventional Hooke's coupling. Second, this geometry exhibits a few unique complications not encountered with the tetrahedral or single-octahedron geometries. A thorough treatment of the solution for the double octahedral geometry will encompass all of the same considerations necessary for the other two geometries. Solution method differences for the other geometries will be noted, when necessary, throughout the discussion.

4.1 Guidelines for Chaining Modules Together

One of the primary advantages of a modular manipulator system is that it is possible to configure the manipulator to suit a particular task. This section will briefly introduce some basic concepts on how to logically specify the assembly of modules.

One main consideration when configuring the manipulator for a given task is the number of degrees-of-freedom required. From a reliability standpoint, it

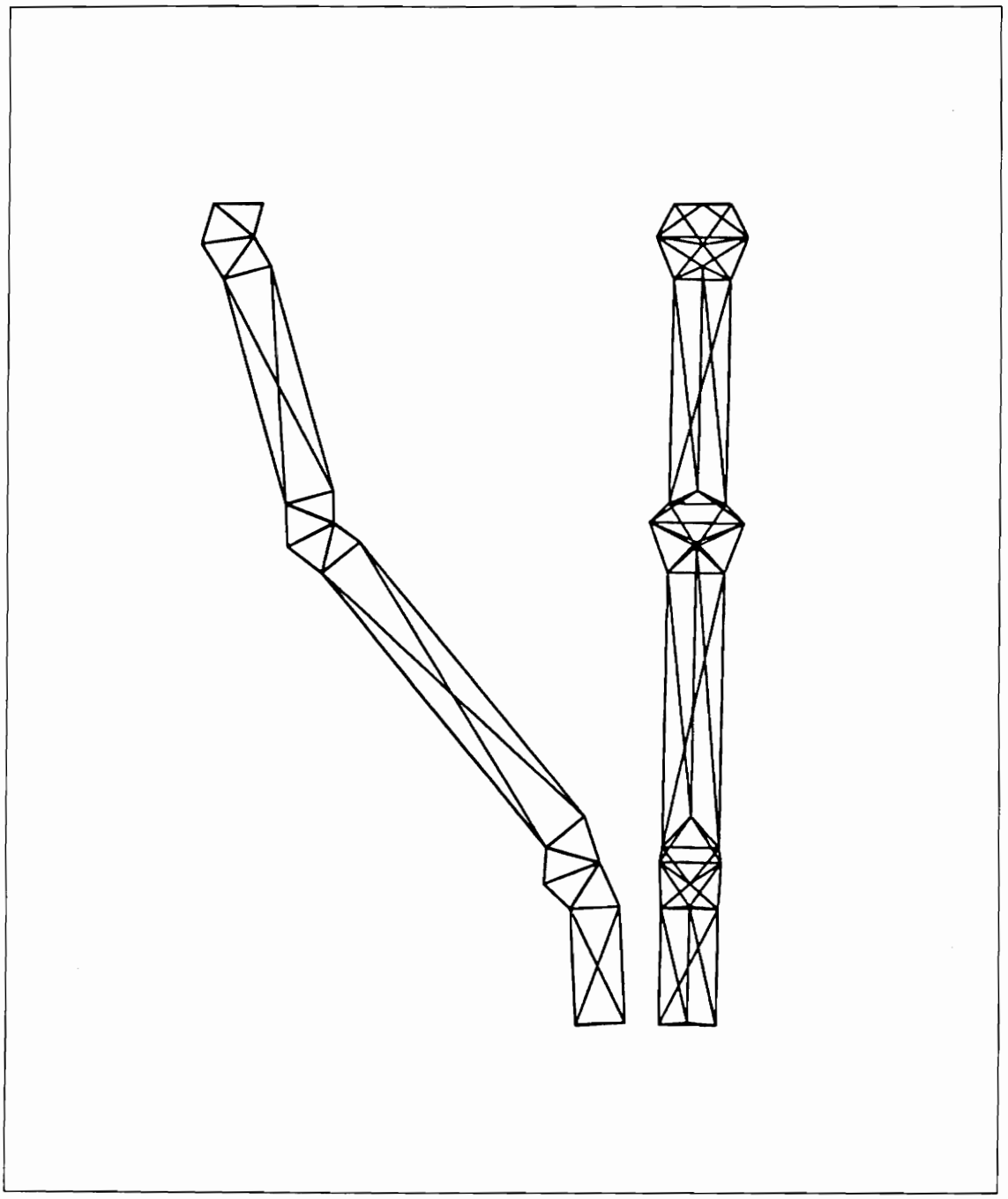


Figure 4.1: Nine Degree-of-Freedom, Long-Reach Manipulator

is generally desirable to minimize the complexity by minimizing the number of degrees-of-freedom present in the system. However, in terms of dexterity, a high number of degrees-of-freedom is favorable. A convenient measure of dexterity is the number of degrees-of-freedom a manipulator possesses in excess of the minimum number required [27]. This is sometimes referred to as the number of degrees-of-redundancy of the manipulator. For example, a four degree-of-freedom planar manipulator, performing a task which requires only three degrees-of-freedom, is said to have a dexterity rating of one, or a single degree-of-redundancy. In general, such a manipulator would be capable of achieving one infinity of different solutions for a specified end position and orientation.

The preceding discussion assumed that high dexterity is desirable for some tasks. Is this a valid assumption? To answer this question it is necessary to observe the environment for which the manipulators were designed.

Non-redundant serial manipulators have proven to be effective for a wide variety of production operations where the task of the manipulator is preplanned and the workspace is relatively free of obstacles. Many of the tasks performed by industrial manipulators require at most six degrees of freedom. For these applications, a highly dexterous manipulator is unnecessary, and only serves to increase the mechanical and computational complexity of the system. In the somewhat unstructured and unknown environment of a waste storage tank, a three degree-of-redundancy manipulator appears to offer realistic flexibility in its ability to achieve solutions in the presence of obstructions. Depending on the complexity of the obstructions, even more degrees-of-freedom may be required and could be utilized.

Another main consideration is the placement and type of joints to be utilized. All modular manipulators can be thought of as distributed actuator systems. The question which naturally arises, is how to efficiently distribute the actuators. A

thorough treatment of this problem should include the development of a parametric model and optimization considerations, which is beyond the scope of this general discussion. However, there are some basic guidelines that will usually result in favorable solutions. These guidelines will be presented below.

As a general rule, it is good practice to cluster several degrees-of-freedom in regions where great dexterity is required and to evenly distribute the degrees-of-freedom elsewhere. Thus, if a high degree of dexterity is required at the end of the manipulator, a six degree-of-freedom, longeron-actuated octahedral module would be an appropriate choice. If only a planar motion is required, a tetrahedral module would suffice. As a reasonable compromise, the examples used in this dissertation are constructed with relatively evenly spaced, batten-actuated, double-octahedral modules.

In addition to considering what type of active modules should be used, the geometries of the static modules must also be selected. Again, this is most effectively resolved by performing an optimization on a parametric model. As a general rule, a good design could be found by assuring that the active joints are near their mid range when the manipulator is in its operating position for a given task. Thus, if it is required that an active joint continually be deflected to one side of its mid range to perform a certain task, the same deflection could be designed into one or both of the adjacent static sections. In short, proper selection of the static sections will assure that the active links can be utilized to the greatest extent possible to produce motions around the anticipated goal positions.

4.2 Notational Convention for Multiple Modules

Before undertaking the forward kinematic analysis, it is first necessary to modify the notation of the previous chapter to accommodate the existence of multiple modules. Figure 4.2 concentrates on the 3^{rd} module of a multi-module truss. The modules are numbered from the base toward the tip. The notation is the same as outlined previously with the exception of the one additional subscript 3 used to denote membership in the 3^{rd} module. It will be assumed that all static length members are of known length. All variable length members will again be identified with a lower case l. Any point of interest on the truss can be described in one of several coordinate frames. For example, the node $n_{5,3}$ is located in coordinate frame $\mathcal{B}_3$ by the vector ${}^{\mathcal{B}_3}\vec{n}_{5,3}$, and it is located in the global coordinate frame, $\mathcal{G}$, by vector ${}^{\mathcal{G}}\vec{n}_{5,3}$.

4.3 Solution of the Conventional Input Forward Kinematics Problem

This section will present a solution method for accomplishing the forward kinematics analysis of a multi-module manipulator. This conventional formulation will assume that the manipulator geometry and the system inputs are known. The system inputs will be those variables that are directly controlled. In this case, the length of the extensible members. Not surprisingly, the formulation of the multi-module, forward solution will rely heavily on the forward kinematic analyses of the individual modules.

The long reach manipulator is essentially built-up by connecting combinations of active and static modules together in an order that will produce a manipulator

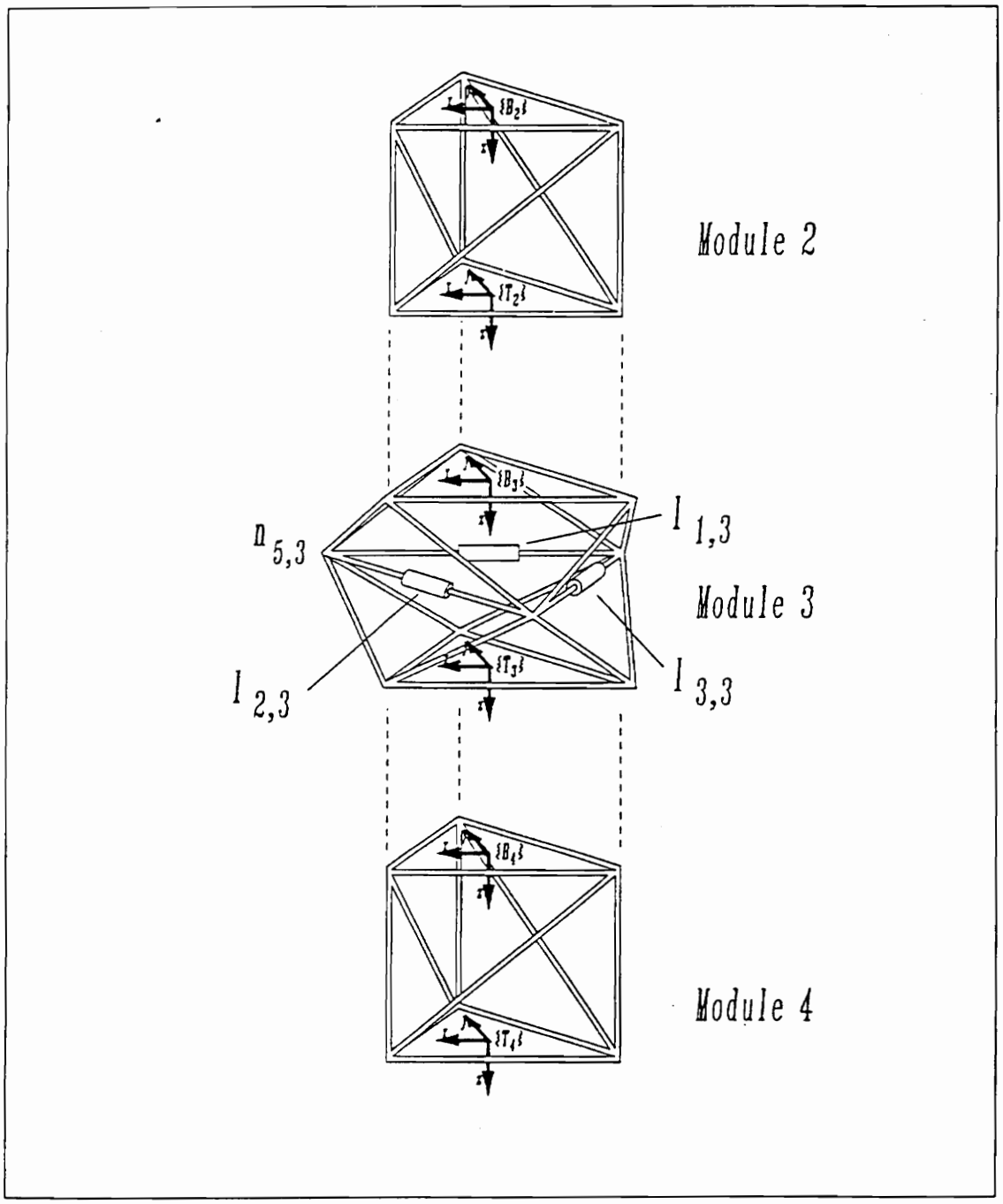


Figure 4.2: 3^{rd} Module of a Multi-Module Truss

appropriate for a specific application. The individual modules are always joined such that they share a common fixed triangular face. The top plane of one module is attached to the base plane of the next.

Referring to Fig. 4.2, note that the individual cells have been joined such that the $\mathcal{T}_i$ coordinate frame is coincident with the $\mathcal{B}_{i+1}$ frame. In the single-module forward kinematic analyses, the relative rotation of frame $\mathcal{T}_i$ described in $\mathcal{B}_i$ could be represented as,

$$\begin{matrix}\mathcal{B}_i \\ \mathcal{T}_i\end{matrix} R = \left[\begin{matrix}\mathcal{B}_i \hat{X}_{\mathcal{T}_i} & \mathcal{B}_i \hat{Y}_{\mathcal{T}_i} & \mathcal{B}_i \hat{Z}_{\mathcal{T}_i}\end{matrix} \right]; \tag{4.1}$$

where $\hat{X}_{\mathcal{T}_i}, \hat{Y}_{\mathcal{T}_i}$ and $\hat{Z}_{\mathcal{T}_i}$ are the unit vectors which form the coordinate axes of frame $\mathcal{T}_i$. It was also found in Chapter 3 these vectors are functions of the variable link lengths of cell i. Again, referring to the i^{th} module of Fig. 4.2 as a typical example, a homogeneous transform which completely describes the position and orientation of frame $\mathcal{B}_{i+1}$ ($\mathcal{B}_{i+1} = \mathcal{T}_i$), in frame $\mathcal{B}_i$ can now be formed as follows:

$$\begin{matrix}\mathcal{B}_i \\ \mathcal{B}_{i+1}\end{matrix} T(l_{j+1}, l_{j+2}, l_{j+3}) = \left[\begin{matrix} \begin{bmatrix}\mathcal{B}_i \\ \mathcal{T}_i\end{bmatrix} R & & {}^{\mathcal{B}_i}\vec{P}_{\mathcal{T}_i} \\ 0 \quad 0 \quad 0 & & 1 \end{matrix} \right]. \tag{4.2}$$

It shall be assumed that the transformation from the base coordinate frame of the first cell to the global coordinate frame is known. Typically this relationship is established either by assignment or by physically measuring the position and orientation of frame $\mathcal{B}_1$ in frame $\mathcal{G}$. In either case, the frame $\mathcal{B}_1$ is rigidly fixed in frame $\mathcal{G}$, which makes $\begin{matrix}\mathcal{G} \\ \mathcal{B}_1\end{matrix} T$ a matrix of constants. Now, any point described in one of the base frames can also be described in the global coordinate frame. For example a point Q, which is known with respect to the k^{th} base frame, can be located in the global coordinate frame as follows:

$${}^{\mathcal{G}}\vec{P}_Q = \begin{bmatrix}\mathcal{G} \\ \mathcal{B}_1\end{bmatrix} T \begin{bmatrix}\mathcal{B}_1 \\ \mathcal{B}_2\end{bmatrix} T \begin{bmatrix}\mathcal{B}_2 \\ \mathcal{B}_3\end{bmatrix} T \ldots\ldots \begin{bmatrix}\mathcal{B}_{k-2} \\ \mathcal{B}_{k-1}\end{bmatrix} T \begin{bmatrix}\mathcal{B}_{k-1} \\ \mathcal{B}_k\end{bmatrix} T \; {}^{\mathcal{B}_k}\vec{P}_Q; \tag{4.3}$$

where each transform in the above expression is a function of the forward kinematics of only one module. Note that the transformations which correspond to the static sections are constants, the others are simply functions of the variable link lengths of the active modules.

4.3.1 Computational Considerations

It is apparent that the forward kinematic solution for each active module will be used repeatedly in forming the coordinate transformations described above. Because of this, it is reasonable to use the single-module forward kinematic analysis as a subroutine in the forward kinematic solution of a multi-module truss. Since the transformations for each active module are dependent only on the extensible link lengths of one module, a more efficient solution could be accomplished by having several dedicated sub-processors which calculate the individual coordinate transformations in parallel. These sub-processors could also "absorb" the constant transformations of the static modules by pre-multiplying these transformations into the variable transformations. This information would then be passed back to the host processor for use in transforming local position descriptions to the global reference frame. This is a trivial advantage in the case of the tetrahedral module which has closed form expressions for the forward kinematics. The time saved by parallel processing is almost entirely wasted on the time it takes to pass the necessary parameters back and forth between processors. However, the octahedral based modules present an entirely different situation. Here, no closed form expressions exist for the conventional specification forward kinematics. Because of the time required to accomplish even a relatively fast iterative evaluation, the parallel processing solution can result in a significant increase in overall solution speed. Even if the canonical input specification is used, parallel processing results in a much quicker solution.

4.3.2 Number of Possible Solutions

In applying the above algorithm to solve the forward kinematics of a multi-module manipulator, an explicit assumption was made concerning the forward kinematic solution for each module. If the solution to the forward kinematics of each active module is limited to one branch, as discussed in Chapter 3, then a unique transformation is obtained for each module. This results in a unique solution for the entire long-chain structure. If, however, the remaining branches of the kinematic solution are deemed acceptable, then there exists many more mathematically correct assemblies of the overall manipulator. For the nine degree-of-freedom manipulator discussed above there are 32,768 possible assemblies. This further emphasizes the need to design the individual modules such that only one physical assembly is possible during operation.

The above result may be generalized to any multi-module truss manipulator with n active modules.

$$number\ of\ solutions \;=\; \prod_{i=1}^{n} N_i, \tag{4.4}$$

where N_i is the number of acceptable solutions for module i. Applying this to the specific modular manipulator described with all possible branches considered acceptable, the total number of solutions is

$$number\ of\ solutions \;=\; 2^{q_1} \times 16^{q_2} \times 32^{q_3} \tag{4.5}$$

where,

q_1 = number of tetrahedral modules
q_2 = number of octahedral modules
q_3 = number of double-octahedral modules

4.4 Solution of the Canonical Input Forward Kinematics Problem

There may be occasions where it is desirable to evaluate the position of the manipulator while not necessarily knowing the true system inputs. In this case, an alternate input specification may prove extremely useful. Again, considering the nine degree-of-freedom manipulator illustrated in Fig 4.1, it can be assumed that the inputs to the system are the nine angles $\phi_{1,1} \ldots \phi_{3,3}$. Thus, for any active module i, a homogeneous transformation which completely describes the position and orientation of frame $\mathcal{B}_{i+1}$ relative to $\mathcal{B}_i$ can be found using the procedures outlined in Chapter 3.

$$ {}^{\mathcal{B}_i}_{\mathcal{B}_{i+1}}T(\phi_{i,1}, \phi_{i,2}, \phi_{i,3}) = \begin{bmatrix} & {}^{\mathcal{B}_i}_{\mathcal{T}_i}R & & {}^{\mathcal{B}_i}\vec{P}_{\mathcal{T}_i} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.6} $$

Any point that is described in one of the base frames can now be described in the global coordinate frame. For example a point Q, which is known with respect to the k^{th} base frame, can be located in the global coordinate frame as follows:

$$ {}^{\mathcal{G}}\vec{P}_Q = \left[{}^{\mathcal{G}}_{\mathcal{B}_1}T\right]\left[{}^{\mathcal{B}_1}_{\mathcal{B}_2}T\right]\left[{}^{\mathcal{B}_2}_{\mathcal{B}_3}T\right] \ldots\ldots \left[{}^{\mathcal{B}_{k-2}}_{\mathcal{B}_{k-1}}T\right]\left[{}^{\mathcal{B}_{k-1}}_{\mathcal{B}_k}T\right] {}^{\mathcal{B}_k}\vec{P}_Q \tag{4.7} $$

Again, each transform in the above expression is a function of the forward kinematics of only one module. If the alternate input set is utilized, the solution to the long-chain forward kinematics problem is closed-form and linear. Using this specification set, only one unique solution is possible for the entire manipulator. A single transformation, which is a function of the nine system inputs ($\phi_{1,1} \ldots \phi_{3,3}$), can now be identified. This matrix is formed by multiplying the individual transformations of the previous equation. Since this description can be found in symbolic form, it is now possible to differentiate the terms of this transfor-

mation to symbolically identify the desired partial derivatives. The chain rule can then be applied, if desired, to convert these partial derivatives to the conventional input specification set.

Using this alternate specification set will substantially decrease computation time. This is especially true if the forward kinematic algorithms are called repetitively for solving the long-chain inverse problem.

Chapter 5

Inverse Kinematic Analysis of a Multi-Module, Truss-Type Manipulator

This chapter will briefly introduce several methods for solving the inverse kinematic problem for the modular, long-reach manipulator system. It was stated earlier that the goal of all kinematic analysis is to develop input/output relationships. In the case of forward kinematics, the goal is to determine the position and orientation of the manipulator (positional output), given the system inputs (either conventional or canonical inputs). The inverse kinematic analysis requires the reverse of this process. For the modular manipulator system described, this inverse process is not as straightforward as it may appear. The main complication that arises is that of input redundancy. In short, it is possible to construct a manipulator system that possesses more actuators than the number of degrees-of-freedom specified in the goal position. Allowing these extra degrees-of-freedom increases the system dexterity, however, it also greatly complicates the solution to the inverse positioning problem. The following sections will discuss some of the terminology necessary for describing these redundant systems and will briefly present a simple planar

example and discuss some solution techniques.

5.1 Characterization of Spaces and Mapping Processes

Before proceeding it is necessary to clarify some of the terms that are often used to describe redundant systems. Unfortunately, there seems to be no universally accepted definitions for some of these terms. Therefore, their definitions, as used within the context of this dissertation, will be presented here. In the following definitions, the term *space* will always refer to an orthogonal curvilinear coordinate space. In each case, the coordinate space will be used to characterize some positional property of the manipulator. This definition of space should not be confused with the rigorous mathematical definition of a vector space.

Input Space This is an n-dimensional space where n represents the total number of inputs. These inputs may be in the form of conventional inputs, in which case the space may also be called the actuator space or joint space, or alternatively, the inputs may be in the form of canonical inputs. Note that the actuator space is generally linearly related to the manipulator joint space. In the case of parallel manipulators, a point in the conventional input space may correspond to several different configurations of the manipulator. However, a point in the canonical input space always corresponds to a unique configuration of the manipulator. The input space will be bounded by the physical constraints of the actuator limitations.

Goal Space This is an m-dimensional space where m represents the total number of parameters defined in the goal specification. In conventional robotics, m is generally limited to six degrees-of-freedom (the total number of parameters

required to position and orient a free-floating object in space). However, for the redundant manipulators described, it is possible to specify more than six parameters in a goal specification. For example, it may be desirable to specify the position and orientation of the end-effector and the position of an intermediate module. This would require a nine degree-of-freedom goal space. In general, a point in the goal space may correspond to more than one configuration of the manipulator.

Solution Space The solution space is a subspace of the input space that contains all possible inputs that achieve a particular desired goal. Although this description may utilize all variables of the input space, it will be shown that that this space can always be characterized by at most an $n - m$ dimensional space. Such a solution space is said to yield on the order of $n - m$ infinities of solutions. In the event that m is greater than n, the problem is overconstrained and no solution generally exists.

It can be observed that the input space is solely a function of the given manipulator design. Conversely, the goal space is only dependent on the task to be accomplished. The solution space unifies both the manipulator and task and hence is dependent not only on the manipulator design but also on the specified goal position.

With the above definitions established, the forward and inverse kinematic analyses can now be conceptualized as a mapping operation. As illustrated in Fig. 5.1-A and Fig. 5.1-B, the forward kinematic analysis can be represented as a mapping operation which transforms points in the input space into points in the goal space. Likewise, the inverse kinematic analysis maps points in the goal space into regions of the input space. The dimension of this region is dependent on the degree-of-redundancy of the manipulator.

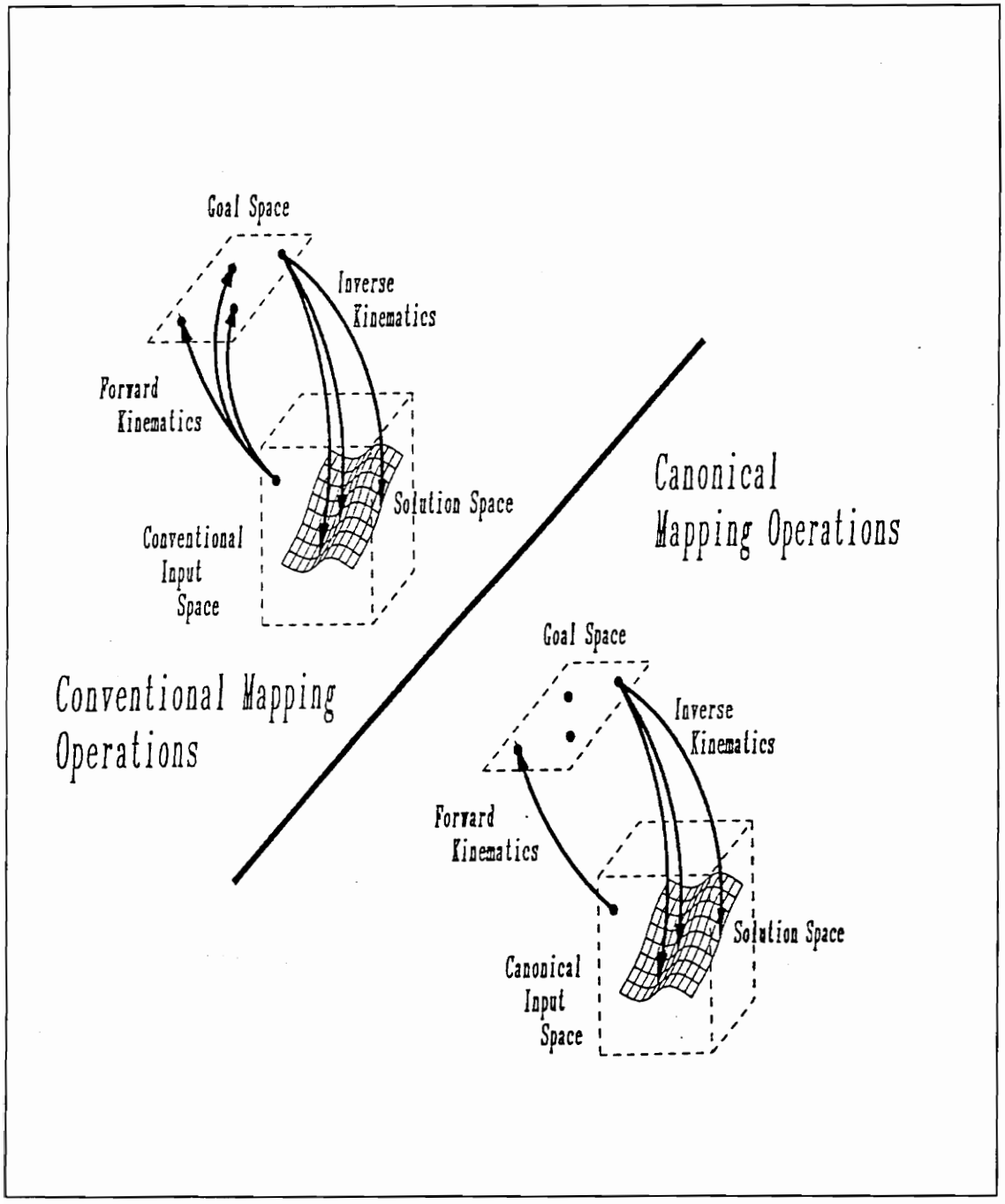


Figure 5.1: Mapping Representation of Forward and Inverse Kinematics

5.1.1 Example: A Redundant Planar Manipulator

A simple planar three degree-of-freedom manipulator is shown in Fig. 5.2. For this example, the desired goal position will be limited to an X and Y position only. The described planar manipulator has a three-dimensional input space and a two dimensional goal space. This manipulator therefore possesses a single degree-of-redundancy and has a one-dimensional solution space that represents all possible solutions to a given goal position. In general, the inverse kinematic problem can be expected to have on the order of one infinity of possible solutions.

As shown in Fig. 5.3, the input space of this manipulator will be represented by a three-dimensional orthogonal space, where each axis of the input space corresponds to one of the input parameters (θ_1, θ_2, or θ_3). The input space will be bounded by the actual limits of each input parameter. In the absence of joint limits, this input range will be from 0 to 360 degrees. This restriction simply prevents the generation of physically identical solutions every 360 degrees of input. The goal space of this manipulator will also be represented by an orthogonal space. In this case, one of two dimensions corresponding to the X and Y values of all possible goal positions. Any point in the input space can be directly transformed, or mapped, into a unique point in the goal space as follows:

$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (5.1)$$

$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad (5.2)$$

This mapping operation is the forward kinematic solution. Now, reversing the process, a single point in the goal space maps to a a set of curves in the input space. One curve corresponding to each branch of the inverse solution.

To better illustrate this concept, a specific example will be worked. Consider the planar manipulator discussed previously with design values selected as follows:

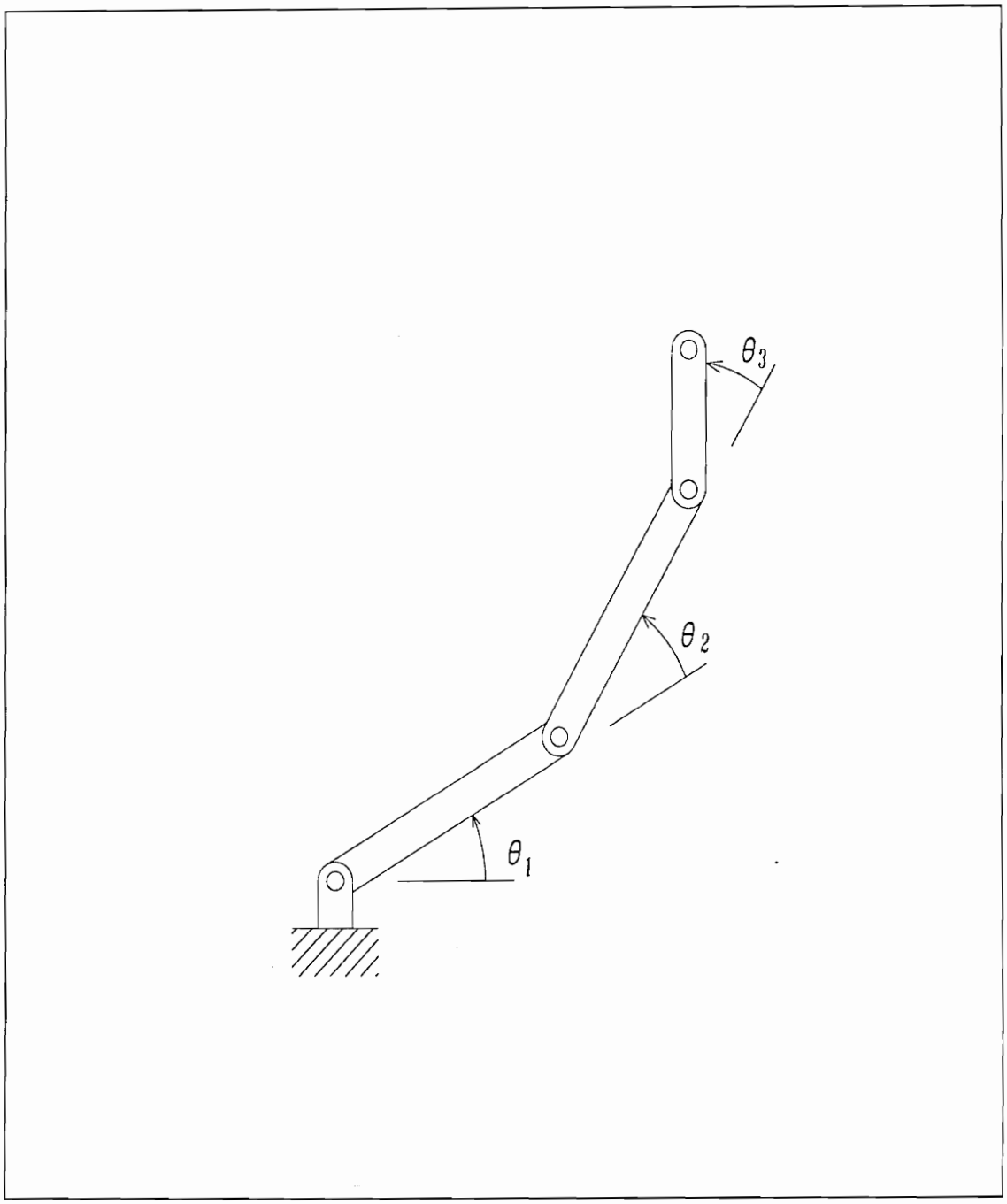


Figure 5.2: A Simple Planar Redundant Manipulator

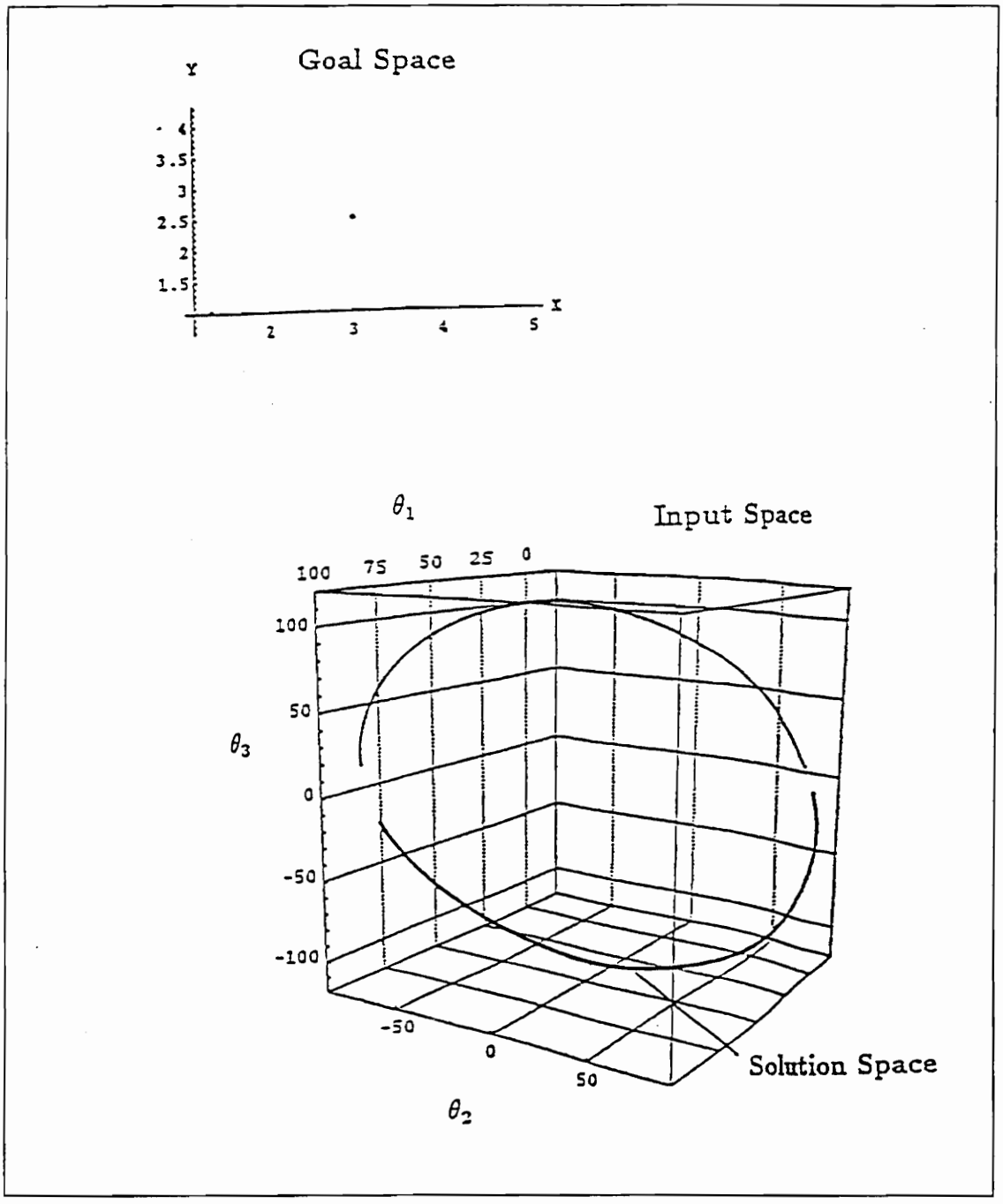


Figure 5.3: Mapping Example: The Planar Three Degree-of-Freedom Manipulator

$L_1 = L_2 = 2.0$ units, $L_3 = 1.0$ units, and θ_1, θ_2, and θ_3 may vary from 0 to 360 degrees. For this example the desired goal position will be $x = 2.5$ units, $y = 2.5$ units. This is consistent with Fig. 5.3. The two curves superimposed on the input space of Fig. 5.3 represent all possible solutions to the inverse kinematic problem. The solid curve represents solutions in branch A and the dashed curve represents all solutions in branch B.

Figure 5.4 illustrates the physical interpretation of these two curves. Again, the solid depiction is used for branch A and dashed for B. It is possible that the two curves could join and, in some cases, fuse to form a single curve. The intersection of the branch solution spaces can only occur at an interior singularity. It is also possible that the solution space curves for a certain goal position may degenerate to a point. This occurs at boundary singularities.

5.2 Selection of a Solution

Clearly, it is undesirable to present an operator of a manipulator system with all the possible solutions to a redundant inverse kinematic problem. The alternatives are then limited to presenting the operator with the one best solution or several good solutions. The following sections will present two different methods for selecting appropriate solutions to the redundant inverse problem.

5.2.1 Null Space Optimization Methods

In the previous section, it was shown that there exists a region of solutions to a redundant inverse kinematic problem. Chapter 6 will develop methods for efficiently evaluating solutions within the solution space so that a local optimum solution may be selected which maximizes or minimizes some desired objective function. Briefly, this optimization procedure will progress as follows. Suppose that a feasi-

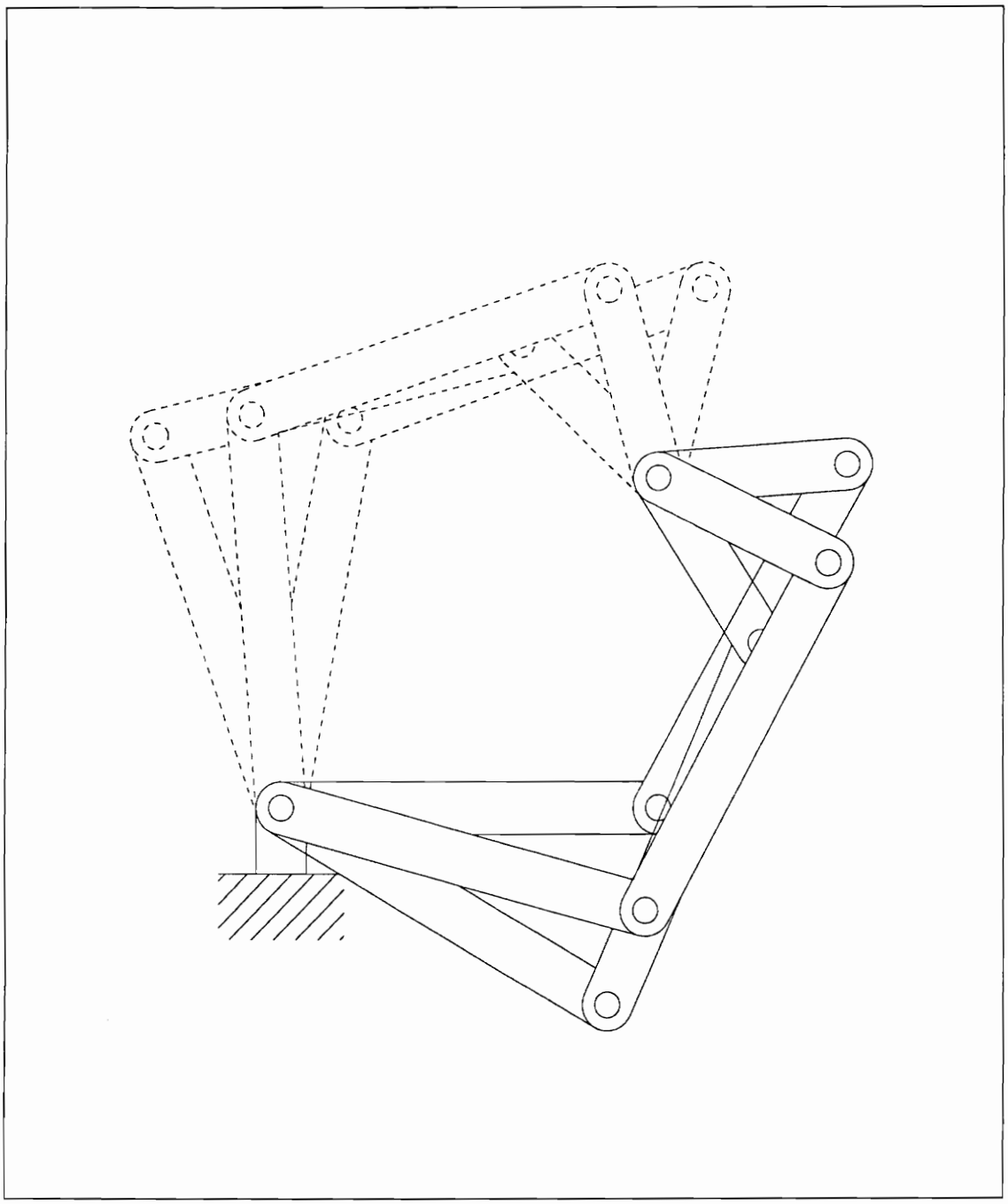


Figure 5.4: Some Possible Solutions to the Given Inverse Kinematic Problem

ble solution to an inverse kinematic problem has been found. By definition, this solution must lie within the solution space of this inverse problem. The task at hand is to identify if any other solutions within a finite neighborhood of the current solution achieves a better objective function value. It is easy to identify a finite neighborhood of points in the input space. However, a neighboring point in the input space may no longer satisfy the original positioning problem. In contrast, a neighboring point in the solution space must, by definition, satisfy the original problem. How can neighboring points in the more restrictive solution space be easily identified? What is required is a means for quantifying the directions in the input space that accurately represent the solution space within a small neighborhood. It is not necessary to have a global definition of the solution space to achieve a local minimum, only a local definition of the solution space. This linearized, local model of the solution space will be called the manipulator null space for reasons that will become apparent in Chapter 6.

5.2.2 Order Reduction Methods

Another, perhaps more obvious, approach to this problem is to simply reduce the dimension of the solution space. It appears that the difficulty in generating solutions is that there is no one unique solution, and the operator, or algorithm, simply has too many choices. This method effectively eliminates some of these choices. The goal is to apply this method in a way that primarily eliminates the "bad" feasible solutions, while preserving the "good" solutions.

At its crudest level this order reduction consists of more or less arbitrarily limiting or fixing some of the systems degrees-of-freedom. The philosophy here is that since it is a redundant manipulator system, these extra degrees-of-freedom are not really necessary. On the basis of ease of implementation there is certainly merit to the solution. However, for many real world problems, including the waste removal

manipulator, this method is not practical. It simply eliminates the extra degrees-of-freedom. However, it is these extra degrees-of-freedom that made a redundant manipulator desirable. The goal was to improve dexterity. If this dexterity is not required for a task, a non-redundant manipulator would be a better choice.

At a more refined level, order reduction methods have shown great promise for quickly arriving at near optimal solutions for very high degree-of-freedom systems. The concept of shape control, a form of order reduction, has been shown to produce favorable solutions to some thirty and sixty degree-of-freedom systems in real time. Some of the concepts of shape control will be applied to the positional control of the modular, long-reach manipulator in Chapter 7.

Chapter 6

Null Space Optimization Techniques

This chapter will introduce null space optimization techniques that afford an efficient method for searching through and selecting an optimal positioning solution for a redundant manipulator. In Chapter 5 the null space of the manipulator was described as a local, linearized model of the solution space. This model is obtained by examining the *null space* of the manipulator's Jacobian matrix. The following sections will review some properties of the manipulator Jacobian matrix, including evaluation procedures for determining the Jacobian matrix. The focus will then shift to a numerical analysis of the Jacobian to extract information about the current manipulator position and all neighboring positions within the solution space for a specified problem.

6.1 The Jacobian Matrix

The Jacobian matrix for a manipulator represents the effect of the individual system inputs on the system outputs. In short, it is a matrix of the partial derivatives of the system outputs with respect to the system inputs. Again, the system inputs

could be of either conventional or canonical form. In all cases, the system output will be considered the position of the manipulator in terms of the goal space parameters. Thus, if the end-effector orientation and X position is used to define the goal space, then these same four parameters would also be considered the system outputs. If the system inputs are represented by a vector $\vec{\tau}$ in the input space and the system outputs are represented by a vector $\vec{\chi}$ in the goal space, then the following equation may be used to symbolically represent the functional dependence of the system outputs on the system inputs.

$$\vec{\chi} = \mathcal{F}(\vec{\tau}) \tag{6.1}$$

In equation 6.1 the symbol $\mathcal{F}$ essentially represents the relationship between input and output as described by the forward kinematic analysis. Often this functional relationship is highly non-linear and difficult, if not impossible, to express in closed-form.

For a redundant system with n inputs and m outputs the Jacobian matrix may now be defined as follows:

$$\mathcal{J}(\vec{\tau}) = \begin{bmatrix} \frac{\partial \mathcal{F}_1}{\partial \tau_1} & \frac{\partial \mathcal{F}_1}{\partial \tau_2} & \frac{\partial \mathcal{F}_1}{\partial \tau_3} & \cdots & \frac{\partial \mathcal{F}_1}{\partial \tau_n} \\ \frac{\partial \mathcal{F}_2}{\partial \tau_1} & \frac{\partial \mathcal{F}_2}{\partial \tau_2} & \frac{\partial \mathcal{F}_2}{\partial \tau_3} & \cdots & \frac{\partial \mathcal{F}_2}{\partial \tau_n} \\ \frac{\partial \mathcal{F}_3}{\partial \tau_1} & \frac{\partial \mathcal{F}_3}{\partial \tau_2} & \frac{\partial \mathcal{F}_3}{\partial \tau_3} & \cdots & \frac{\partial \mathcal{F}_3}{\partial \tau_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathcal{F}_m}{\partial \tau_1} & \frac{\partial \mathcal{F}_m}{\partial \tau_2} & \frac{\partial \mathcal{F}_m}{\partial \tau_3} & \cdots & \frac{\partial \mathcal{F}_m}{\partial \tau_n} \end{bmatrix} \tag{6.2}$$

This $m \times n$ Jacobian matrix can now be used to generate an expression for the rate of change of the outputs given a certain rate of change of the inputs. This expression is valid only for the set of inputs, $\vec{\tau}$ for which the Jacobian was found.

$$\dot{\vec{\chi}} = \mathcal{J}(\vec{\tau})\ \dot{\vec{\tau}} \tag{6.3}$$

For a non-redundant manipulator in a non-singular configuration, it is possible to determine the rate of change of the inputs that will produce a specified output velocity by inverting $\mathcal{J}(\vec{\tau})$ and proceeding as follows:

$$\dot{\vec{\tau}} = \mathcal{J}^{-1}(\vec{\tau})\ \dot{\vec{\chi}} \tag{6.4}$$

This inversion process, however, is not possible for the non-square Jacobian matrix of a redundant manipulator.

6.1.1 Evaluation of the Jacobian Matrix

The null space optimization method relies on repetitively evaluating the Jacobian matrix. Because of this, it is important to establish a method for rapid formulation of the Jacobian matrix. The following sections will present two method for determining the Jacobian matrix of a given manipulator. These two methods were selected because they are particularly well suited to evaluating the Jacobian of the proposed modular, long-reach manipulator system. To serve as a simplified example for this discussion, both methods will be applied to the planar redundant manipulator described in Chapter 5.

The Differential Forward Approach

The differential forward approach is perhaps the most broad reaching method for evaluating the Jacobian matrix of a given manipulator. This numerical method may be applied to any manipulator system. A closed-form solution to the forward kinematics problem is not required. However, it is advantageous to have the fastest

possible solution to the forward kinematic problem. This often means that the canonical formulation of the forward kinematic solution should be utilized.

The Jacobian matrix is composed of the partial derivatives of the system outputs with respect to the system inputs. These partial derivatives may be evaluated as follows:

$$\frac{\partial \mathcal{F}_i}{\partial \tau_j} = \lim_{\epsilon \to 0} \frac{\mathcal{F}_i(\vec{\tau} + \epsilon \vec{e}_j) - \mathcal{F}_i(\vec{\tau})}{\epsilon} \tag{6.5}$$

where $\vec{e}_j$ is an n dimensional column vector containing all zeros except for its j^{th} element, which is unity.

Note that equation 6.5 can be used to solve for all m parameters of $\mathcal{F}$ with only one solution of the forward kinematics. To completely evaluate the Jacobian therefore requires $n + 1$ different calls of the forward kinematics routine. One call is required to determine $\mathcal{F}(\vec{\tau})$, and n more calls are required to find $\mathcal{F}(\vec{\tau} + \epsilon \vec{e}_j)$ for $j = 1 \ldots n$.

This method will now be applied to the planar manipulator example of Chapter 5. Again, the geometry will be specified as $L_1 = L_2 = 2$ and $L_3 = 1$. The position for which the Jacobian will be evaluated is $\theta_1 = 0°$, $\theta_2 = 45°$, and $\theta_3 = 30°$. For this manipulator there are two relationships which define the forward kinematics. For this particular case, both are closed-form expressions.

$$\mathcal{F}_x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \tag{6.6}$$

$$\mathcal{F}_y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \tag{6.7}$$

To determine the complete Jacobian matrix requires substituting the appropriately adjusted θ's into these equations a total of four times. Accomplishing this substitution with $\epsilon = 10^{-6}$ results in the following 2×3 Jacobian matrix.

$$\mathcal{J}([0^o, 45^o, 30^o]^T) = \begin{bmatrix} -2.3801 & -2.3801 & -0.9659 \\ 3.6730 & 1.6730 & 0.2588 \end{bmatrix} \tag{6.8}$$

The X and Y velocities of the manipulator end-effector can now be found for any joint velocities as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathcal{J}([0^o, 45^o, 30^o]^T) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \tag{6.9}$$

This method provides the only possible scheme for evaluating the Jacobian matrix of a manipulator for which the forward kinematics is not known in closed form. For example, this method must be utilized to find the Jacobian of a longeron-actuated, octahedral VGT if the conventional input specification of $l_1 \ldots l_6$ is used.

6.1.2 Closed-Form Evaluations

For manipulators in which the forward kinematics can be expressed in closed form, a more attractive solution can sometimes be obtained by direct differentiation. The word sometimes has been used because not all closed-form solutions yield differentiable equations. Some closed-form solutions are highly non-linear and may contain operations where, for example, a positive value is used in some circumstances and a negative value is used in others. Yet, these solutions are closed-form. Returning to the example of the planar manipulator, equations 6.6 and 6.7 are clearly differentiable with respect to the input variables. Differentiating yields the following Jacobian matrix:

$$\mathcal{J}(\vec{\theta}) = \begin{bmatrix} -L_1 S\theta_1 - L_2 S\theta_{12} - L_3 S\theta_{123} & -L_2 S\theta_{12} - L_3 S\theta_{123} & -L_3 S\theta_{123} \\ L_1 C\theta_1 + L_2 C\theta_{12} + L_3 C\theta_{123} & L_2 C\theta_{12} + L_3 C\theta_{123} & L_3 C\theta_{123} \end{bmatrix} \tag{6.10}$$

where $C\theta = \cos\theta$, $S\theta = \sin\theta$, and $\theta_{ijk} = \theta_i + \theta_j + \theta_k$. Substituting in the position $\theta_1 = 0°$, $\theta_2 = 45°$, $\theta_3 = 30°$ yields:

$$\mathcal{J}([0°, 45°, 30°]^T) = \begin{bmatrix} -2.3801 & -2.3801 & -0.9659 \\ 3.6730 & 1.6730 & 0.2588 \end{bmatrix} \tag{6.11}$$

6.2 The Jacobian Null Space

This section will introduce the Jacobian null space as a useful concept for identifying other nearby solutions to a redundant inverse kinematic problem.

Equation 6.3 relates the rate of change of the system inputs to the rate of change of the system outputs. For convenience this equation is repeated here:

$$\dot{\vec{\chi}} = \mathcal{J}(\vec{\tau})\, \dot{\vec{\tau}} \tag{6.12}$$

It is also possible to write a similar equation in terms of small motions instead of instantaneous velocities.

$$\Delta\vec{\chi} = \mathcal{J}(\vec{\tau})\, \Delta\vec{\tau} \tag{6.13}$$

Suppose that the manipulator is currently at a feasible solution (i.e., a point in the solution space for a given problem). Further, suppose that the manipulator is producing no change of its output values. In this situation the following homogeneous equation may be written:

$$\mathbf{0} = \mathcal{J}(\vec{\tau})\, \Delta\vec{\tau} \tag{6.14}$$

If $\mathcal{J}(\vec{\tau})$ were non-singular and therefore invertible, there would be only one trivial solution to equation 6.14; $\Delta\vec{\tau} = \mathbf{0}$. In short, there could be no motion of the system inputs without resulting in a motion of the system outputs.

For redundant manipulators, $\mathcal{J}(\vec{\tau})$ is generally singular and equation 6.14 does have other non-trivial solutions. Since $\mathcal{J}(\vec{\tau})$ has more columns than rows, it is impossible for all columns to be linearly independent. Therefore there must exist some linear combinations of the columns that have no effect on the system outputs. It is possible then, to adjust the system input values in such a way as to produce no changes in the system outputs. Because these adjustments of the system inputs did not alter the system outputs, the new point in the input space must also lie within the solution space of the manipulator. The input space directions in which these adjustments take place will be called null space directions. All possible null space directions define the Jacobian matrix null space (or kernel) $\mathcal{N}(\mathcal{J})$ [38, page 468]. This represents all possible solutions to the homogeneous velocity equation given in equation 6.14. Likewise, it is possible to find the range space of the Jacobian matrix, $\mathcal{R}(\mathcal{J})$, which identifies all possible directions of the output vector. Note that, in most situations, the range space of the Jacobian is of little consequence. It will only be of value when the manipulator is in a singular configuration. In that special case it will provide information about how the system outputs are coupled or otherwise constrained.

The dimension of the Jacobian matrix null space is called the nullity of the Jacobian matrix, $\nu(\mathcal{J})$. In general the nullity of the Jacobian matrix is calculated as:

$$\nu(\mathcal{J}) = n - \rho(\mathcal{J}) \tag{6.15}$$

where $\rho(\mathcal{J})$ is the rank of the Jacobian matrix. The rank is simply the dimension of the Jacobian matrix range space. Note that in the absence of other column degeneracies, the nullity of the Jacobian should be $n - m$. This assumes that the manipulator is not in a singular configuration where it becomes impossible for the

manipulator to control all m outputs simultaneously. At these singular positions the null space dimension increases proportionally to the number of new column dependencies in the Jacobian.

A convenient way of enumerating all possible directions of the null space is to identify a set of null space basis vectors which are linearly independent and span the null space. Thus, if the nullity of a Jacobian matrix is one, then only one possible null space basis vector may be identified. If the nullity is four, then four linearly independent null space basis vectors are required to span the null space.

Figure 6.1 presents a graphical representation of the solution space, the associated null space, and two possible null space vectors for the planar manipulator described earlier.

The following section will present computational methods for determining the Jacobian matrix nullity, rank, null space, and range space. In addition, this section will provide some general insights into determining the relative closeness to a singularity configuration.

6.2.1 Computational Methods for Determining the Null Space Basis

Most methods for determining the null space or range space are derived from orthogonal factorization methods. In short, the goal of these methods is to identify and extract the orthogonal components present in the columns of the Jacobian matrix. A natural way to identify these orthogonal components is to decompose the given Jacobian matrix into a set of orthogonal matrices whose product is equal to the original matrix. Two of these orthogonal factorization methods will be presented in the following sections: Q-R factorization and Singular Value Decomposition (SVD). The intent of the following sections will be to illustrate possible interpretations of the algorithms' results rather than to present the theory be-

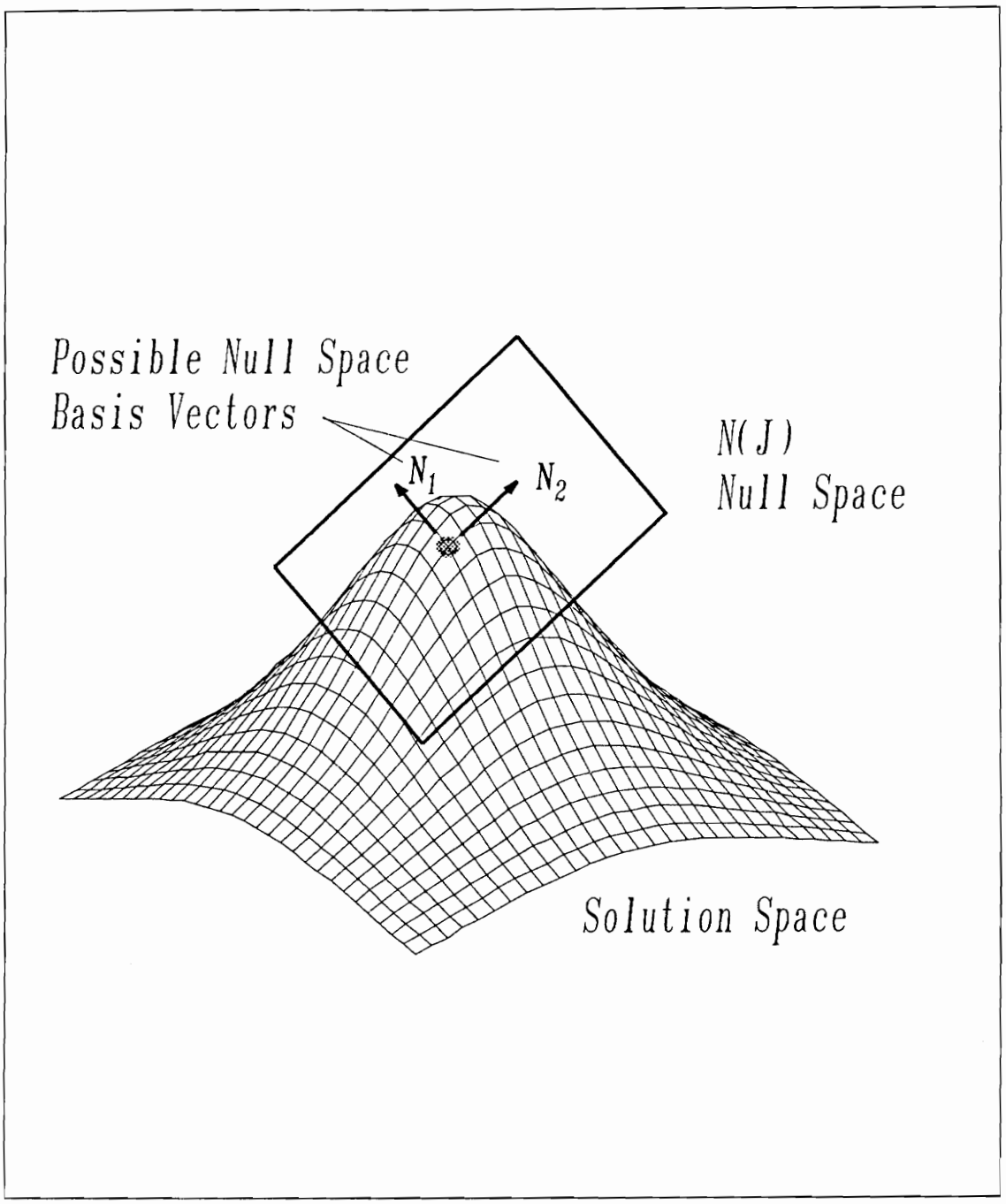


Figure 6.1: The Manipulator Solution Space, Null Space, and Null Space Vectors

hind factorization methods. Both factorization methods will provide the greatest amount of information if the Jacobian matrix is augmented with sufficient rows of zeros to result in a square matrix. The system output vector, $\vec{\chi}$, must also be augmented accordingly. Adding these extra rows adds the constraint that $0 = 0$, and thus preserves the original system of equations. When discussing the following procedures, it will be assumed that the original matrix is an augmented, square Jacobian matrix of dimension $n \times n$.

Q-R Factorization

Q-R factorization is a combination of two methods independently known as the Householder and Gram-Schmidt methods [57]. Q-R factorization provides a fast and relatively robust method for determining the rank, range space, nullity, and null space of a given matrix.

Q-R factorization is based on the following result of linear algebra. It is possible to factor a matrix, $\mathbf{A}$, into a real orthonormal matrix, $\mathbf{Q}_1$ and an upper triangular matrix, $\mathbf{R}_1$, as follows:

$$\mathbf{A} = \mathbf{Q}_1\mathbf{R}_1 \tag{6.16}$$

Many numerical analysis packages such as MATLAB [43] and LINPACK [13] contain routines for performing Q-R factorization. A good discussion of the concepts involved in Q-R factorization are contained in Golub and VanLoan [16, pages 146-153] and Stoer and Bulirsch [57, pages 389-400].

After performing Q-R factorization, the rank of matrix $\mathbf{A}$, $\rho(\mathbf{A})$, may be found as the number of non-zero diagonal elements of $\mathbf{R}_1$. A set of orthonormal range space basis vectors can then be identified as the first ρ columns of $\mathbf{Q}_1$. These ρ vectors span the same space as the columns of $\mathbf{A}$.

The nullity of matrix $\mathbf{A}$, $\nu(\mathbf{A})$, is equal to the number of columns of $\mathbf{A}$ minus the rank of $\mathbf{A}$, or,

$$\nu(\mathbf{A}) = n - \rho(\mathbf{A}) \tag{6.17}$$

Still more information about $\mathbf{A}$ can be extracted by performing Q-R factorization on the transpose of $\mathbf{A}$ as follows:

$$\mathbf{A}^T = \mathbf{Q}_2\mathbf{R}_2 \tag{6.18}$$

The nullity of the original matrix $\mathbf{A}$, $\nu(\mathbf{A})$, is equal to the number of all zero columns of $\mathbf{R}_2$. In addition, a set of null space basis vectors may now be identified from the last ν columns of $\mathbf{Q}_2$.

Returning to the planar manipulator of Section 6.1.1, where the augmented Jacobian matrix for a given position has been identified as

$$\mathcal{J} = \begin{bmatrix} -2.3801 & -2.3801 & -0.9659 \\ 3.6730 & 1.6730 & 0.2588 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \tag{6.19}$$

This Jacobian matrix can be decomposed using Q-R factorization into

$$\mathbf{Q}_1 = \begin{bmatrix} -0.5438 & -0.8392 & 0.0000 \\ 0.8392 & -0.5438 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \tag{6.20}$$

$$\mathbf{R}_1 = \begin{bmatrix} 4.3768 & 2.6984 & 0.7425 \\ 0.0000 & 1.0876 & 0.6699 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \tag{6.21}$$

Thus, the rank of this Jacobian is two, and one possible set of range space vectors are

$$\vec{\mathcal{R}}_1 = \begin{bmatrix} -0.5438 \\ 0.8392 \\ 0.0000 \end{bmatrix} \qquad \vec{\mathcal{R}}_2 = \begin{bmatrix} -0.8392 \\ -0.5438 \\ 0.0000 \end{bmatrix} \tag{6.22}$$

As mentioned previously, the range space of the Jacobian is of interest only in singular configurations. For this reason, the Q-R factorization of the Jacobian will only be computed when the nullity is found to be greater than $n - m$. Whereas, the Q-R factorization of the Jacobian transpose will always be computed. Decomposing $\mathcal{J}^T$ yields,

$$\mathbf{Q}_2 = \begin{bmatrix} -0.6797 & 0.7121 & 0.1761 \\ -0.6797 & -0.5211 & -0.5162 \\ -0.2758 & -0.4705 & 0.8382 \end{bmatrix} \tag{6.23}$$

$$\mathbf{R}_2 = \begin{bmatrix} 3.5019 & -3.7050 & 0.0000 \\ 0.0000 & 1.6218 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \tag{6.24}$$

Now it can be observed that the nullity of the Jacobian matrix is one, and a null space basis vector is represented by

$$\vec{\mathcal{N}}_1 = \begin{bmatrix} 0.1761 \\ -0.5162 \\ 0.8382 \end{bmatrix} \tag{6.25}$$

This null space vector can be verified by substitution into equation 6.14.

$$\mathcal{J} \cdot \vec{\mathcal{N}}_1 = \begin{bmatrix} -2.3801 & -2.3801 & -0.9659 \\ 3.6730 & 1.6730 & 0.2588 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \begin{bmatrix} 0.1761 \\ -0.5162 \\ 0.8382 \end{bmatrix} = \begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix} \tag{6.26}$$

Figure 6.2 presents the effects of adjusting the present input values by moving in the direction of the null space vector as follows:

$$\vec{\theta} = \vec{\theta}_c + \delta \cdot \vec{\mathcal{N}}_1 \tag{6.27}$$

Where $\vec{\theta}_c$ represents the "current" position of the manipulator. Note that for even relatively large values of δ the goal position (X,Y) does not appreciably change.

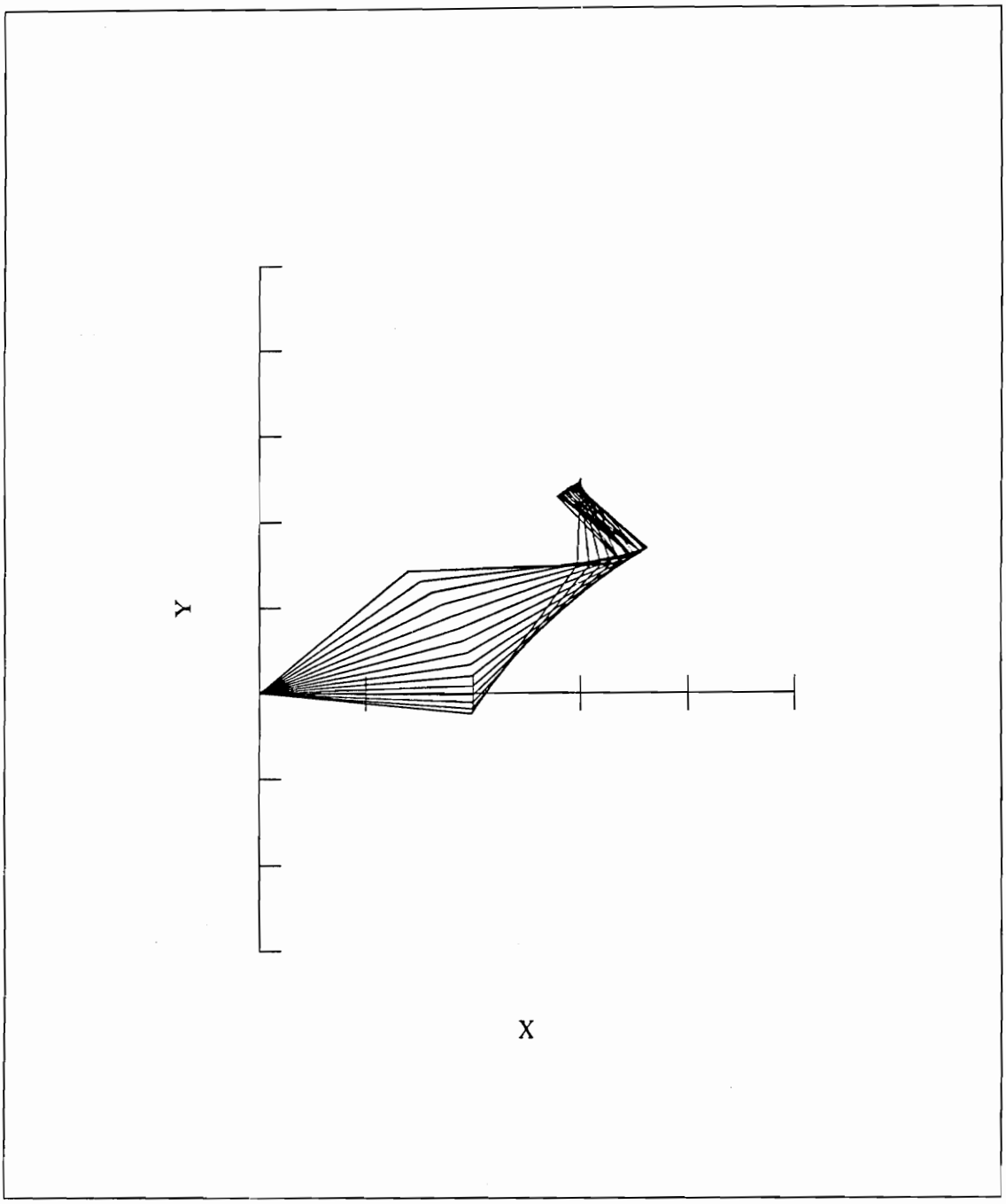


Figure 6.2: Effects of Moving Along a Null Space Direction

In many instances Q-R factorization might be favored over singular value decomposition methods. The primary reason for this is that the singular value decomposition method requires 5 to 10 times the number of arithmetic computations to decompose the Jacobian [16] [21, page 218]. However, singular value decomposition does present the user with valuable additional information and always results in realistic solutions. There are instances where the Jacobian matrix may become so ill conditioned that the Q-R factorization fails to provide meaningful results. As the next section will show, singular value decomposition also provides an efficient method for obtaining the pseudo-inverse of the Jacobian matrix. This pseudo-inverse will be valuable for obtaining and maintaining feasible solutions within the solution space of a given inverse positioning problem.

Singular Value Decomposition

It can be argued that singular value decomposition is the most sophisticated of the orthogonal decomposition methods. The foundations of singular value decomposition date back to the mid 1870's when Beltrami and Jordan independently discovered that such a decomposition was possible [21, pages 218-220]. This early work was limited to square matrices and remained somewhat of an impractical procedure until the advent of modern computing facilities. Much credit for the evolution of practical singular value decomposition methods must be given to Golub [16] who demonstrated their feasibility for a wide range of applications, most notably, the solution of linear least-squares curve fitting problems. Although this represents an over-constrained system (ie., one with more rows than columns) it is closely related to the under-constrained problem which is of interest in identifying the Jacobian null space. Other excellent discussions of singular value decomposition methods are contained in Nakamura [34], Press, et al. [44], and Maciejewski [28].

Singular value decomposition is based on the premise that it is possible to decompose any matrix $\mathbf{A}$ in the following manner:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T \tag{6.28}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices and Σ is a diagonal matrix. If $\mathbf{A}$ is a square $n \times n$ matrix (an augmented Jacobian matrix), then $\mathbf{U}$, Σ, and $\mathbf{V}$ must also be of dimension $n \times n$.

The non-zero elements of Σ are referred to as the singular values of $\mathbf{A}$. These are values of λ that satisfy the following two equations:

$$\mathbf{A}\vec{u} = \lambda\vec{v} \qquad\qquad \mathbf{A}^H\vec{v} = \lambda\vec{u} \tag{6.29}$$

where $\mathbf{A}^H$ is the Hermitian transpose of $\mathbf{A}$. Since this application is only concerned with the singular value decomposition of the Jacobian matrix, which must be real valued, $\mathbf{A}^H = \mathbf{A}^T$. Equation 6.29 can be therefore be rewritten as:

$$\begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \vec{u} \\ \vec{v} \end{bmatrix} = \lambda \begin{bmatrix} \vec{u} \\ \vec{v} \end{bmatrix} \tag{6.30}$$

This is essentially the definition of an eigenvalue, λ, with a corresponding eigenvector $[\vec{u}, \vec{v}]^T$. The terminology of calling these eigenvalues singular values stems from the fact that the matrix

$$\mathcal{S} = \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix} - \lambda\mathbf{I} \tag{6.31}$$

becomes singular at these values of λ.

Now to interpret the results of singular value decomposition. Performing singular value decomposition on the augmented Jacobian matrix immediately provides a great deal of information about the manipulator system. The rank of the Jacobian, ρ, is equal to the number of non-zero diagonal elements of Σ. The nullity of

the Jacobian matrix, ν, is equal to the number of diagonal elements of $\mathbf{\Sigma}$ which are zero.

The range space of the Jacobian is identified from the first ρ columns of $\mathbf{U}$, and the null space of the Jacobian is found from the last ν columns of $\mathbf{V}$. This is the same information that was afforded by the Q-R factorization methods.

Further information is obtained by observing the relative magnitude of the singular values. The ratio of the largest singular value, σ_l, to the smallest non-zero singular value, σ_s, provides a matrix condition number, κ.

$$\kappa = \frac{\sigma_l}{\sigma_s} \tag{6.32}$$

The σ values are the diagonal elements of $\mathbf{\Sigma}$. A small condition number indicates a well-conditioned Jacobian matrix. A large condition number indicates an ill-conditioned Jacobian matrix. When the manipulator approaches a singular configuration, the condition number will approach infinity. This can provide the positional control program with a great deal of insight as to the functioning of the manipulator. For example, if, while moving the manipulator, the condition number becomes too large, ($\approx 10^5$), the positional control program may forcibly set the smallest singular values to zero. This increases the dimension of the null space and correspondingly decreases the range space dimension. The range space basis vectors should then be evaluated to pinpoint which output degrees-of-freedom have been lost. The positional control program should then inhibit position specifications which move in the "lost" directions. By voluntarily surrendering these degrees of freedom before a true singularity is encountered, the control program has avoided specifying positions for which a large numerical error could occur. Sacrificing these small singular values is justified because computationally the system of equations has become so sensitive that small round-off errors corrupt the entire

solution. In addition these small singular values correspond to directions of the range space that produce little effect on the system outputs (i.e., they are nearly zero and thus nearly correspond to null space directions, which produce no effect on the output).

It was also mentioned earlier that singular value decomposition provides an easy way of evaluating the pseudo-inverse of a singular matrix. Consider Equation 6.3 again:

$$\mathcal{J}(\vec{r})\ \dot{\vec{r}} = \dot{\vec{\chi}} \tag{6.33}$$

Substituting the singular value decomposition representation of $\mathcal{J}(\vec{r})$ yields:

$$\left[\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right]\ \dot{\vec{r}} = \dot{\vec{\chi}} \tag{6.34}$$

Since $\mathbf{U}$ and $\mathbf{V}$ are orthonormal and $\mathbf{\Sigma}$ is diagonal, Equation 6.34 can be rewritten as:

$$\dot{\vec{r}} = \left[\mathbf{V}\mathbf{\Sigma}^*\mathbf{U}^T\right]\ \dot{\vec{\chi}} \tag{6.35}$$

where $\mathbf{\Sigma}^*$ is a diagonal matrix whose elements are $\frac{1}{\sigma_i}$ for all non-zero elements of $\mathbf{\Sigma}$, and zero for all other elements. $\left[\mathbf{V}\mathbf{\Sigma}^*\mathbf{U}^T\right]$ represents the pseudo-inverse of the Jacobian matrix, $\mathcal{J}^\sharp(\vec{r})$. Multiplying out Equation 6.35 results in the particular solution for $\dot{\vec{r}}$ which minimizes, in a least squared sense, the magnitude of $\dot{\vec{r}}$. The null space basis vectors, when combined with this particular solution, span the solution space of the inverse velocity problem. Equation 6.35 could also be written for small displacements of $\vec{\chi}$ as:

$$\Delta\vec{r} = \left[\mathbf{V}\mathbf{\Sigma}^*\mathbf{U}^T\right]\ \Delta\vec{\chi} \tag{6.36}$$

where again, the pseudo-inverse identifies the the particular solution that minimizes $\Delta\vec{r}$.

Returning to the planar manipulator of Section 6.1.1, where the augmented Jacobian matrix has been identified as

$$\mathcal{J} = \begin{bmatrix} -2.3801 & -2.3801 & -0.9659 \\ 3.6730 & 1.6730 & 0.2588 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \tag{6.37}$$

This Jacobian matrix can now be decomposed using singular value decomposition. This results in

$$\mathbf{U} = \begin{bmatrix} -0.6497 & -0.7602 & 0.0000 \\ 0.7602 & -0.6497 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \tag{6.38}$$

$$\mathbf{\Sigma} = \begin{bmatrix} 5.2388 & 0.0000 & 0.0000 \\ 0.0000 & 1.0841 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \tag{6.39}$$

$$\mathbf{V} = \begin{bmatrix} 0.8282 & -0.5321 & 0.1761 \\ 0.5379 & 0.6664 & -0.5162 \\ 0.1573 & 0.5222 & 0.8382 \end{bmatrix} \tag{6.40}$$

From these results it can be observed that the rank of the Jacobian is two, and the nullity is one. A possible set of range space basis vectors is

$$\mathcal{R}_1 = \begin{bmatrix} -0.6497 \\ 0.7602 \\ 0.0000 \end{bmatrix} \qquad \mathcal{R}_2 = \begin{bmatrix} -0.7602 \\ -0.6497 \\ 0.0000 \end{bmatrix} \tag{6.41}$$

and, the null space basis vector (the null space direction) is

$$\mathcal{N}_1 = \begin{bmatrix} 0.1761 \\ -0.5162 \\ 0.8382 \end{bmatrix} \tag{6.42}$$

Note that since there is only one null space basis vector, it must be unique.

Furthermore, it is now possible to observe the condition number of the Jacobian and have some relative measure of how close the current position is to a singularity. For this position the condition number is calculated as:

$$\kappa = \frac{\sigma_l}{\sigma_s} = \frac{5.2388}{1.0841} = 4.8324 \tag{6.43}$$

Thus the current position is relatively far away from any singularities that might jeopardize the manipulators ability to simultaneously control both the X and Y position of the end-effector.

Lastly, the pseudo-inverse of the Jacobian can be computed as in Equation 6.35.

$$\mathcal{J}^{\sharp} = \begin{bmatrix} 0.2704 & 0.4391 & 0.0000 \\ -0.5340 & -0.3213 & 0.0000 \\ -0.3857 & -0.2901 & 0.0000 \end{bmatrix} \tag{6.44}$$

Calculating this pseudo-inverse is of great importance to identifying an initial configuration within the solution space, as the next section will illustrate.

6.3 Example: Null Space Optimization Method Applied a Redundant Planar Manipulator

This section will seek to implement the concepts developed thus far in this dissertation. To serve as a simple example, the null space optimization method will first be applied to the planar manipulator described previously. For this example, the desired goal position will be $X = 3.0$, $Y = 2.5$. It will be assumed that the manipulator is currently in an arbitrary position described by the input vector $\vec{\tau}_c = [0^o, 0^o, 0^o]^T$. Substituting these values into the forward kinematic expressions of Equations 6.6 and 6.7, yields an augmented output vector of $\vec{\chi}_c = [5, 0, 0]^T$. The subscript c will always denote that these are the "current" values of the corresponding variables. Likewise the subscript d will be used to indicate "desired"

values.

Recall that the null space evaluation methods require that the initial position lie within the solution space of a given inverse problem. Therefore, before the null space optimization procedures can begin, the system must first converge to a feasible solution. That is to say, *any* solution where the end-effector is at the desired goal position, $\vec{\chi}_d = [3.0, 2.5, 0.0]^T$. The first observation to be made is that the manipulator is not yet at the desired position. The error in the manipulators position is

$$\Delta\vec{\chi} = \vec{\chi}_d - \vec{\chi}_c = \begin{bmatrix} -2.0 \\ 2.5 \\ 0.0 \end{bmatrix} \tag{6.45}$$

Starting from the current position, calculate the Jacobian matrix using one of the methods of in Section 6.1.1 . This results in

$$\mathcal{J}(\vec{\tau}_c) = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 5.0 & 3.0 & 1.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \tag{6.46}$$

Performing singular value decomposition on this Jacobian yields

$$\mathbf{U} = \begin{bmatrix} 0.0000 & -1.0000 & 0.0000 \\ 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \tag{6.47}$$

$$\mathbf{\Sigma} = \begin{bmatrix} 5.9161 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \tag{6.48}$$

$$\mathbf{V} = \begin{bmatrix} 0.8452 & -0.5345 & 0.0000 \\ 0.5071 & 0.8018 & -0.3162 \\ 0.1690 & 0.2673 & 0.9487 \end{bmatrix} \tag{6.49}$$

It is immediately evident that the manipulator is currently in a singular configuration. The rank of the Jacobian is one, the nullity is two, and the range space is therefore limited to one dimension, described by the following basis vector:

$$\mathcal{R}_1 = \begin{bmatrix} 0.0000 \\ 1.0000 \\ 0.0000 \end{bmatrix} \tag{6.50}$$

This manipulator, in the current configuration has no ability to control motions along the X axis. Taking the pseudo-inverse of the Jacobian yields:

$$\mathcal{J}^{\sharp}(\vec{\tau}_c) = \begin{bmatrix} 0.0000 & 0.1429 & 0.0000 \\ 0.0000 & 0.0857 & 0.0000 \\ 0.0000 & 0.0286 & 0.0000 \end{bmatrix} \tag{6.51}$$

Substituting this into Equation 6.35 yields the particular solution,

$$\Delta\vec{\tau}_p = \mathcal{J}^{\sharp}\Delta\vec{\chi} = \begin{bmatrix} 0.3571 \\ 0.2143 \\ 0.0714 \end{bmatrix} radians \tag{6.52}$$

This particular solution can now be used to update the current values of $\vec{\tau}_c$ as follows:

$$\vec{\tau}_c = \vec{\tau}_c + \Delta\vec{\tau}_p \tag{6.53}$$

This entire process is repeated until a feasible solution is obtained. The feasible solution is identified as one which satisfies:

$$\Delta\chi_i \leq \mu_i \qquad\qquad i = 1 \ldots n \tag{6.54}$$

where $\vec{\mu}$ is a convergence tolerance vector that specifies the maximum acceptable error for each output parameter.

Figure 6.3 illustrates convergence using this procedure. The resulting feasible

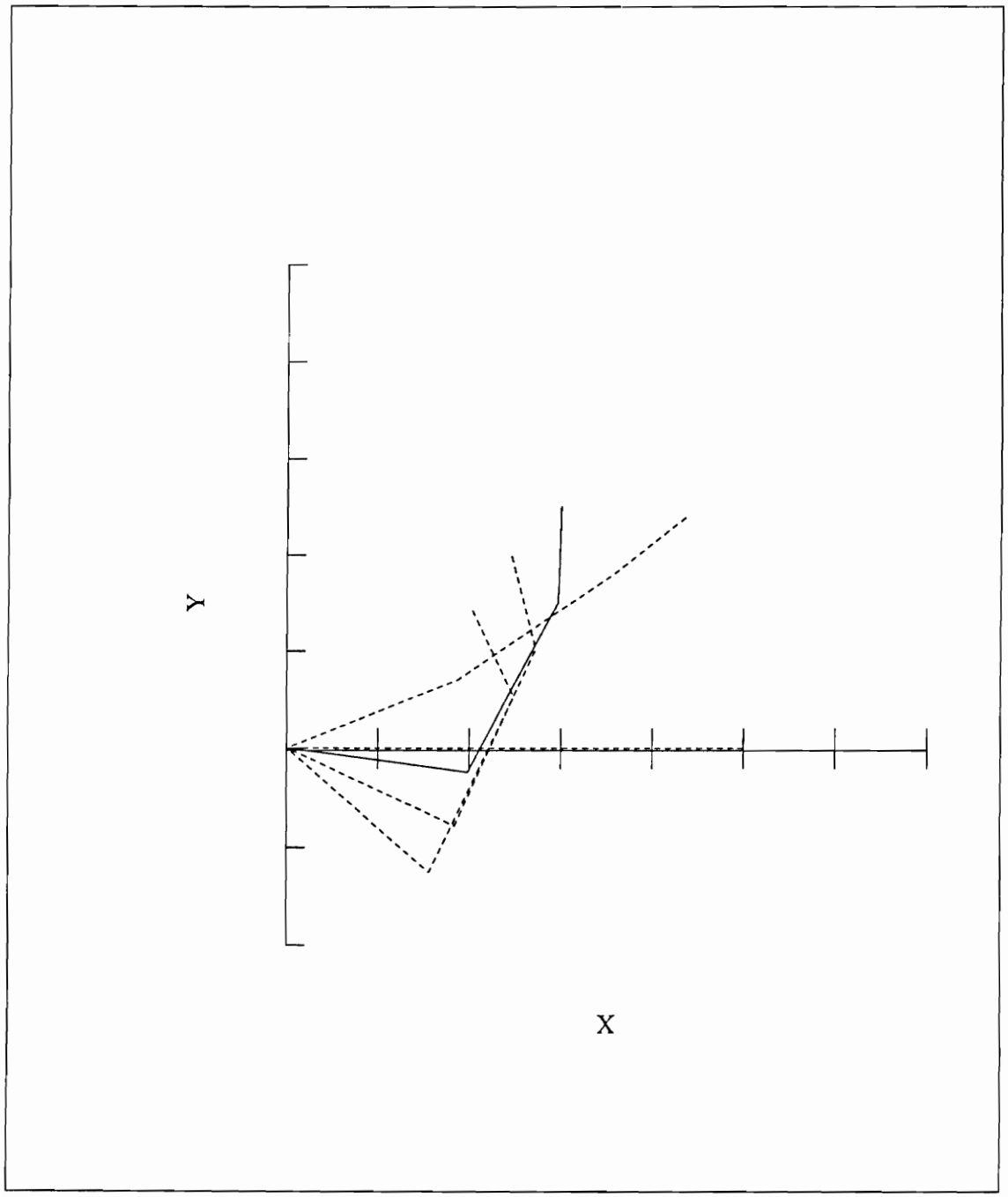


Figure 6.3: Convergence From an Arbitrary Initial Position To a Feasible Solution

solution was obtained in four iterations and is represented with solid lines. The input values for the manipulator in this position are

$$\vec{\tau} = \begin{bmatrix} -6.92^o \\ 67.42^o \\ 27.82^o \end{bmatrix} \tag{6.55}$$

Now that a feasible solution has been obtained, the null space optimization procedure can begin. For this example, the condition number of the Jacobian will be selected as the objective function. Any scalar quantity, or combinations of scalar quantities, could serve as the objective function. The following algorithm will seek a solution with a minimum condition number, thus ensuring that the final solution is a maximum "distance" away from any singularities. Because the condition number, κ, must be computed for this choice of objective function, singular value decomposition will be used to identify the null space instead of Q-R factorization.

The Jacobian matrix at the above feasible position can now be evaluated. It is

$$\mathcal{J}([-6.9^o, 67.4^o, 27.8^o]^T) = \begin{bmatrix} -2.4851 & -2.7546 & -1.0000 \\ 2.9392 & 0.9575 & -0.0023 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \tag{6.56}$$

Performing singular value decomposition on this Jacobian yields:

$$\mathbf{U} = \begin{bmatrix} -0.7917 & -0.6110 & 0.0000 \\ 0.6110 & -0.7917 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \tag{6.57}$$

$$\mathbf{\Sigma} = \begin{bmatrix} 4.7365 & 0.0000 & 0.0000 \\ 0.0000 & 1.3729 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \tag{6.58}$$

$$\mathbf{V} = \begin{bmatrix} 0.7945 & -0.5889 & 0.1482 \\ 0.5839 & 0.6738 & -0.4529 \\ 0.1668 & 0.4463 & 0.8792 \end{bmatrix} \tag{6.59}$$

As expected, there exists only one null space direction for this configuration. It is

$$\vec{\mathcal{N}}_1 = \begin{bmatrix} 0.1482 \\ -0.4529 \\ 0.8792 \end{bmatrix} \tag{6.60}$$

Also note that the objective function for this configuration can be extracted from $\boldsymbol{\Sigma}$ as

$$\kappa(\vec{\tau}) = \frac{4.7365}{1.3729} = 3.45 \tag{6.61}$$

The objective function for all adjacent configurations within the solution space of the problem can now be evaluated with the following equation:

$$\kappa(\vec{\tau} + \delta \cdot \vec{\mathcal{N}}_1) \qquad\qquad \delta = \pm\epsilon \tag{6.62}$$

where epsilon represents a small increment. If either a positive or negative value of δ results in a more favorable evaluation of the objective function, the current value of $\vec{\tau}$ should be modified by moving in the appropriate direction of the null space vector. When both positive and negative values of δ fail to yield improved solutions, the manipulator is in a local optimum configuration. To preserve feasibility, the forward kinematic solution should be checked each time $\vec{\tau}$ is incremented. Anytime $\Delta\vec{\chi}$ violates the restrictions imposed by the tolerance vector, $\vec{\mu}$, the system must be reconverged.

Figure 6.4 illustrates the null space optimization procedure progressing from the initial feasible solution to a final optimum configuration. The vectors drawn at the tip of the manipulator illustrate the effect of varying each joint independently while in the optimum configuration. The final optimum value for the objective function is $\kappa = 3.39$. Note from Fig. 6.4 that each input produces different motions of the

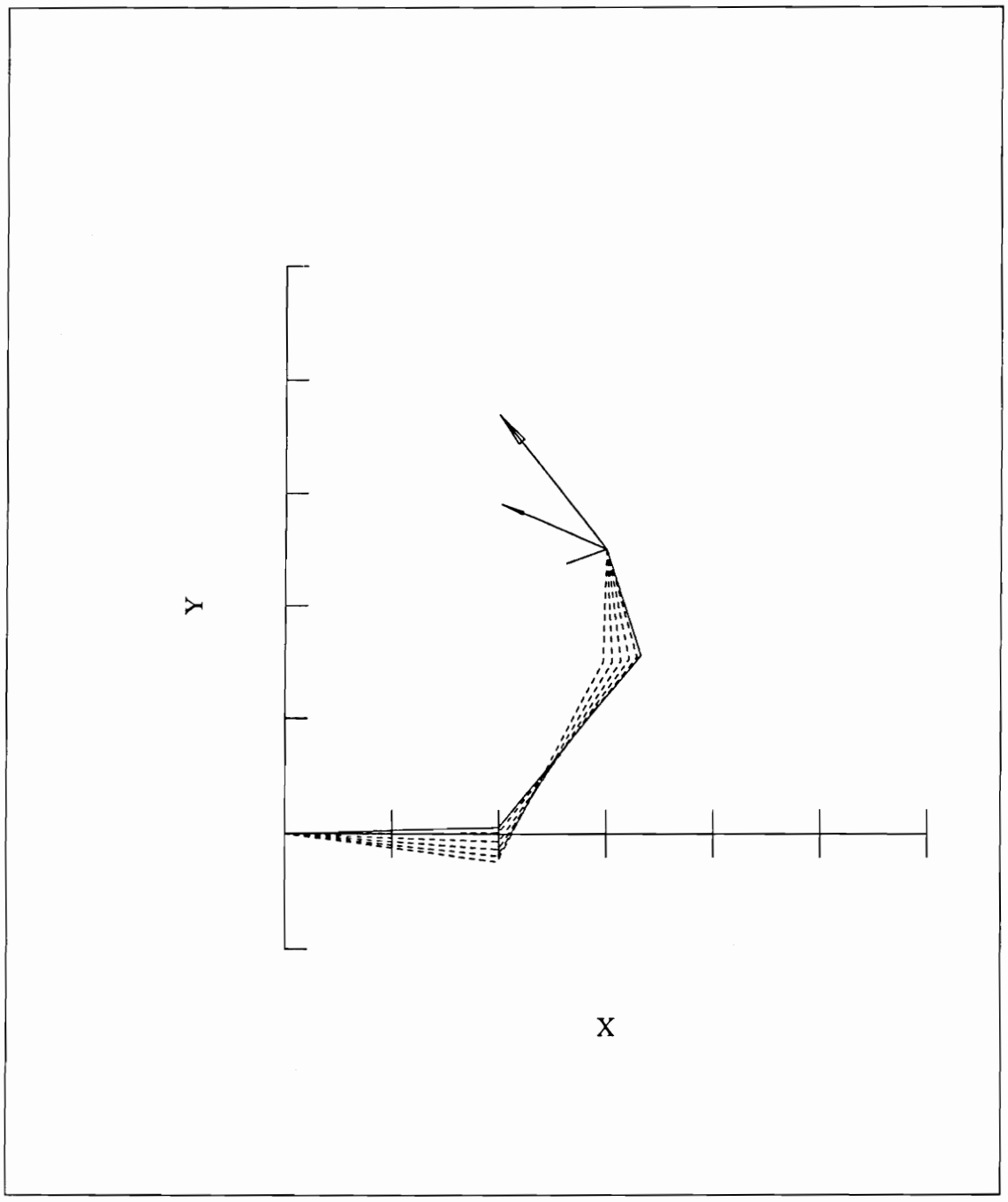


Figure 6.4: Optimization within the Null Space; Minimum κ

tip. In fact, θ_1 and θ_3 produce motions at right angles to each other. Thus, a motion can easily be produced in any direction. In contrast to this, if the above procedure is executed to maximize the condition number, Fig. 6.5 results. Here $\kappa = 4.46$ is the optimum value. Note that each joint now produces motions which are primarily vertical. Thus, it is more difficult, in this configuration, to produce a horizontal motion.

In this limited example, the null space was one dimensional. This same procedure is easily extended to problems with multi-dimensional null spaces. The one modification that is necessary is that the objective function should be evaluated in each null space direction as follows:

$$\mathcal{O}_i\left(\vec{\tau} + \delta \cdot \vec{\mathcal{N}}_i\right); \qquad \delta = \pm\epsilon \tag{6.63}$$

where $\vec{\tau}$ is the current solution, and ϵ is a small step increment. Note that it is necessary to evaluate both directions (positive and negative) for each null space vector. After these evaluations the solution should proceed in the direction of the most improving objective function.

The initial convergence to a feasible solution and the initial null space optimization can be time consuming. However, once an initial optimum solution is found convergence and optimization for a nearby goal position occur within a few iterations. Furthermore, for small motions of the specified goal position, the convergence and optimization routines can run simultaneously. This is done by using the particular solution, as generated by the pseudo-inverse, in combination with the optimum homogeneous solution, as indicated by the null space evaluation.

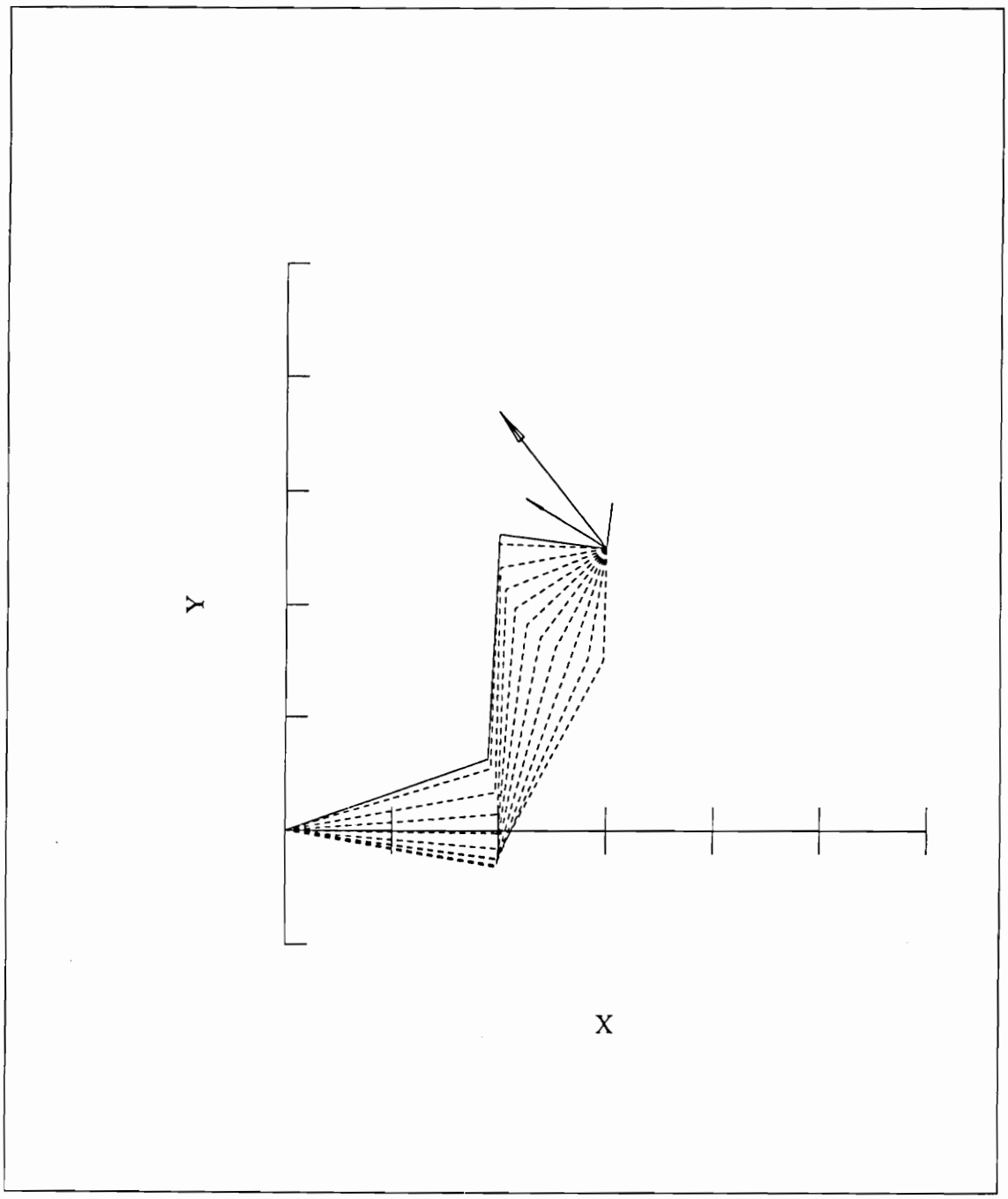


Figure 6.5: Optimization within the Null Space; Maximum κ

6.4 Computational Efficiency

It is clear that in order to achieve positional control in real-time, the null space optimization method must produce solutions efficiently. The following results were obtained by programming in Microsoft "C" version 6.00 on an IBM compatible personal computer. The null space optimization method was verified experimentally by applying null space techniques on the nine degree-of-freedom modular, long-reach manipulator shown in Fig. 6.6. Incrementing from an optimal position to a new, nearby goal position, required on average two iterations, and in extreme cases near singularities as many as six iterations. By far the largest portion of the computational burden is involved in computing the singular value decomposition. To compute the number of floating point operations (FLOPs) required for any manipulator system the following equation can be used [16]:

$$FLOP\ COUNT = 2m^2n + 11n^3 \tag{6.64}$$

where n is the number of degrees-of-freedom of the manipulator system and m is generally the number of degrees-of-freedom of the goal space. At singularities the number of singular values identified by singular value decomposition decreases and m decreases proportionately. The second column of Table 6.1 presents the the maximum number of floating point operations required to compute the singular value decomposition of a manipulator of n degrees-of-freedom. This information was generated using the assumption that the goal specification contains six degrees-of-freedom. The dimension of the goal space could potentially be greater. The third column is an estimate of the number of calculations required to perform one iteration of the optimum positioning routine. The fourth column presents the number of computations required to perform four iterations. Note that, in general, the solutions will be generated with fewer than four iterations.

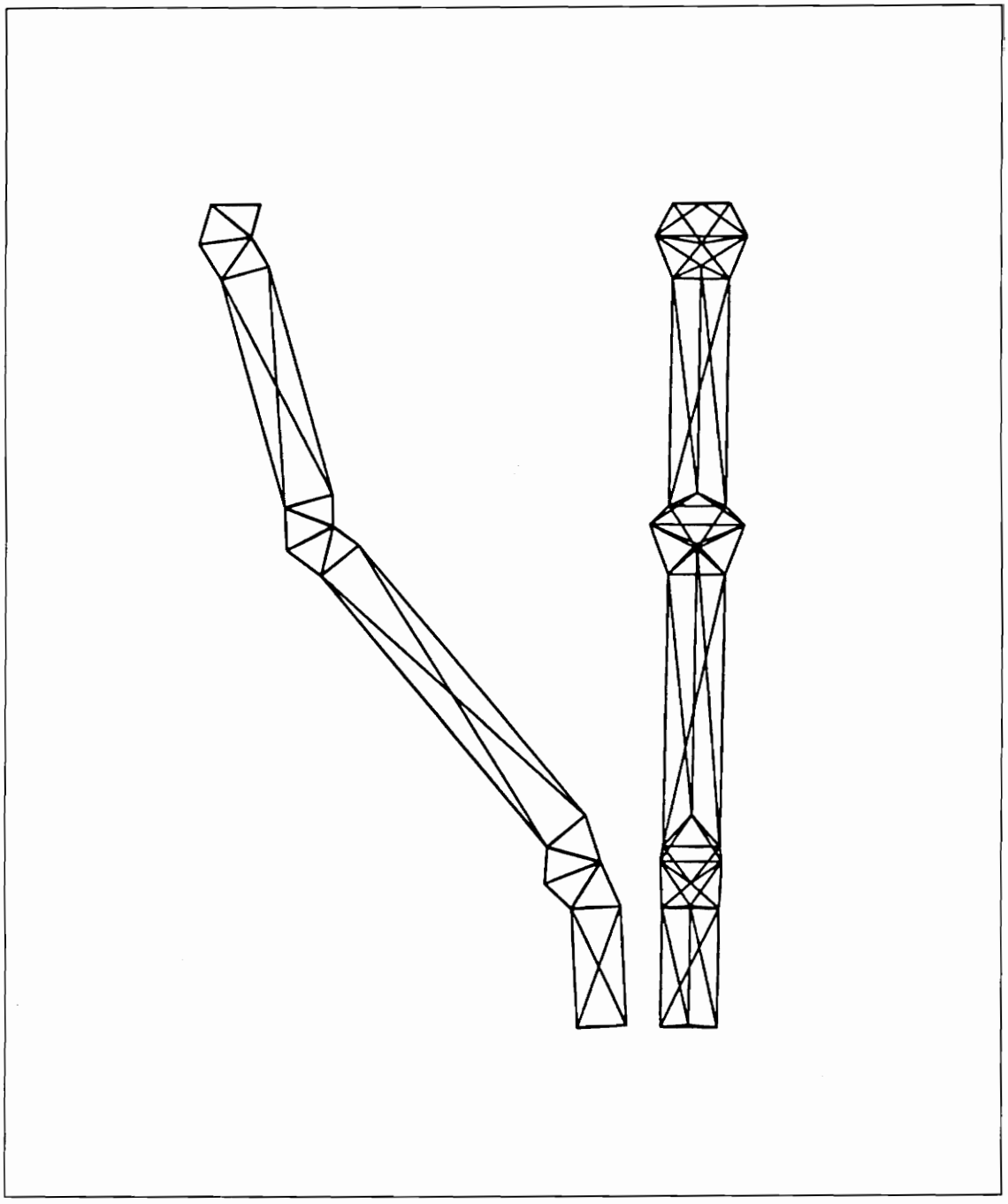


Figure 6.6: Nine Degree-of-Freedom Long-Reach Manipulator

Table 6.1: Estimated KFLOPS (FLOPS/10^3) required for an n Degree-of-Freedom System.

Manipulator Degrees-of-Freedom	SVD KFLOPS	One Iteration KFLOPS	Four Iterations KFLOPS
6	2.8	3.5	14.0
7	4.3	5.3	21.4
8	6.2	7.8	31.0
9	8.7	10.8	43.3
12	19.9	24.8	99.4
15	38.2	47.8	191.0
18	65.4	81.8	327.2
21	103.4	129.2	516.9
30	299.2	374.0	1495.8
60	2380.3	2975.4	11,901.6

To serve as a convenient touchstone, executing a MATLAB benchmark test on an IBM compatible, 66 MHz, 486 localbus machine yields a FLOP rate of 3,121 KFLOPS/second. Thus, this particular machine could generate optimal solutions for an 18 degree-of-freedom manipulator at a frequency of approximately 10 Hz ($\frac{3121}{327.2}$). Although, this is probably sufficient for a slow moving manipulator, the solution frequency for a 60 degree-of-freedom manipulator, 0.3 Hz, is unacceptable for real-time computation. Either a faster computing platform is required or the kinematic degrees-of-freedom must be reduced.

Chapter 7

Order Reduction Methods

The previous chapter illustrated how null space optimization could be employed to produce locally optimum solutions to the redundant inverse kinematic problem. Chapter 6 also demonstrated that there exists an upper bound as to the number degrees-of-freedom that can be satisfactorily solved in real-time for a given computational platform. How then can higher degree-of-freedom systems be adequately addressed? For very redundant systems, those with solution spaces of perhaps ten dimensions or greater, an alternative approach is to employ some type of order reduction method. By applying these methods, the dimension of the solution space can be artificially limited to a size for which solutions can be readily obtained while still preserving some of the improved dexterity provided by the redundant manipulator system. This chapter will introduce three basic types of order reduction methods: degree-of-freedom anchoring, degree-of-freedom coupling, and shape control methods.

7.1 Conventions for Identifying Optimum Solutions

This chapter will seek to identify both full dimension optimal solutions and reduced dimension optimal solutions. The vector $\vec{\tau}$ will again be used to represent the input variables in general. The true or full dimension local optimum solution will be designated as $\vec{\tau}^{\star}$. This is the local optimum solution that is identified if no artificial restraints are placed on the system. The reduced dimension optimum solutions will be denoted as $\vec{\tau}_1^{\dagger}$, $\vec{\tau}_2^{\dagger}$, and so on. These are the optimum solutions that are identified within the reduced order solution spaces. The subscripts will represent that different restrictions are being placed on the system degrees-of-freedom.

7.2 Degree-of-Freedom Anchoring

Direct degree-of-freedom anchoring represents the crudest method of order reduction. In its simplest form, one or more degrees-of-freedom are arbitrarily fixed or anchored. The dimension of the solution space is then reduced by the number of anchored degrees-of-freedom. A slightly more refined version of this approach would be to anchor selected degrees-of-freedom in a manner that is suited to a particular task.

Consider, as an example, the four degree-of-freedom planar manipulator shown in Fig. 7.1. If the goal space is limited to an X, Y position, then this manipulator has two degrees-of-redundancy. If one of the input degrees-of-freedom is anchored the resulting system has only a single degree-of-redundancy. Thus, θ_2 could be arbitrarily set to zero, and the manipulator would still be capable of achieving solutions. In fact, for this particular system, there still exists multiple solutions.

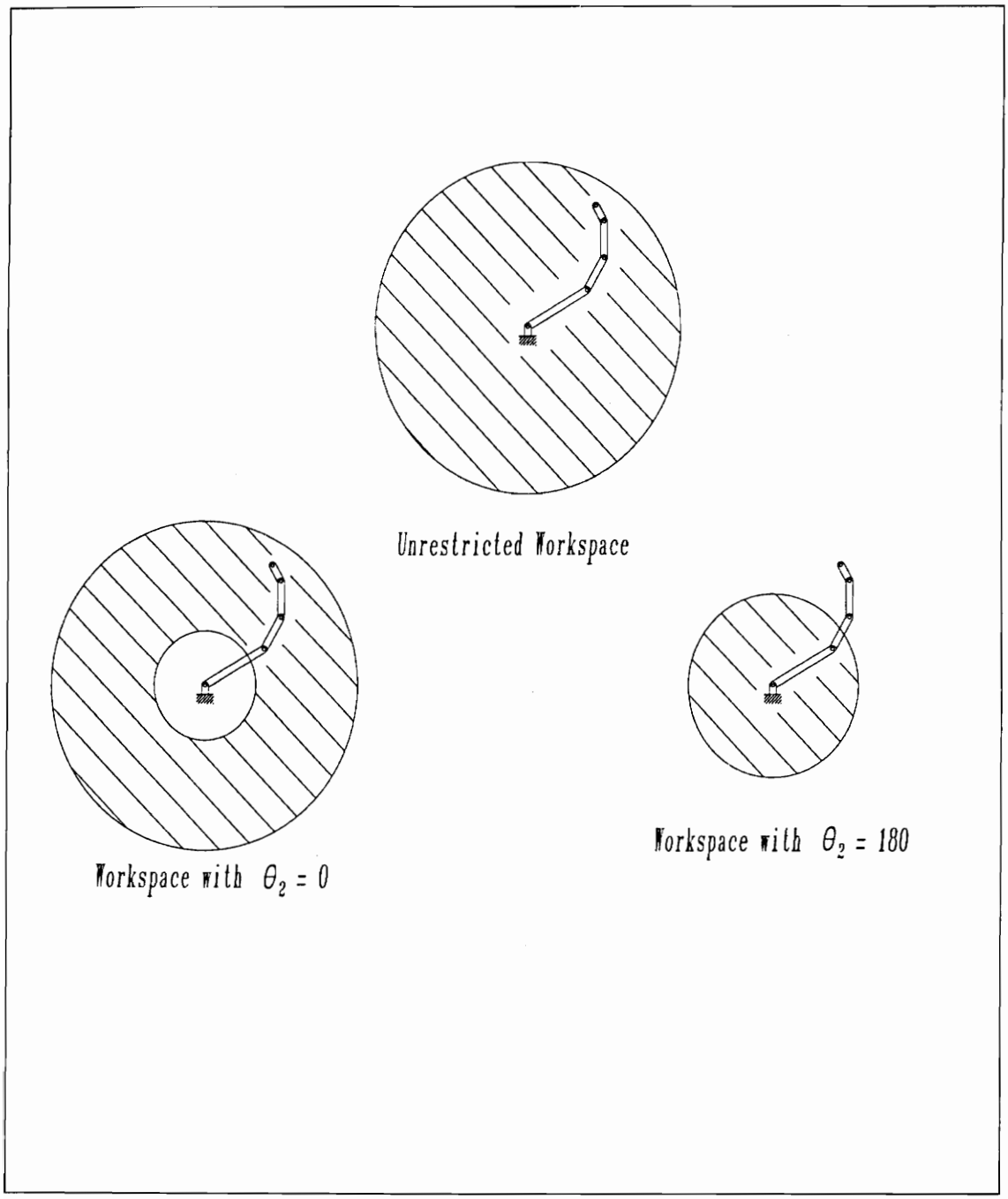


Figure 7.1: Four Degree-of-Freedom Planar Manipulator with Anchored Workspaces Shown

The task of identifying an optimum solution of the reduced solution space is now one dimensional. The obvious problem with arbitrarily fixing some inputs is that the workspace of the manipulator is severely limited. A slightly more refined approach is to set θ_2 equal to 0^o or 180^o depending on the distance of the specified goal position from the origin. Using this slightly modified approach yields one of two limited workspaces shown in Fig. 7.1. The appropriate workspace can be selected for a given task. This method is insufficient for many applications and presents complications when the specified goal positions migrate across the task boundaries.

Because of their severe limitations, neither form of direct degree-of-freedom anchoring is recommended. They are included only to serve as a foundation for the following two methods.

7.2.1 Method of Successive Anchoring

This method is an adaptation of a conventional multi-dimensional optimization algorithm. It is common practice when trying to optimize a function of several variables to perform several successive linear minimizations along the coordinate directions. The function's characteristics are evaluated along a line because, computationally, it is easy to write general optimization code to identify the optimum solution along a line. In essence, this method is anchoring enough system variables to produce a linear solution space. When the optimum to this problem ($\vec{\tau}_1^{\dagger}$) is found, the free variable is fixed and one of the anchored values is freed. The new linear solution space is now evaluated for an optimum ($\vec{\tau}_2^{\dagger}$). Note that by definition, $\vec{\tau}_2^{\dagger}$ must be the same solution or a better solution than $\vec{\tau}_1^{\dagger}$. This procedure then continues until there is no change in the successive line-search optimizations. When this occurs $\vec{\tau}^{\dagger} = \vec{\tau}^{\star}$, that is to say $\vec{\tau}^{\dagger}$ is a local optimum.

In the above discussion, it was assumed that the solution space would be limited

to a line. This was achieved by anchoring a sufficient number of degrees-of-freedom. If fewer degrees-of-freedom are anchored, the optimum of a multi-dimensional solution space must be identified. This is simply the task of finding an optimum using null space optimization methods on a reduced dimension solution space. In fact, the reduced dimension null space can be identified by performing singular value decomposition or Q-R factorization on a matrix which has all the columns of the original Jacobian matrix except for those which correspond to the anchored variables. Using this method, it is therefore possible to analyze, using singular value decomposition or Q-R factorization, the Jacobian matrix a piece at a time. Effectively reducing the dimension of the matrix to be factored. In short, if the original system has too many degrees-of-freedom, it is possible to perform many reduced dimension optimizations instead of one large optimization. The efficiency of this method is dependant on the behavior of the objective function. Fig. 7.2 shows two different objective function contours being evaluated both with improving gradient and linear search techniques. The objective function of Fig. 7.2-A is well suited to reduced order search methods while that of Fig. 7.2-B will require more calculations than the improving gradient method. Again, the simple case represented in the plots of Fig. 7.2 is directly analogous to the systems presented in this dissertation. While these plots show line searches for the reduced dimension search, the methods discussed here would search reduced dimension hyperspaces. Evaluation by the gradient method leads directly to the local optimum, however it requires that all $n-m$ degrees-of-freedom of the solution space be considered. This $n-m$ degree-of-freedom search can be reduced to searching several subspaces of the solution space. Where each subspace is a hyperspace of $n-m-k$ dimensions. k is the number of anchored input variables.

The advantage of the successive anchoring technique is that the manipulator workspace is in no way restricted by the process and the procedure converges to

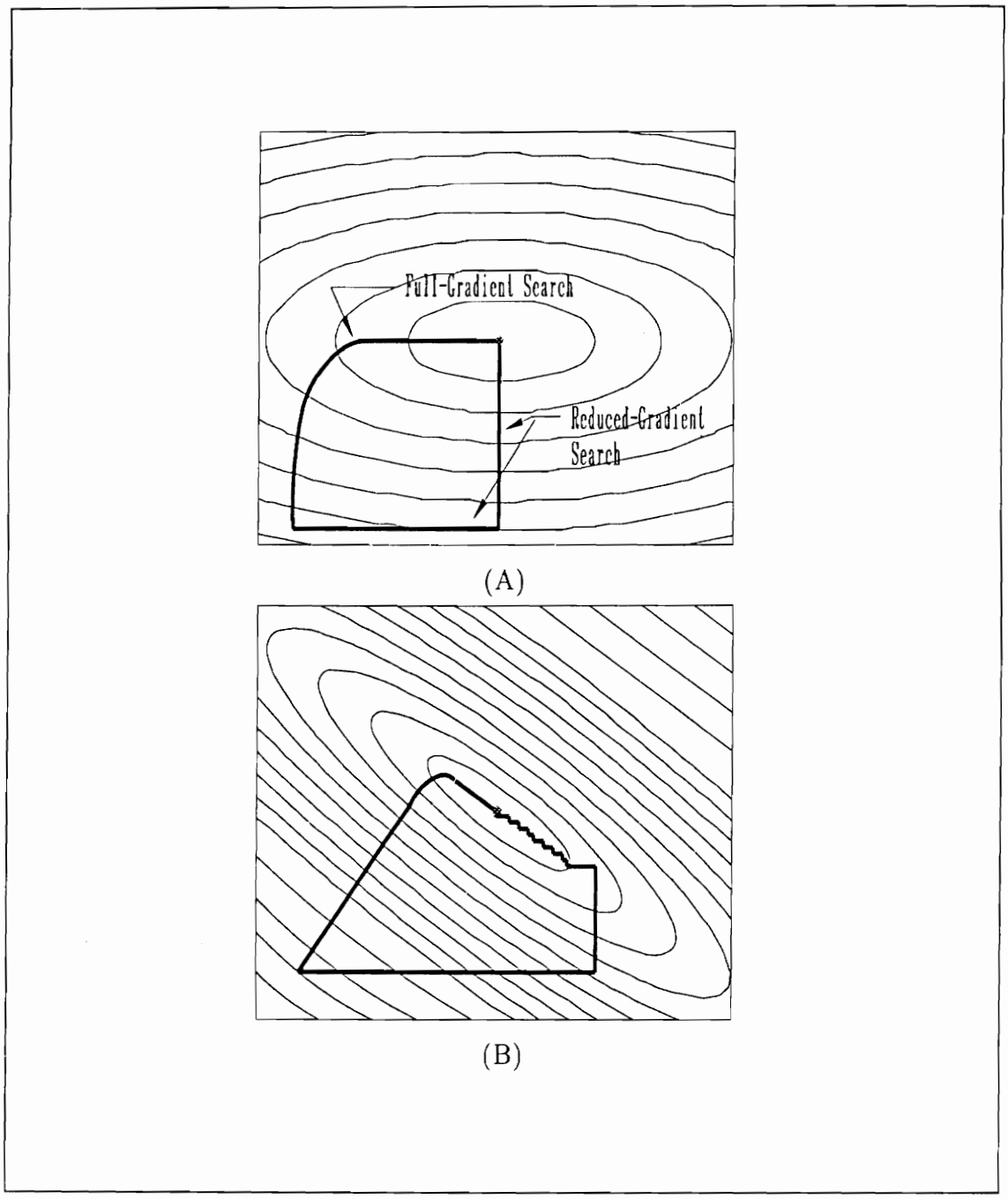


Figure 7.2: Objective Function Contours with Search Methods Shown

the true local optimum solution $\vec{r}^{\star}$.

It was demonstrated earlier that convergence from a distant initial point could possible take more calculations than the full gradient method. However, successive anchoring is efficient for identifying a nearby optimum solution. For this reason it is feasible to use this method when moving a small distance from a previously obtained optimum.

Another argument can be made in favor of this method when it is deemed sufficient to be in close proximity to the full dimension optimum solution. Consider the case where the manipulator is to move from goal A to goal B in several small discrete steps. Starting from an optimal configuration at A, the manipulator could move to the first intermediate position while performing a reduced dimension optimization. Presumably, if the objective function is well behaved, the manipulator will be very near to the true optimum configuration. For the motion to the next intermediate position the anchored variables could be freed and a new set of variables anchored. Again, at the end of this motion, the manipulator should be very near the true optimum configuration. With each successive move, the anchored variables are cycled and periodically introduced into the optimization. If desired, for the final motion to the goal B, the true optimum could be identified by continuing to cycle the anchored variables until convergence.

7.3 Degree-of-Freedom Coupling

From a conceptual design viewpoint, the primary reason for developing redundant manipulators is to increase dexterity. The extra degrees-of-freedom afforded by these devices can be used for obstacle avoidance or structural optimization. Although these tasks are required for some applications, a great many problems can be resolved without kinematic redundancies or with only limited degrees-of-

redundancy. For parallel, truss-type manipulators, there is another, more practical, reason for developing redundant manipulators. A single VGT module has a relatively limited workspace. The increases in strength and stiffness afforded by these devices is obtained at the expense of overall workspace volume. To broaden this restricted workspace, several VGT modules can be joined together to form a long-chain manipulator. For many applications, the increased workspace is the primary motivation for developing long-chain, redundant manipulators. In these cases, the accompanying improved dexterity can be considered an advantageous by-product. Unfortunately, this by-product carries a large computational price.

In Section 4.1 some general rules were formulated for how to best construct a long-chain manipulator. It was stated in this section that many times it is favorable to cluster several degrees-of-freedom in one region. Often, this clustering is done to increase the range of motion for a particular segment of the manipulator. For example, consider the 15 degree-of-freedom manipulator shown in Fig. 7.3 which contains batten-actuated, double-octahedral active modules. The particular geometry of these modules is such that they are only capable of producing 45° angular motions. The first two VGT modules have been placed close together to ensure that the manipulator is capable of bending 90° near the base. For this purpose, these two VGT modules could effectively function as one unit. Although there are physically six actuators present in these first two active modules, the system inputs could be artificially coupled so that $l_{i,1} = l_{i,2};\ i = 1 \ldots 3$. Or, in more general terms, $\tau_{i,1} = \tau_{i,2};\ i = 1 \ldots 3$. That is to say, these two modules will be constrained to always have identical positional inputs. This degree-of-freedom coupling reduces the total number of system degrees-of-freedom to 12. If this proves too restrictive, and slightly more freedom is required, the system inputs could be coupled so that $\tau_{i,1} = k\ \tau_{i,2};\ i = 1 \ldots 3$. This results in the system having 13 degrees-of-freedom, the extra degree-of-freedom being afforded by the

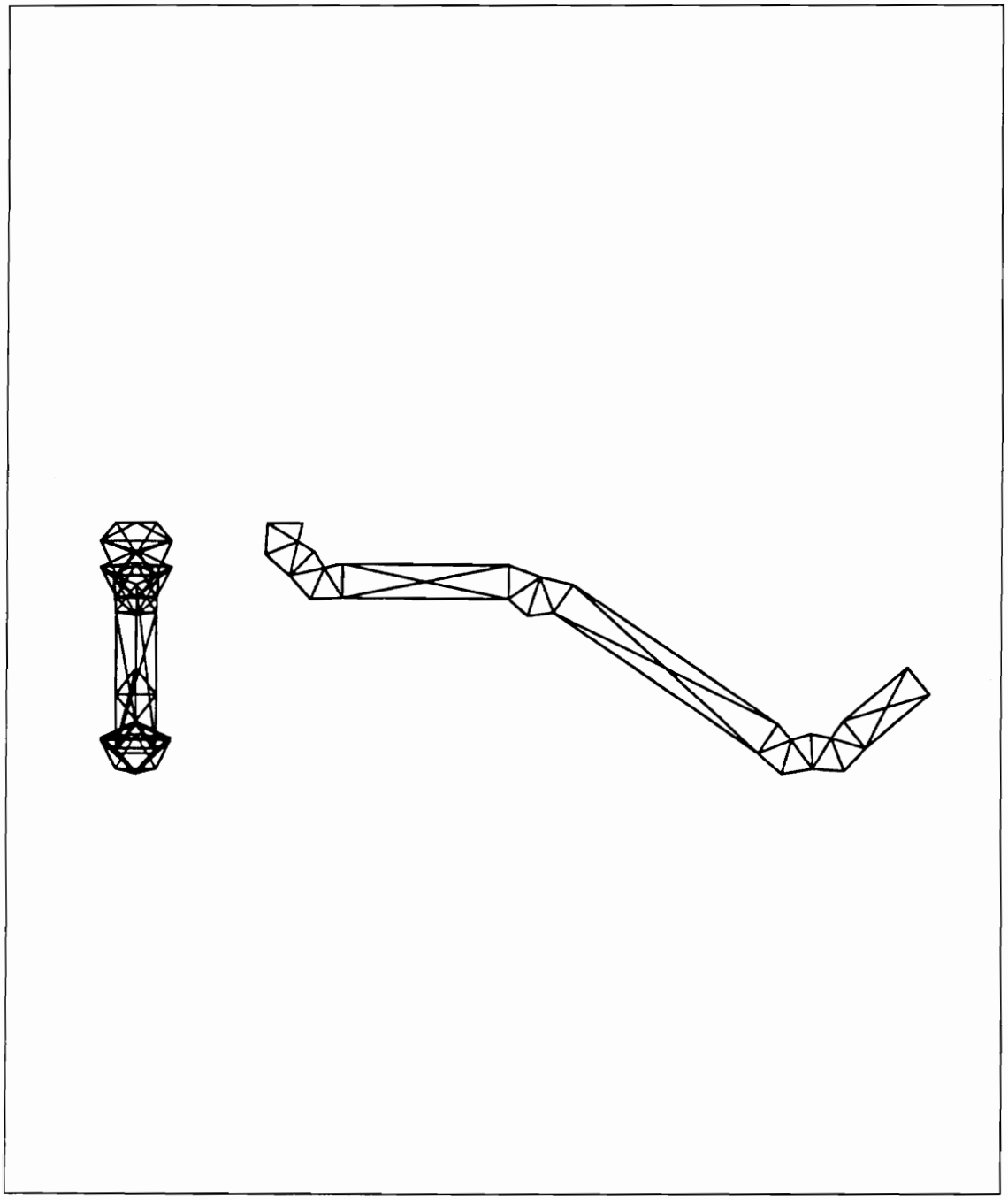


Figure 7.3: A 15 Degree-of-Freedom Manipulator with Coupled Constraints

proportionality variable k.

Although this method does reduce the the system dexterity, if properly implemented, it will not appreciably limit the system workspace. Thus, the reach of the manipulator is preserved, while reducing the computational burden imposed on the host controller.

7.4 Shape Control

This section will outline an algorithm for solving the inverse kinematics of a particular class of long-chain, high-degree-of-freedom VGT-based manipulator. This discussion will be limited to manipulators which satisfy the following criteria:

1. The active modules must be octahedral-based VGT's. This is required to ensure that each segment is capable of motion about all three axes.

2. The manipulator must possess either no static modules or alternating active and static modules where the static modules are relatively short. As a general guideline, the static modules should be no longer than the active modules.

3. The active modules must all have the same actuation scheme and be of similar proportions.

4. The static modules, if present, must all have similar proportions.

These restrictions guarantee that the resulting manipulator is continuously articulated along its length. This repetitive structure and even distribution of the degrees-of-freedom enables the manipulator to mimic the form of long continuous spatial curves. This is a departure from previous methods which imposed no such restrictions on the construction of the manipulator. In many ways this can be viewed as the highest evolution of a redundant manipulator, there are so many

degrees of freedom that the manipulator is appropriately modelled as a continuous curve.

The underlying philosophy of the shape control approach is to find some minimum order curve that satisfies the desired position requirements of the truss. Once a suitable curve is found, the curve is divided into a series of short curve segments. The end conditions of each of these curve segments are then used to determine the goal specification for each corresponding manipulator segment. A segment may be composed of an active module alone or both an active and a static module. By performing this operation, the task of positioning a manipulator with ten segments has been decomposed into the task of positioning ten single-module manipulators. In fact, since the kinematic solution of one segment has no effect on the adjacent segments, these ten subproblems may be solved in parallel.

In general, a cubic curve will be the minimum order curve necessary to realize an arbitrary goal specification where the position is completely specified and the orientation is given in terms of a directional vector. This is a five degree-of-freedom goal specification, as might be used for the batten-actuated, double-octahedral geometry truss. If complete control of position and orientation (six degrees-of-freedom) is required, a swept surface must be utilized.

In short, an overall solution is sought which smoothly progresses from an initial, or base, configuration to a desired terminal, or end-effector, configuration. This is not the only possible solution. The shape control method results in a solution which distributes the total required motion over the individual modules.

7.4.1 Inverse Kinematic Solution for a Continuously Articulated, Modular Manipulator Using Shape Control

As depicted in Fig. 7.4, the shape control solution is achieved in three distinct

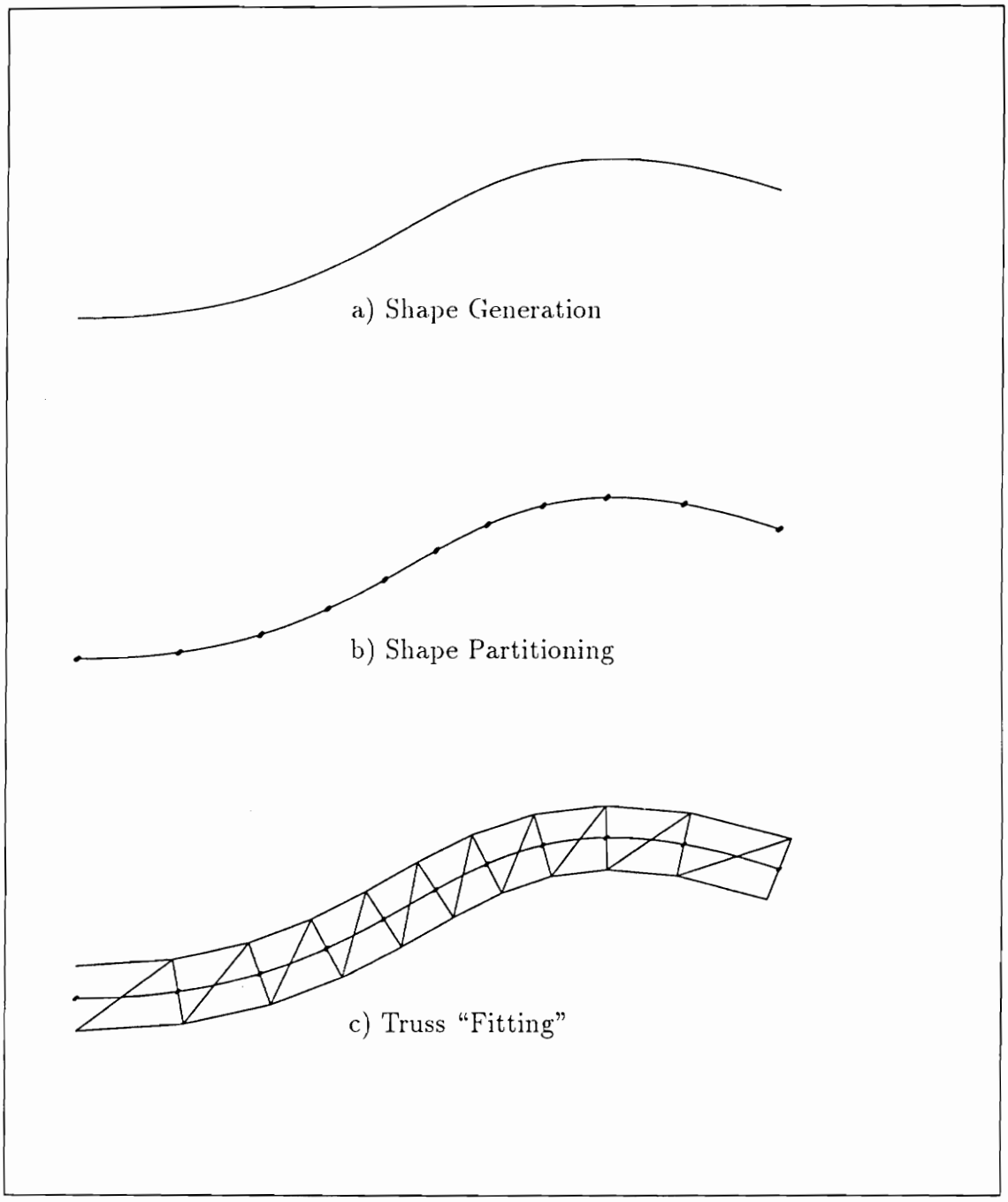


Figure 7.4: The Three Stages to Achieve a Shape Control Solution

stages. First, a shape must be generated which satisfies the desired end conditions. For a planar manipulator, as depicted in Fig. 7.4, this shape can be a curve; however, for most spatial manipulators this shape must be a swept surface. The resulting shape must then be partitioned in a way that provides a reasonable distribution of the intermediate truss planes. Finally, the true actuator lengths must be calculated and the active truss sections fitted to the shape solution. The following sections will summarize each phase of the manipulator shape solution in greater detail. A more comprehensive treatment of this material is contained in Salerno [48].

Parametric Description of Swept Surface

The general cartesian form of a spatial cubic curve can be represented as follows:

$$\begin{aligned} X(u) &= a_{0x} + a_{1x}u + a_{2x}u^2 + a_{3x}u^3, \\ Y(u) &= a_{0y} + a_{1y}u + a_{2y}u^2 + a_{3y}u^3, \\ Z(u) &= a_{0z} + a_{1z}u + a_{2z}u^2 + a_{3z}u^3. \end{aligned} \tag{7.1}$$

Or, in vector form,

$$\vec{C}(u) = \vec{A}_0 + \vec{A}_1 u + \vec{A}_2 u^2 + \vec{A}_3 u^3. \tag{7.2}$$

where u is a parameterization variable which progresses from 0 to 1; 0 corresponding to the initial condition or base of the manipulator, and 1 corresponding to the terminal condition or end of the manipulator. The position of any point along the curve can be expressed as a function of the curve end conditions as follows:

$$\vec{C}(u) = \begin{bmatrix} \vec{C}(0) & \vec{C}(1) & \vec{C}'(0) & \vec{C}'(1) \end{bmatrix} [\mathbf{F}] \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix}; \tag{7.3}$$

where,

$$\vec{C}(u) = \begin{bmatrix} X(u) \\ Y(u) \\ Z(u) \end{bmatrix}, \tag{7.4}$$

$$\vec{C}'(0) = \left.\frac{d\vec{C}(u)}{du}\right|_0 = t_0\hat{q}_0, \tag{7.5}$$

$$\vec{C}'(1) = \left.\frac{d\vec{C}(u)}{du}\right|_1 = t_1\hat{q}_1, \tag{7.6}$$

and

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \tag{7.7}$$

Specification of this cubic curve requires twelve independent parameters, three each for the initial and final position vectors, $\vec{C}(0)$ and $\vec{C}(1)$, and three each for the initial and final tangent vectors, $\vec{C}'(0)$ and $\vec{C}'(1)$. These correspond to the twelve parameters afforded by the general parametric description, $\vec{A}_0, \ldots \vec{A}_3$. Note that the initial and final tangent vectors are represented by a tangent vector magnitude, t, and a tangent unit vector $\hat{q}$. Similarly, the tangent vector for any value of u can be evaluated by differentiating Equation 7.3 with respect to u.

$$\vec{C}'(u) = \begin{bmatrix} \vec{C}(0) & \vec{C}(1) & \vec{C}'(0) & \vec{C}'(1) \end{bmatrix} [\mathbf{F}] \begin{bmatrix} 0 \\ 1 \\ 2u \\ 3u^2 \end{bmatrix} \tag{7.8}$$

Now, for any value of u, both a spatial position vector, $\vec{C}$, and a spatial tangent vector, $\vec{C}'$, can be found. See Fig. 7.5. The position vector will be utilized to specify where the local base coordinate frames will be located. The tangent vector will

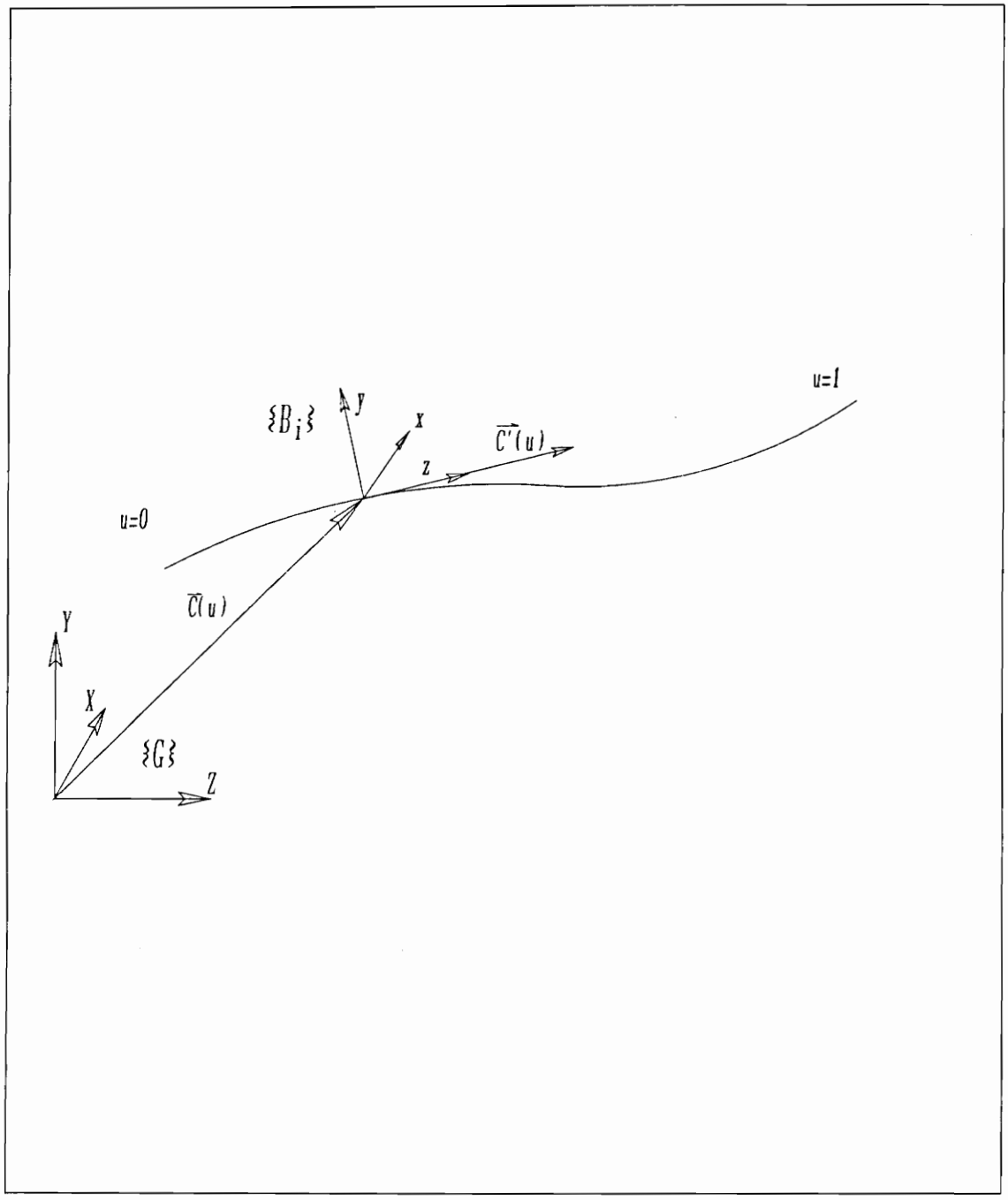


Figure 7.5: Description of i^{th} Base Frame in Global Coordinates

be used to orient one axis of the local base coordinate frame, in this case the z axis. Note that there are still an infinite number of orientations that can satisfy these constraints. They are generated by rotating the local base frame about its z axis. Thus, there is still a need to describe the desired rotation about the local z axis. It is possible to extract the curve normal and bi-normal vectors and use these to orient the local base frame. These two curve characteristics, however, are discontinuous or, in some cases, undefined along the length of the curve. Because of this shortcoming, a cubic curve alone will not provide enough information to properly orient the local base frames. Therefore, more than twelve parameters are needed to define the desired shape.

The additional degrees of freedom required can be obtained by employing a surface rather than a curve. In this case a swept surface is used. This surface is formed simply by moving a given shape, the generatrix, along a guiding curve, the directrix, such that the plane of the generatrix is always perpendicular to the directrix [26]. Such a motion results in the surface illustrated in Fig. 7.6-A; where the directrix function is the prescribed cubic curve and the generatrix function is an equilateral triangle with sides of length L. This method of generation produces a surface that smoothly progresses to the desired end position and tangent direction. It does not however, have any capacity to result in an arbitrary end rotation about the curve tangent. If however, simultaneous with the motion along the curve, the generatrix is also rotated about the curve, the result is a helically swept surface as shown in Fig. 7.6-B. By specifying the proper rate of twist, the rotation of each individual base frame can be assigned such that the resulting surface smoothly twists to the desired end orientation. Since it is convenient to describe the surface in terms of the specified end conditions, two more parameters are added to describe this helical twist. These are the initial twist about the curve tangent, λ_i, and the final twist about the curve tangent, λ_f. Additional information about general

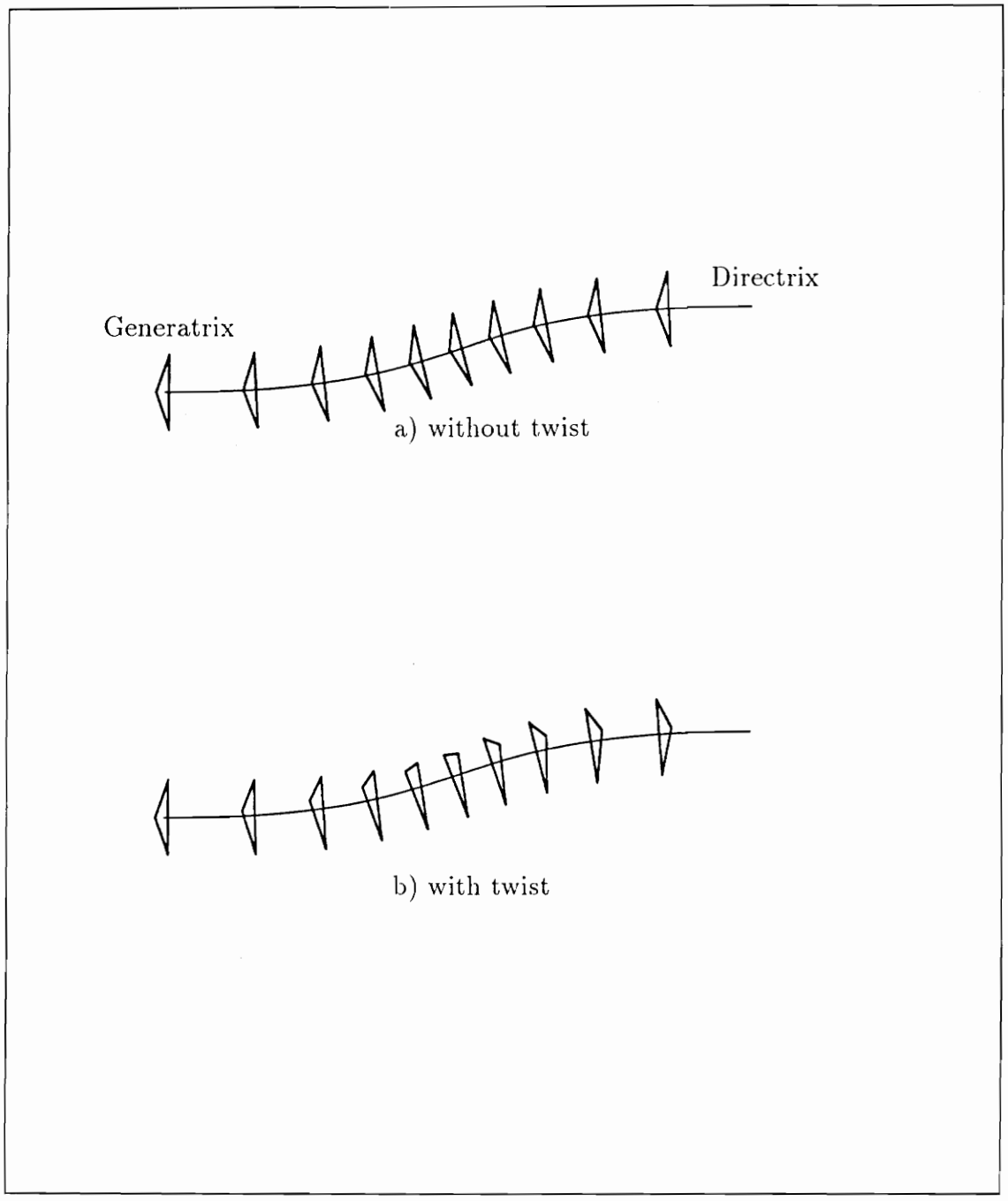


Figure 7.6: Generation of a Swept Surface

Table 7.1: Enumeration of Degrees-of-freedom

Description	Number of Parameters
Manipulator Base Position	3
Manipulator Base Orientation	3
Initial Tangent Vector Magnitude	1
End-Effector Position	3
End-Effector Orientation	3
Terminal Tangent Vector Magnitude	1

swept surfaces is contained in [32, pages 455–460].

The defining equations for the directrix function have already been discussed. It is now necessary to briefly discuss how to conveniently form the generatrix function. This is most easily accomplished by forming a circle that lies in the x-y plane and has a radius of $r = \frac{2}{3}(L \sin 60^o)$. Thus, any point on the circle can be defined as a function of ν as follows:

$$\vec{w}(\nu) = \begin{bmatrix} r\cos\nu \\ r\sin\nu \\ 0 \end{bmatrix}. \tag{7.9}$$

If this expression is evaluated at $\nu = 0, \frac{2\pi}{3}$, and $\frac{4\pi}{3}$, the three resulting points can be used to define the triangular generatrix. As this generatrix is swept along the cubic curve, a bi-parametric surface is generated which is a function of u and ν.

To completely define this swept surface requires 14 parameters. One possible enumeration of these parameters is given in Table 7.1. For the long-reach manipulator system described, when reaching to a specific goal, all of the 14 parameters are fixed except for the initial and terminal tangent vector magnitude, t_0 and t_1. Figure 7.7 shows a plot of five planar cubic curves that have the same position

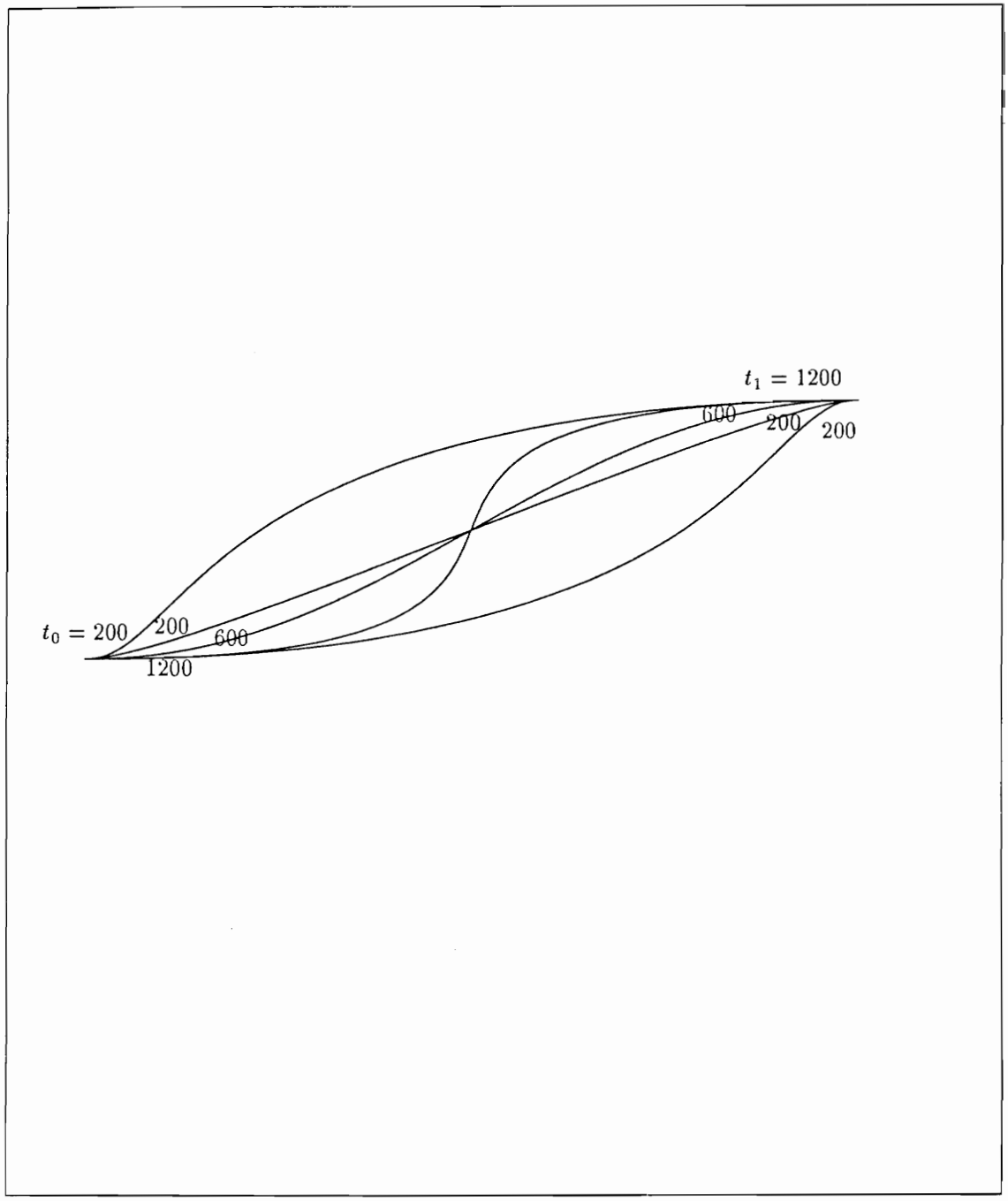


Figure 7.7: Effect of Tangent Vector Magnitude on Curve Shape

and orientation specifications at the ends, however, each curve has a different set of tangent magnitudes. As illustrated in Fig. 7.7, increasing the tangent vector magnitude makes the curve maintain the approximate end slope longer. This figure also implies that there exists two infinities of solutions for a cubic curve that is specified by only the position and orientation of its ends. This ability of the parametric cubic curve to produce multiple solutions is very attractive for the shape control of long-chain VGT's. If the additional flexibility is not required, the two tangent magnitudes can be set to an arbitrary non-zero value. This property also provides alternative solutions for avoiding workspace obstacles or for optimizing some structural index.

Basic Curve Partitioning and Completion of the Shape Solution

This section will concentrate on the partitioning methods and the subsequent procedures necessary to accomplish a complete shape solution. The surface shape has already been defined by the directrix, generatrix, and twist angles. Now the task at hand is to use this information to somehow "fit" a truss structure inside the surface.

Before the variable link lengths of the individual modules can be identified, it is first necessary to partition the shape solution into n segments. Where n is the total number of manipulator segments to be used. For demonstration purposes, let these partition points be placed along the cubic curve at $u = \frac{1}{n}, \frac{2}{n}, \ldots \frac{n}{n}$. This method of partitioning will be referred to as a regular u spacing. For a given cubic curve, this partitioning scheme might result in a curve such as the one illustrated in Fig 7.4-B. Notice that the partition points are identified as p_0, p_1,... p_n; where the subscript denotes which segment terminates at the given partition point. For example, the fifth segment terminates at point p_5, which is located, in this instance, by

$\vec{C}(5/n)$. For the present discussion it will be assumed that a regular u partitioning is sufficient to produce an acceptable solution. More refined partitioning techniques will be addressed in subsequent sections.

For any value of u, the spatial cubic curve description provides information about the curve position and tangent vectors. If the rate of twist of the surface is assumed to be constant with respect to u, then the twist for any value of u is,

$$\lambda(u) = (1-u)\lambda_i + u\lambda_f \tag{7.10}$$

It is possible that such a twist law might result in a surface which has a variable rate of twist with respect to the curve arc length. A more acceptable solution is obtained by assuming the rate of twist is is constant with respect to the segment index number i. Thus, an easy approximation to constant rate of twist with respect to arc length is to have,

$$\lambda(i) = \left(1 - \frac{i}{n}\right)\lambda_i + \frac{i}{n}\lambda_f. \tag{7.11}$$

Now the rotation of any base coordinate frame, $\mathcal{B}_i$, which occurs at a partition point, p_i, can be found using the following procedure.

Begin with frame $\mathcal{B}_i$ coincident with the global reference frame, $\mathcal{G}$. It was mentioned previously that the z axis of frame $\mathcal{B}_i$ should be parallel to the curve tangent. This requires, in general, two rotations, shown in Fig. 7.8. Utilizing X-Y-Z Euler angle descriptions (sometimes referred to as Bryant angle descriptions) [12, pages 42–45][37, pages 347–352], the desired rotation about the x axis of frame $\mathcal{B}_i$ is,

$$\gamma(u) = -\text{ATAN2}({}^{\mathcal{G}}t_y, {}^{\mathcal{G}}t_z); \tag{7.12}$$

where t_x, t_y, and t_z are the X, Y, and Z components of the tangent vector $\vec{T}(u)$.

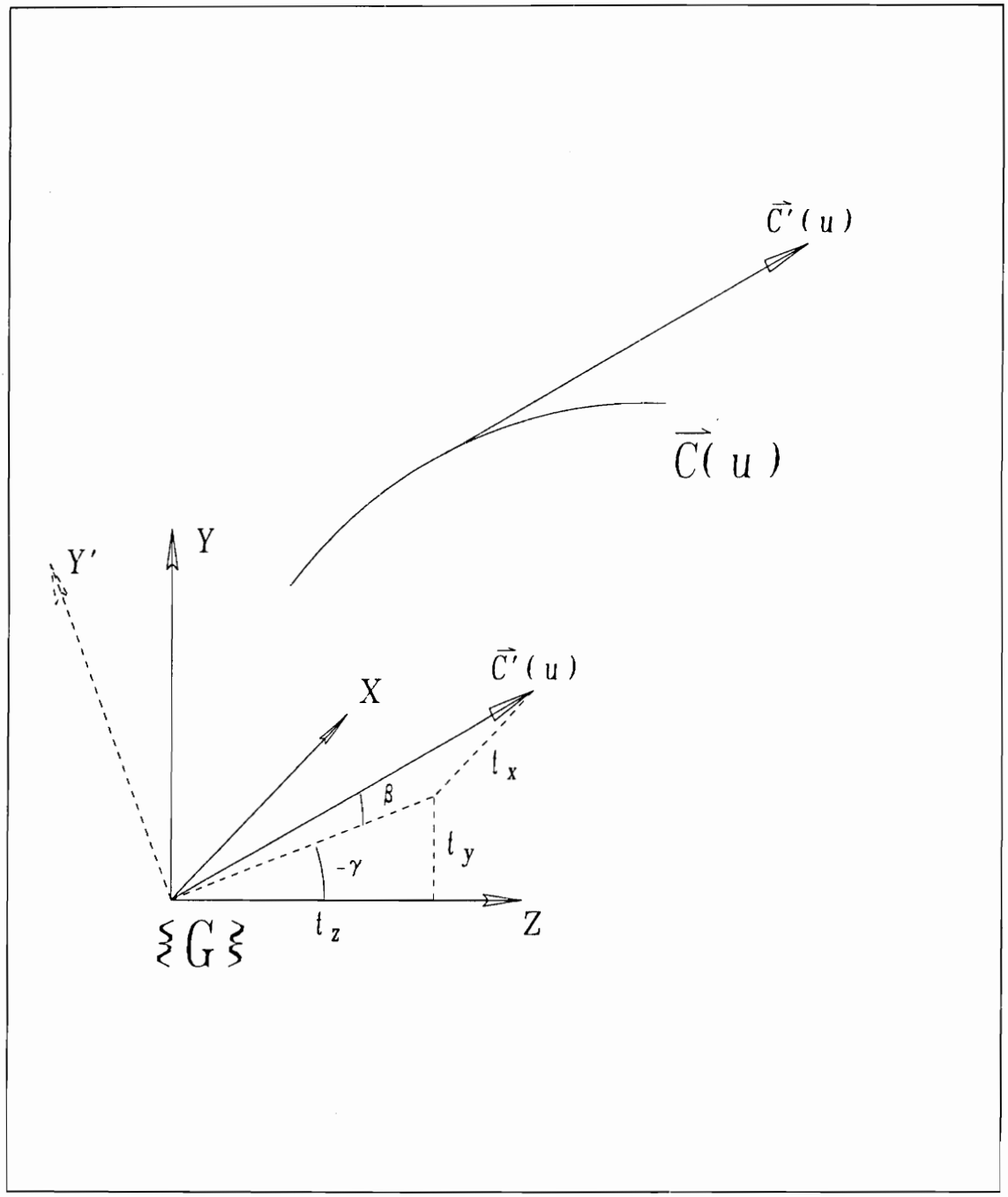


Figure 7.8: X-Y-Z Euler (Bryant) Angle Description of Tangent Vector

The second rotation, which occurs about the now rotated y' axis of frame $\mathcal{B}_i$, is found as,

$$\beta(u) = \text{ATAN2}({}^{\mathcal{G}}t_x, \sqrt{{}^{\mathcal{G}}t_y{}^2 + {}^{\mathcal{G}}t_z{}^2}), \tag{7.13}$$

if t_z is positive. Or, if t_z is negative,

$$\beta(u) = \text{ATAN2}({}^{\mathcal{G}}t_x, \sqrt{{}^{\mathcal{G}}t_y{}^2 + {}^{\mathcal{G}}t_z{}^2}) + \pi. \tag{7.14}$$

Note that the z axis of $\mathcal{B}_i$ is now coincident with the curve tangent. Finally, the rotation of frame $\mathcal{B}_i$ about its own z'' axis ($z'' = \frac{\vec{T}(u)}{||\vec{T}(u)||}$) is given from the previous definition of the surface twist. That is,

$$\alpha(u) = \lambda(u). \tag{7.15}$$

These three rotations can now be combined into one rotation matrix as follows:

$$R_{xyz}(u) = ROT(\hat{X}_{\mathcal{B}_i}, \gamma)ROT(\hat{Y}_{\mathcal{B}_i}, \beta)ROT(\hat{Z}_{\mathcal{B}_i}, \alpha) \tag{7.16}$$

$$= \begin{bmatrix} c\alpha c\beta & -s\alpha c\beta & s\beta \\ c\alpha s\beta s\gamma + s\alpha c\gamma & -s\alpha s\beta s\gamma + c\alpha c\gamma & -c\beta s\gamma \\ -c\alpha s\beta c\gamma + s\alpha s\gamma & s\alpha s\beta c\gamma + c\alpha s\gamma & c\beta c\gamma \end{bmatrix} \tag{7.17}$$

This rotation matrix can be used in conjunction with the translation information to yield a homogeneous transformation that maps vectors from the $\mathcal{B}_i$ coordinate frame to the global reference frame $\mathcal{G}$.

$${}^{\mathcal{G}}_{\mathcal{B}_i}T = \begin{bmatrix} \begin{bmatrix} & R_{X-Y-Z}(u) & \end{bmatrix} & \begin{bmatrix} \vec{C}(u) \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & 1 \end{bmatrix} \tag{7.18}$$

Since the three corner points of the static triangles are fixed with respect to the local base frames, the preceding transformation can also be used to translate and rotate the generatrix to its new position and orientation. This results in a bi-parametric description of the corner points as follows:

$$ {}^{\mathcal{G}}\vec{w} = \left[{}^{\mathcal{G}}_{\mathcal{B}_i}T(u) \right] \begin{bmatrix} r\cos\nu \\ r\sin\nu \\ 0 \\ 1 \end{bmatrix} \tag{7.19} $$

This expression can now be evaluated at any value of u for values of $\nu = 0, 2\pi/3$, and $4\pi/3$, to yield three vectors that define the three corner points. The lengths of the extensible members are then found as the distance between appropriate corners of adjacent triangles. Figure 7.9 shows a continuously articulated manipulator constructed using this algorithm with a regular u partition.

More Advanced Partitioning Techniques

It is evident from observation of Fig. 7.9 that a regular u spacing does not result in an evenly partitioned curve. This could produce unacceptable solutions in many cases. More advanced curve partitioning methods have been developed [48]. A discussion of these techniques will not be undertaken here. However, it should be mentioned that better solutions can be obtained by redistributing the partition points. Often times, this redistribution of points can be used for optimization of a given objective function. One possible partitioning algorithm is to redistribute the partition points such that they fall at regular arc length intervals along the cubic curve. A solution identified with an equal arc length algorithm is shown in Fig. 7.10.

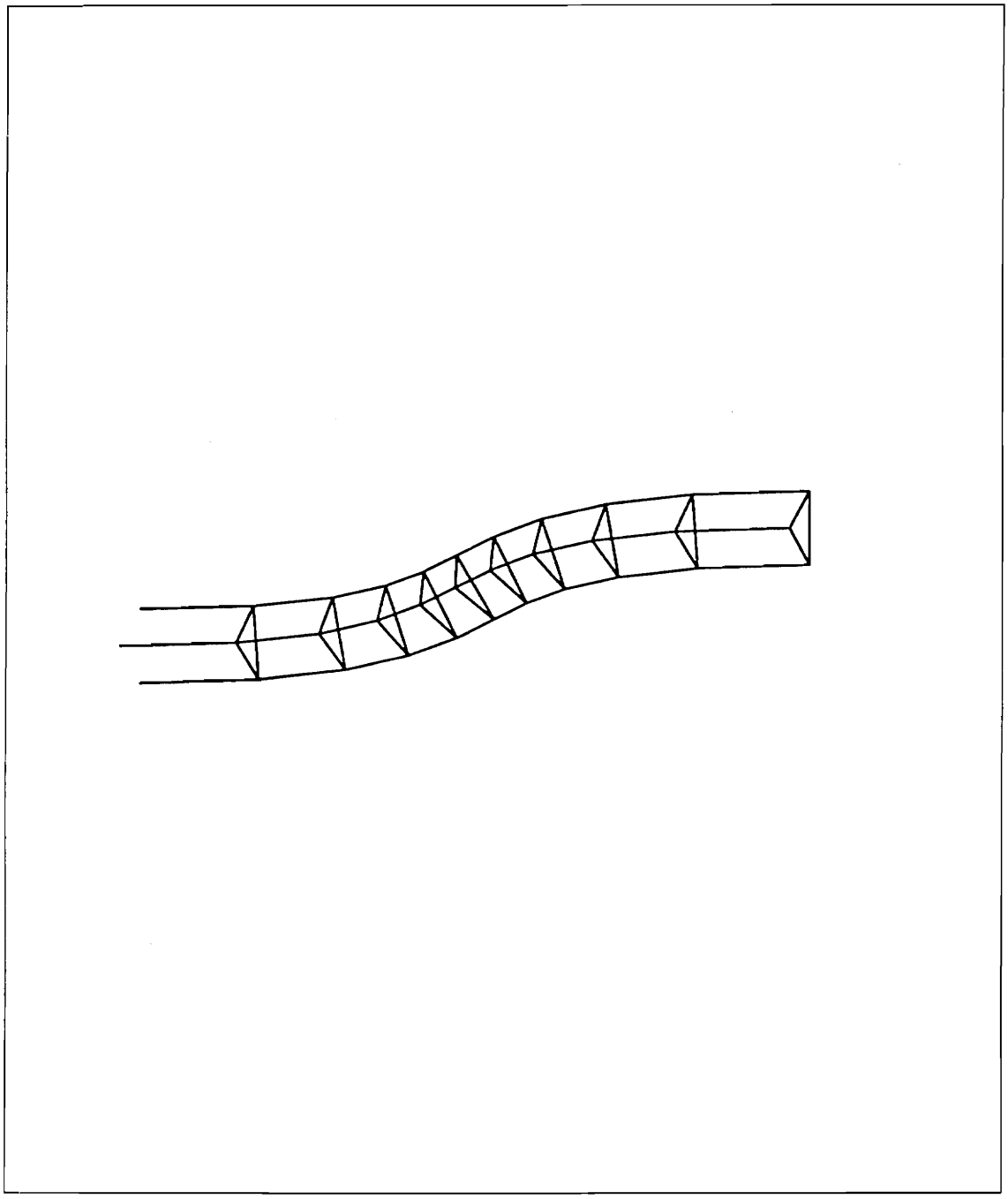


Figure 7.9: Spatial Truss Partitioned with Regular u spacing

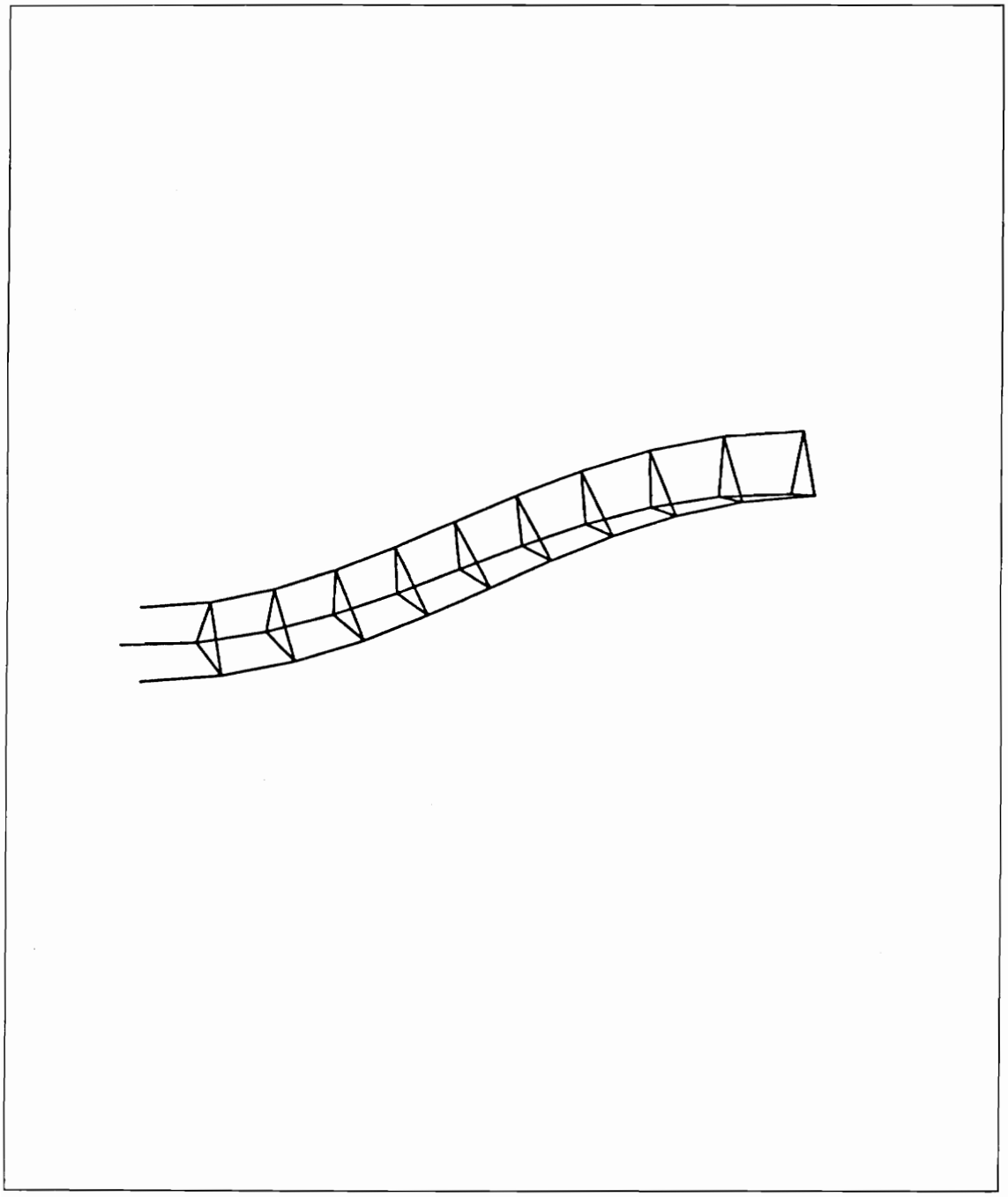


Figure 7.10: Spatial Truss Partitioned with Equal Arc Length Algorithm

Chapter 8

Modular, Long-Reach Manipulator Examples

This chapter will present examples of how the methods previously presented can be used to solve typical redundant inverse positioning problems. Two groups of examples will be presented. The first group will illustrate both null space optimization and degree-of-freedom coupling techniques. These two method are presented together because the use of one of these methods does not preclude the use of the other. Often the most practical solution to this type of problem is obtained by using both methods simultaneously. The second group of examples will present several shape control solutions. Because of the restrictions that accompany the shape control method, this technique is always applied independent of the other solution methods.

8.1 Inverse Kinematic Solution for a Fifteen Degree-of-Freedom Manipulator

This section will illustrate how the null space optimization methods can be applied to the redundant inverse kinematic problem posed by a typical configuration of the

long-reach, modular manipulator. To show the algorithms' ability to quickly optimize within multidimensional solution spaces, an examples will be given for the fifteen degree-of-freedom manipulator shown in Fig 8.1. This manipulator is constructed from alternating batten-actuated, double-octahedral VGT modules and static truss sections. Because the batten-actuated, double octahedral geometry has limited ability to produce a roll about its longitudinal axis, the goal specification for these examples will be limited to specifying a target position (x_d, y_d, z_d), and a unit vector, $\hat{q}_d$, that is normal to the last end plane of the manipulator. This unit vector possesses only two degrees-of-freedom, and as such can be completely specified by any two quantities of $\hat{q}_d$. The desired goal position will therefore be represented by a 5×1 column vector of the form

$$\vec{\chi}_d = \begin{bmatrix} x_d \\ y_d \\ z_d \\ q_{xd} \\ q_{yd} \end{bmatrix} \tag{8.1}$$

To begin the inverse analysis, an efficient forward kinematic solution must first be formulated. Chapter 4 presented methods for identifying the position and orientation of the end-effector given the appropriate input parameters. The resulting positional relationship was represented with a homogeneous transform. Applying this to the fifteen degree-of-freedom manipulator yields:

$$\overset{\mathcal{B}}{\mathcal{T}}T = T_{v1} \cdot T_{s1} \cdot T_{v2} \cdot T_{s2} \cdot T_{v3} \cdot T_{s3} \cdot T_{v4} \cdot T_{s4} \cdot T_{v5} \cdot T_{s5} \tag{8.2}$$

where T_{vi} represents the variable transformation from the base plane to the top plane of the i^{th} VGT module. These transformations are solely a function of the specified input parameters (either conventional or canonical). T_{si} represents the static transformation from the base plane to the top plane of the i^{th} static truss

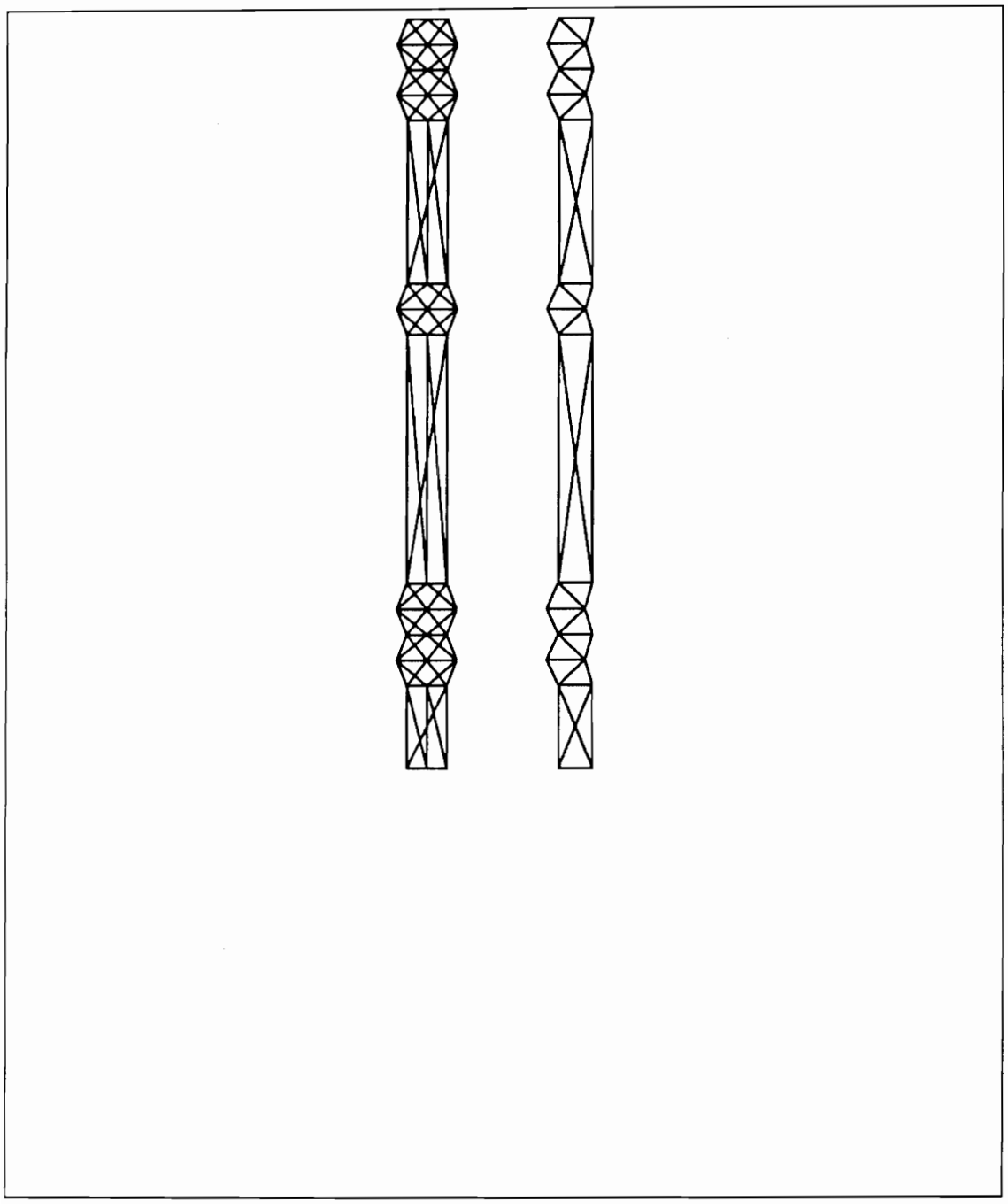


Figure 8.1: Fifteen Degree-of-Freedom Manipulator

section.

To reduce the number of matrix multiplications, the static transformation could be combined with the variable transformations while off-line as follows:

$$T_i(\tau_{1,i}, \tau_{2,i}, \tau_{3,i}) = T_{vi} \cdot T_{si} \tag{8.3}$$

This results in the following representation of the forward kinematics.

$$\overset{B}{T}T = T_1 \cdot T_2 \cdot T_3 \cdot T_4 \cdot T_5 \tag{8.4}$$

Thus, given the input values, either conventional or canonical, the end-effector position in the goal space can be identified by extracting the appropriate values from $\overset{B}{T}T$. If the elements of these transformations are identified as

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8.5}$$

then the goal space position can be found as:

$$\vec{\chi}(\vec{\tau}) = \begin{bmatrix} t_{14} \\ t_{24} \\ t_{34} \\ t_{13} \\ t_{23} \end{bmatrix} \tag{8.6}$$

Although it is not necessary for a manipulator with only fifteen degrees-of-freedom, observation of Fig. 8.1 leads one to conclusion that the first and second, and forth and fifth VGT modules are candidates for the degree-of-freedom coupling method of order reduction. If these modules are coupled so that VGT modules one and two behave as a three degree-of-freedom unit, and modules four and five

behave as another independent three-degree-of-freedom unit, then a nine degree-of-freedom manipulator results. The individual transformations of the forward kinematics can then be modified so that

$$\overset{B}{T}T = T_{12} \cdot T_3 \cdot T_{45} \tag{8.7}$$

where, T_{12} and T_{45} represent the three-degree-of-freedom coupled transformations. Again the modified nine degree-of-freedom forward kinematic algorithm would yield a goal space position, $\vec{\chi}(\vec{r})$, identical in form to Equation 8.6.

Using the differential forward approach, the Jacobian matrix for both the original fifteen degree-of-freedom, and the modified nine degree-of-freedom manipulator may now be evaluated for any appropriately sized input $\vec{r}$. For the original fifteen degree-of-freedom manipulator, the unaugmented Jacobian matrix will be 5×15. In the case of the coupled, nine degree-of-freedom manipulator, the unaugmented Jacobian will be 5×9. After forming the Jacobian matrix, singular value decomposition must be performed and the pseudo-inverse evaluated. The system should then be converged until a feasible solution is identified.

For non-singular configurations of these manipulators the associated null spaces will be ten and four dimensional respectively. To perform null space optimization on these manipulators requires perturbing a feasible solution in each of the null space basis directions. If any motion in these directions presents an improving objective function, a linear combination of the improving directions should be formulated so that the solution can proceed in the direction of greatest improvement. The solution should continue along this improving gradient until a local optimal solution is obtained.

Future motions of the manipulator can then be made while performing the pseudo-inverse and null space computations simultaneously. If the objective func-

tion is well behaved (smooth), this method typically converges to an optimum solution in less than four iterations. Figure 8.2 illustrates the optimum solution of the coupled nine degree-of-freedom manipulator where the objective function minimizes the variations of the n variable link lengths as follows:

$$L_{VAR} = \sum_{i=1}^{n}(||l_i - l_{avg}||) \tag{8.8}$$

where l_{avg} is the average of the variable link lengths.

8.1.1 Shape Control Example

This section will present a brief outline of a shape control solution for a very high degree-of-freedom modular truss manipulator. To serve as an example, this discussion will focus on the thirty degree-of-freedom manipulator shown in Fig. 8.3. This manipulator is constructed of longeron-actuated, octahedral VGT's and straight static sections that are joined in a manner that satisfies the four criteria presented in Chapter 7.

Conventional manipulator positioning methods would require the specification of thirty independent parameters. The shape control approach reduces this number to eight parameters. The specification of a goal position and orientation will utilize six of the eight parameters. The two remaining degrees-of-freedom will be used to specify the initial and final tangent vector magnitudes. Since the tangent vector magnitudes have no effect on the initial or final position and orientation, these two degrees-of-freedom can be used to generate two infinities of solutions that satisfy the given positional requirements. If desired, these two parameters could be used to search for an optimum solution with respect to some objective function.

Suppose it is required to position the manipulator end-effector at (X_d, Y_d, Z_d), with a specified end-effector orientation. For this example, all position information

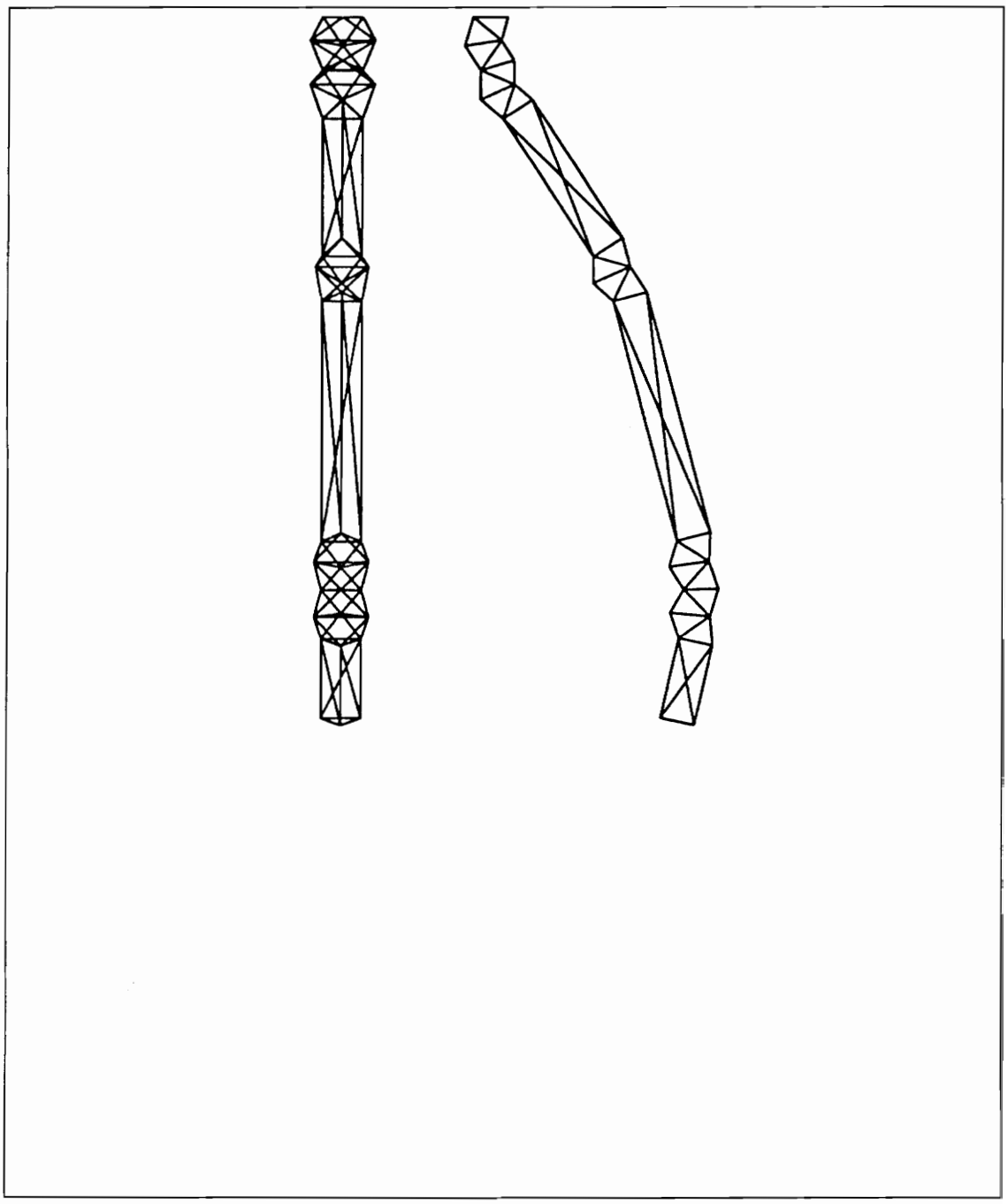


Figure 8.2: Coupled, Nine Degree-of-Freedom Manipulator, Active Link Length Variation Minimized

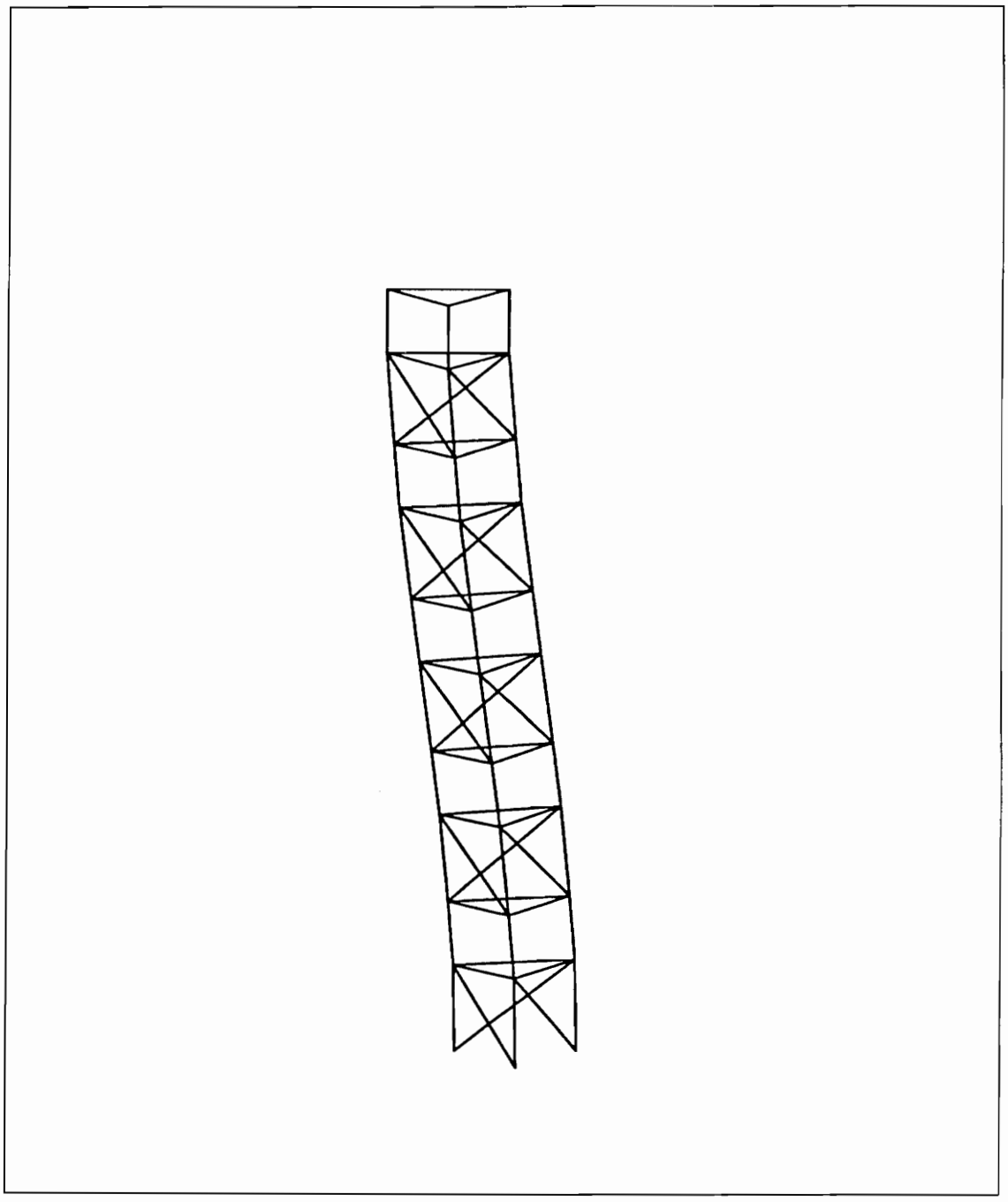


Figure 8.3: Thirty Degree-of-Freedom Modular Manipulator

is given relative to the base of the first module. Whatever form of orientation description is used, it is always possible to translate the given form into an $X - Y - Z$ Euler angle description. The procedure for accomplishing this is straight forward and is outlined by Craig [12] among others. The $X - Y - Z$ Euler angle representation is used because it is particularly well suited for the generation of a swept surface. In short this rotation procedure can be outlined as follows:

1. Rotate about the X axis by an amount γ.
2. Rotate about the resulting Y axis, Y', by an amount β.
3. Rotate about the resulting Z axis, Z'', by an amount α.

For this example, it will be assumed that the original goal orientation was given in the form of $X - Y - Z$ Euler angles, α, β, and γ. After converting the goal orientation to the $X - Y - Z$ Euler angle form, if necessary, the goal configuration may now be represented in the form of curve end conditions as:

$$\vec{C}(0) = \begin{bmatrix} X(0) \\ Y(0) \\ Z(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \vec{C}(1) = \begin{bmatrix} X(1) \\ Y(1) \\ Z(1) \end{bmatrix} = \begin{bmatrix} X_d \\ Y_d \\ Z_d \end{bmatrix} \tag{8.9}$$

$$\vec{C}'(0) = \frac{d}{du}\vec{C}(u)\Big|_0 = t_0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \vec{C}'(1) = \frac{d}{du}\vec{C}(u)\Big|_1 = t_1 \begin{bmatrix} \cos\beta \\ \cos\gamma \\ \sin\gamma \end{bmatrix} \tag{8.10}$$

$$\lambda(0) = 0^\circ \qquad \lambda(1) = \alpha \tag{8.11}$$

where, t_0 and t_1 are the initial and final tangent vector magnitudes. Now that the curve end conditions have been specified the solution can proceed as outlined in

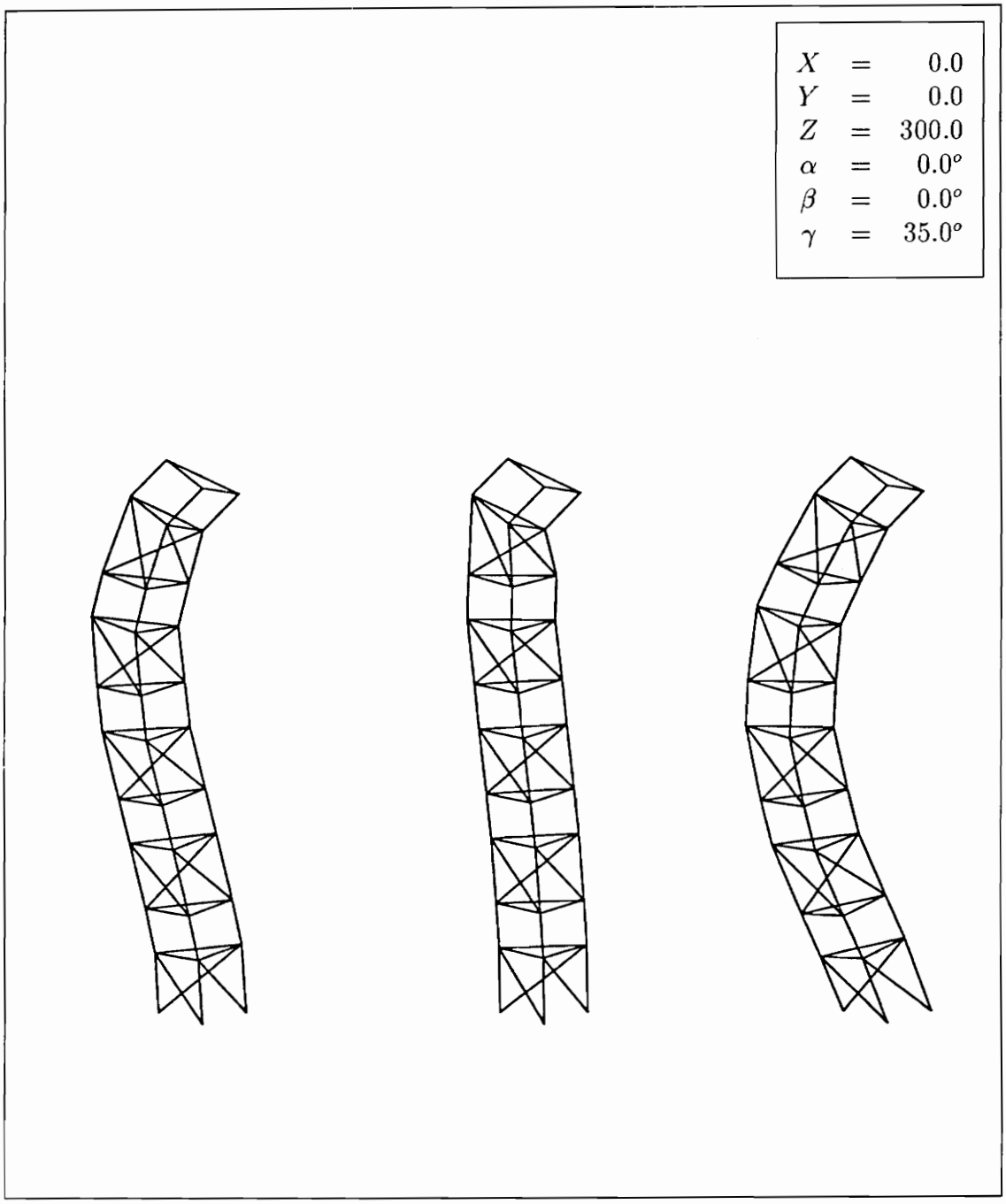


Figure 8.4: Three Shape Solutions For a Given Goal Configuration

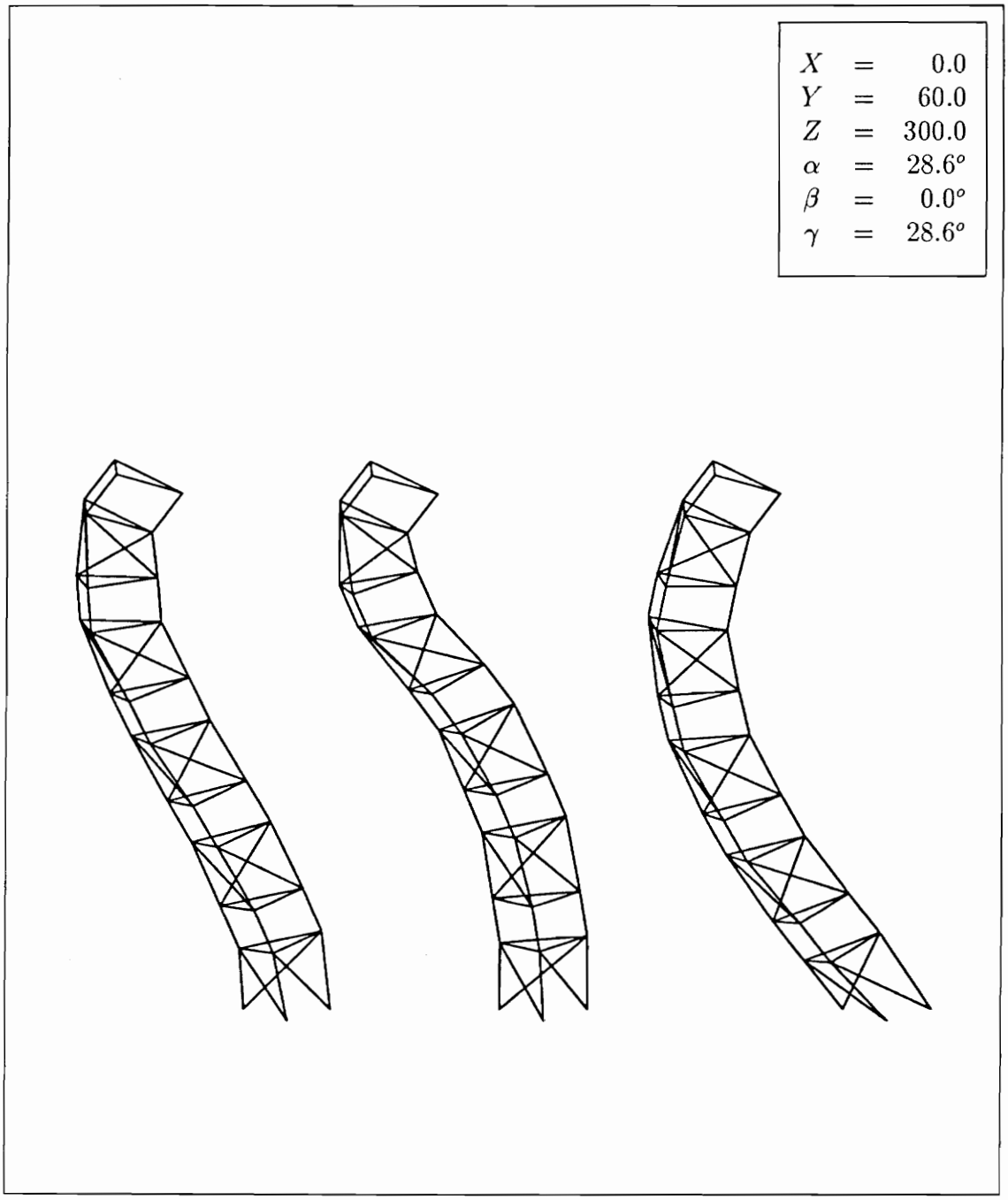


Figure 8.5: Three Shape Solutions For a Given Goal Configuration

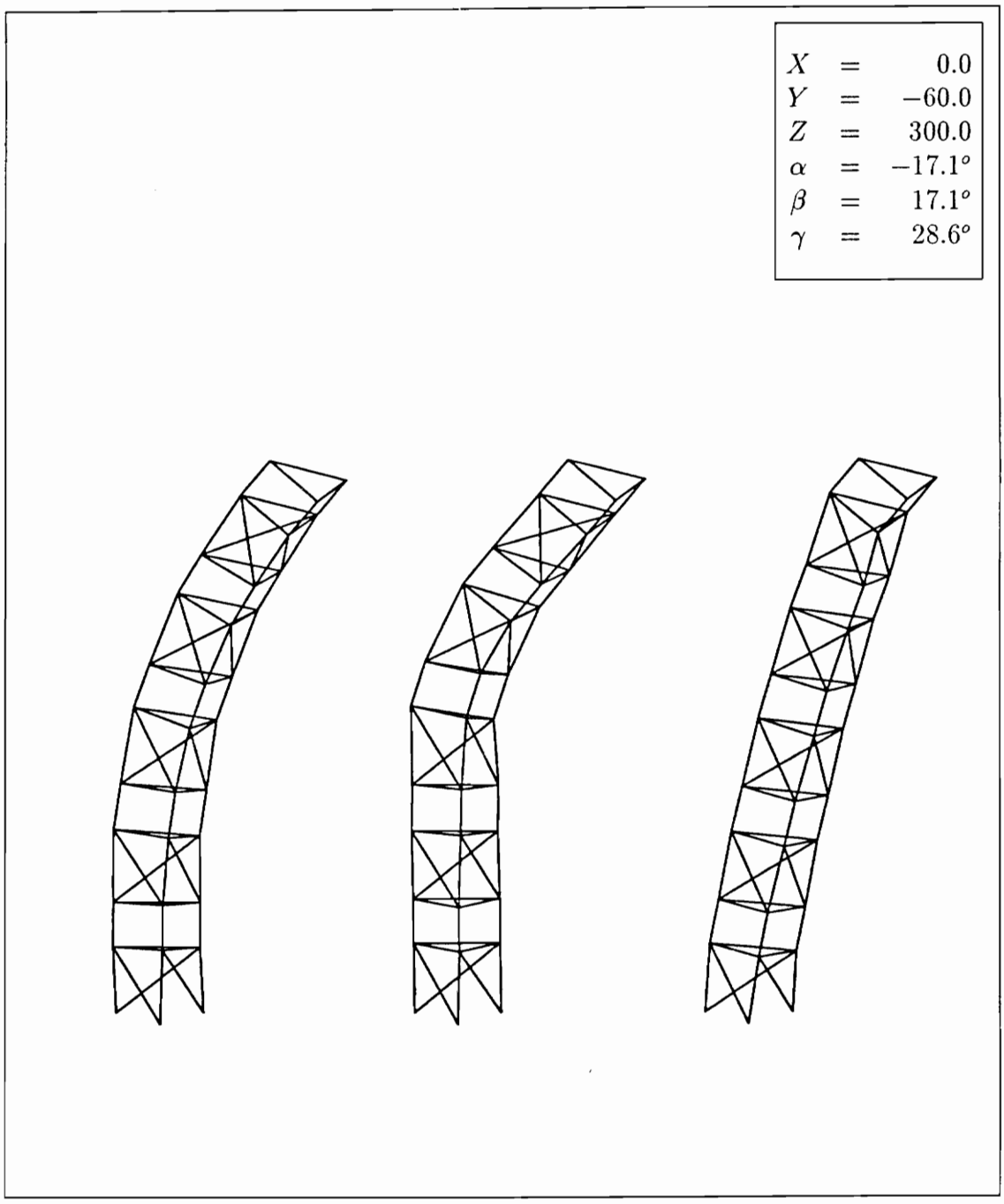


Figure 8.6: Three Shape Solutions For a Given Goal Configuration

Section 7.3 Figures 8.4-8.6 present several alternative solutions to three specified inverse positioning problems.

If required for obstacle avoidance, directrix curves of higher order or curve splines could be used to achieve the desired goal configuration. Doing this, however, requires that more parameters be specified by the user. Because of this, splines and higher order curves are discouraged unless absolutely necessary.

Chapter 9

Conclusions and Recommendations for Further Research

Conclusions

This dissertation can be divided into two distinct topics; the proposal of a new type of long-reach manipulator, and the development of the associated kinematic algorithms to control the positioning of this device. The attributes, both positive and negative, of this new manipulator system are summarized in the following paragraphs.

The proposed manipulator system is composed of modules of Variable Geometry Truss (VGT) actuators and static truss sections. This "all truss" design ensures that the loads sustained by the primary members will be either pure tension or compression (neglecting link inertias). All of the primary members are therefore utilized in their strongest loading configuration. The result is an extremely stiff manipulator with a high strength-to-weight ratio. In addition to these properties, the modular nature and open framework of the proposed manipulator are ideally suited to the waste removal tasks for which this manipulator was originally con-

ceived. The modular structure allows the manipulator to be inserted into a limited workspace one segment at a time. This makes it possible to remove waste and clear a suitable workspace for the insertion of subsequent segments. The open framework forms a convenient passageway for conveying waste or other materials in and out of containment and also provides a conduit for power and control cabling.

The complication in building such a manipulator lies within the active modules themselves. First, to physically ensure the proper mobility of the active module, careful attention must be given to the relative motion of each adjoining truss component. This internal joint design is typically simple for conventional serial manipulators. Most conventional manipulator joints need only provide for single axis rotation or translation. A typical joint within a VGT module may be required to produce rotations about three axes. In addition to a more complicated physical design, the forward and inverse kinematic analyses of the individual modules are more complicated than for a comparable [1] serial manipulator. Chapter 3 addressed these issues and showed that satisfactory results could be obtained for all of the proposed geometries.

Although structurally superior to a conventional serial manipulator, the VGT modules possess a relatively limited workspace. One way of expanding this limited workspace is to join several modules together to form a long-chain manipulator. While this relieves the workspace limitations, it poses the problem of kinematic redundancy. Resolving this issue is the other focal point of this dissertation. Null space optimization algorithms developed in Chapter 6 are capable of quickly identifying optimal solutions to inverse kinematic problems with perhaps as many as twenty degrees of redundancy. An important characteristic of these algorithms is that they mimic the modular nature inherent in the truss design. Adding segments to, or removing segments from, the manipulator corresponds directly to adding or

[1] equivalent number of degrees-of-freedom

deleting kinematic transformations from the algorithm. Thus, a library of transformations can be established, one transformation associated with each potential segment of the manipulator system.

For manipulator systems with greater kinematic redundancy, several order reduction methods have been presented. The intent of using order reduction techniques is not be to completely resolve the kinematic redundancy, but instead to provide some control of the solution space dimension. In workspace regions where there is no danger of collisions, the solution space dimension can be reduced, and solutions generated at a faster rate. In regions that demand the unrestricted dexterity of the manipulator system, a full-dimension, or larger-dimension, solution space should be utilized.

Future Research

There remain many potentially fruitful areas of research concerning VGT manipulators. One such area is the continued development of improved computational methods for the control of these devices. One possible approach, which appears particularly well suited to these modular problems is to use the dynamic programming methods. It is hoped that a full investigation of these techniques will be accomplished in the coming year.

The analysis methods presented in this dissertation are in no way limited to the particular geometries discussed. A natural extension of this work would be to optimize the design of the modules for a given task. Or, as a subset of this greater problem, chose an optimum manipulator assembly for a given task, given only a limited selection of modules.

Other topics of interest include the development of real-time solid modeling for automated obstacle avoidance and real-time force analysis. Both of these tasks

would require developing very fast, customized software which only needs to consider straight, two-force members. Finally, much attention must be given to the physical design of these manipulators. Although several electric batten-actuated, double-octahedral modules have been built, a hydraulically actuated unit has not yet been produced. For the high-load, long-reach scenarios cited in this research, hydraulic actuation will be a necessary.

Although this research has been directed at long-reach, modular manipulators, many of the algorithms developed are applicable to a much wider class of problems. As demonstrated by the planar examples, the null space optimization procedures are applicable to any redundant manipulator system.

Bibliography

[1] Arnautovic, S.H., "Adaptive Control of Redundant Manipulators via Pseudovariables," *Proceedings of the IEEE International Conference on Robotics and Automation*," Cincinnati, OH, May 13-18, 1990.

[2] Arun, V., C.F. Reinholtz, and L.T. Watson, "Application of New Homotopy Techniques to Variable Geometry Trusses," *Transactions of the ASME Journal of Mechanical Design*, Vol. 114, No. 3, Sept. 1992, pp. 422-427.

[3] Arun, V., and C.F. Reinholtz, "Enumeration and Analysis of Variable Geometry Truss Manipulators," *Proceedings of the 1990 ASME Mechanisms Conference*, DE-Vol. 26, Chicago, IL, Sept. 16-19, 1990, pp. 93-98.

[4] Arun, V., B. Padmanabhan, K. Kolady, and C.F. Reinholtz, "Determination of the Workspace of the 3-DOF Double-Octahedral Variable-Geometry Truss Manipulator," *Proceedings of the 22nd ASME Mechanisms Conference*, DE-Vol. 45, Sept. 1992, pp. 493-500.

[5] Blume, C., et al., "A Combination of Manipulator and Robot for New Out-Door Applications in Unstructured Enviornments," *Proceedings of the IEEE International Conference on Robotics and Automation*, Piscataway, NJ, May 14-19, 1989, pp. 383-390.

[6] Carignan, C.R., "Trajectory Optimization for Kinematically Redundant Arms," *Journal of Robotic Systems*, Vol. 8, No. 2, 1991, pp. 221-248.

[7] Cheney, W., and D. Kincaid, *Numerical Mathematics and Computing*, Monterey: Brooks/Cole, 1985.

[8] Chirikjian, G.S., and J.W. Burdick, "Kinematics of Hyper-Redundant Manipulators," *Proceedings of the ASME Mechanisms Conference*, Chicago, IL, Sept. 16-19, 1990.

[9] Chirikjian, G.S., and J.W. Burdick, "A Modal Approach to the Kinematics of Hyper-Redundant Manipulators," Robotics and Mechanical Systems Technical Report No. RMS-89-3, CA Inst. of Technology, Sept., 1989.

[10] Clark, W.W., "A Planar Comparison of Actuators for Vibration Control of Flexible Structures," *Proceeding of the 1989 Structures, Structural Dynamics and Materials Conference*, Mobile, AL, April 1989, pp. 1495-1503.

[11] Colbaugh, R., and M. Jamshidi,"Robot Manipulator Control for Hazardous Waste-Handling Applications," *Journal of Robotic Systems*, Vol. 9, No. 2, 1992, pp. 215-250.

[12] Craig, J.J., *Introduction to Robotics, Mechanics and Control*, Reading: Addison-Wesley, 1986.

[13] Dongarra, J.J., C.B. Moler, J.R. Bonch, and G.W. Stewart, *Linpack User's Guide*, SIAM, Philadelphia,PA, 1979.

[14] Dorsey, J.T., "Vibration Characteristics of a Deployable Controllable-Geometry Truss Boom," NASA Technical Paper 2160, June 1983.

[15] Fichter, E.F., " A Stewart Platform-Based Manipulator: General Theory and Practical Construction," *The International Journal of Robotic Research*, Vol. 5, No. 2, Summer 1985.

[16] Golub, H., and C.F. Van Loan, *Matrix Computations*, Baltimore: Johns Hopkins UP, 1983.

[17] Griffis, M., and J. Duffy, "A Forward Displacement Analysis od a Class of Stewart Platforms," Gainsville: Center for Intelligent Machines and Robotics, Jan. 1989.

[18] Hollerbach, "Optimum Kinematic Design for a Seven Degree of Freedom Manipulator," *Proceedings from Robotics Research - The Second International Symposium*, Cambridge, MA, pp. 216-222.

[19] Hunt, K.H., "Structural Kinematics of In-Parallel-Actuated Robot Arms," *Transactions of the ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 105, 1983, pp. 705-712.

[20] Jain, S., and S. Kramer, "Design of a Variable Geometry Robot Based on an n-Celled Tetrahedron-Tetrahedron Truss," *Proceedings of the 1988 ASME Design Technology Conference*, Kissimmee, FL, DE Vol. 15-3, 1988, pp. 119-123.

[21] Kahanar, D., et al., *Numerical Methods and Software*, Englewood Cliffs: Prentice Hall, 1989.

[22] Kelmar, L., and P.K. Khosla, "Automatic Generation of Forward and Inverse Kinematics for a Reconfigurable Modular Manipulator System," *Journal of Robotic Systems*, Vol. 7, No. 4, 1990, pp. 599-620.

[23] Kelmar, L., and P.K. Khosla, "Automatic Generation of Kinematics for a Reconfigurable Modular Manipulator System," *Proceedings of IEEE International Conference on Robotics Automation*, Philadelphia, PA, Vol. 2, 1988, pp. 663-668.

[24] Krieg, S.A., et al., *Single-Shell Tank Waste Retrieval Study*, Prepared for U.S. D.O.E. Office of Enviromental Restoration and Waste Management, Richland: Westinghouse Hanford Co., June 1990.

[25] Lee, K.M., and A. Chao, "On the Analysis of a Three-Degrees-of-Freedom Manipulator," *The International Journal of Robotics and Automation*, Vol. 3, No. 2, 1988, pp. 90-96.

[26] Lossing, D.L., and A.L. Eshleman, "Planning a Common Database for Engineering and Manufacturing," *Proceedings of the SHARE XLIII Conference*, Chicago, IL, Aug. 1974.

[27] Mabie, H., and C.F. Reinholtz, *Mechanisms and Dynamics of Machinery*, New York: J. Wiley and Sons, 1987.

[28] Maciejewski, A.A., "Real-Time SVD for the Control of Redundant Robotic Manipulators," *Proceedings of the IEEE International Conference on Systems Engineering*, Fairborn, OH, Aug. 24-26, 1989, pp. 549-552.

[29] Mayorga, R.V., and A.K.C. Wong, "A Global Approach for the Optimal Path Generation of Redundant Robot Manipulators," *Journal of Robotic Systems*, Vol. 7, No. 1, 1990, pp. 107-128.

[30] Miura, K., H. Furuya, and K. Suzuki, "Variable Geometry Truss and its Application to Deployable Truss and Space Crane Arm," *Proceedings from the 35th Congress of the International Astronautical Federation*, Lausanne, Switz., Oct. 1984.

[31] Miura, K., and H. Furuya, "An Adaptive Structure Concept for Future Space Applications," *Proceedings from the 36th Congress of the International Astronautical Federation*, Stockholm, Swed., Oct. 1985.

[32] Mortenson, M.E., *Geometric Modeling*, New York: J. Wiley and Sons, 1985.

[33] Naccarato, F. and P.C. Hughes, "Inverse Kinematics of Variable-Geometry Truss Manipulators," *Journal of Robotic Systems*, Vol. 8, No. 2., 1991, pp. 249-266.

[34] Nakamura, Y., *Advanced Robotics: Redundancy and Optimization*, Reading: Addison-Wesley, 1991.

[35] Natori, M., K. Iwasaki, and F. Kuwao, "Adaptive Planar Truss Structures and Their Vibration Characteristics," *Proceedings of the 1987 Structures, Structural Dynamics and Materials Conference*, Monterey, CA, April 1987, pp. 143-151.

[36] Nayfeh, A. H., "Kinematics of Foldable Discrete Space Cranes," publication of the Department of Aerospace Engineering and Engineering Mechanics, Cincinnati: U of Cincinnati, 1985.

[37] Nikravesh, P.E., *Computer Aided Analysis of Mechanical Systems*, Englewood Cliffs: Prentice Hall, 1988.

[38] Noble, B., *Applied Linear Algebra*, Englewood Cliffs: Prentice Hall, Inc., 1969.

[39] Oh, S., "Inverse Kinematic Control for Redundant Manipulators," *Proceedings of the IEEE Workshop on Intelligent Control*, Troy, NY, Aug. 26, 1985, pp. 53-57.

[40] Padmanabhan, B., "Design of Robotic Manipulators Using Variable Geometry Trusses as Joints," *Proceedings of the First National Applied Mechanisms and Robotics Conference*, Vol. 11, 1989, paper 89 AMR-8C-7.

[41] Padmanabhan, B., V. Arun, and C.F. Reinholtz, "Closed-Form Inverse Kinematic Analysis of Variable Geometry Truss Manipulators, " *Transaction of the ASME Journal of Mechanical Design*, Vol 114, No. 3, Sept. 1992, pp. 438-443.

[42] Padmanabhan, B., P.H. Tidwell, R.J. Salerno, and C.F. Reinholtz, "VGT-Based Gimbals: Practical Construction and General Theory," *Proceedings of the 22nd ASME Mechanisms Conference*, DE-VOL 46, Sept. 1992, pp. 437-443.

[43] *PC-MATLAB for MS-DOS Personal Computers*, South Natick: The MathWorks, Inc., 1989.

[44] Press, W.H., et al., *Numerical Recipes in C*, New York: Cambridge UP, 1988.

[45] Reinholtz, C.F., and D. Gokhale, "Design and Analysis of Variable Geometry Truss Robots," *Proceedings of the 10th Applied Mechanisms Conference*, New Orleans, LA, Dec. 6-9, 1987.

[46] Rhodes, M.D., and M.M. Mikulas Jr., "Deployable Controllable Truss Beam," NASA Technical Memorandum 86366, June 1985.

[47] Robertshaw, H.H., and C.F. Reinholtz, "Variable Geometry Trusses," *Proceedings of the US Army Research Office Workshop on Smart Materials, Structures and Mathematical Issues*, Sept. 15-16, 1988.

[48] Salerno, R.J., *Shape Control of High Degree-of-Freedom Variable Geometry Truss Manipulator*, Thesis, Virginia Polytechnic Inst. and State U, Blacksburg: VPI & SU, 1989.

[49] Salerno, R.J., C.F. Reinholtz, and H.H. Robertshaw, "Shape Control of High Degree of Freedom Variable Geometry Trusses," *Proceedings of the NASA Workshop on Controls/Structures Interaction*, Williamsburg, VA, July 11-13, 1988.

[50] Salerno, R.J., C.F. Reinholtz, and S.G. Dhande, "Kinematics of Long-Chain Variable Geometry Truss Manipulators," *Advances in Robot Kinematics*, Eds. S. Stifter and J. Lenarcic, Vienna, Aus.: Springer-Verlag, 1991, pp. 179-187.

[51] Salerno, R.J., H.H. Robertshaw, C.G. Horner, W.J. Milsap, D.W. Bennett, "Proposed use of a Variable Geometry Truss Manipulator for Radioactive

Waste Romoval from Underground Stroage Tanks," *Proceedings of the 1992 Waste Management Conference*," Tucson, AZ, March 1992.

[52] Sincarsin, W.G., "Trussarm Conceptual Design," Dynacon Report 28-611/0402, April 1987.

[53] Sincarsin, W.G., and P.C. Hughes, "Trussarm Candidate Geometries," Dynacon Report 28-611/0401, April 1987.

[54] Sklar, M., and D. Tesar, "Dynamic Analysis of Hybrid Serial Manipulator Systems Containing Parallel Modules," *Transactions of the ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 105, No. 2, June 1988, pp. 109-115.

[55] *Spar Aerospace Limited Applies Advanced Technology to Environmental Restoration Projects*, Brampton, Ontario: Spar Aerospace Limited, [1992?].

[56] Stewart, D., "A Platform with Six Degrees of Freedom," *Proceedings of the Institute of Mechanical Engineers*, Vol. 180, Part 1, No. 15, 1965-1966, pp. 371-386.

[57] Stoer, J., and R. Bulirsch, *Introduction to Numerical Analysis*, Trans. R Bartel, et al., New York: Springer-Verlag, 1993.

[58] Sugimoto, K., "Kinematic and Dynamic Analysis of Parallel Manipulators by Means of Motor Algebra," *Transactions of the ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 109, No. 1, Mar. 1987, pp. 3-7.

[59] Sugimoto, K., "Computational Scheme for Dynamic Analysis of Parallel Manipulators," *Transactions of the ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 111, No. 1, Mar. 1989, pp. 29-33.

[60] Suh, K.C., and J.M. Hollerbach, "Local versus Global Torque Optimization of Redundant Manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation*, Raleigh, NC, March 31 - April 3, Vol. 2, 1987, pp. 619-624.

[61] Tesar, D., and D.J. Cox, "The Dynamic Modeling and Command Signal Formulation for Parallel Multi-Parameter Robotic Devices," Gainsville: Center for Intelligent Machines and Robotics, 1981.

[62] Tidwell, P.H., "Design and Construction of a Double-Octahedral Variable Geometry Truss Manipulator," Master's Thesis, Virginia Polytechnic Inst. and State U, Blacksburg: VPI & SU, July 1989.

[63] Tidwell, P.H., C.F. Reinholtz, H.H. Robertshaw, and C.G. Horner, "Kinematic Analysis of Generalized Adaptive Trusses," *Proceedings of the First Joint US/Japan Conference on Adaptive Structures*, Maui, Hawaii, Nov. 13-15, 1990.

[64] Wynn, R.H. Jr., H.H. Robertshaw, and C.G. Horner, "An Analytical Study of a Six Degree of Freedom Active Truss for use in Vibration Control," Proceeding of the AIAA Structures, Structural Dynamics, and Materials Conference, Long Beach, CA, April 1990, AIAA Paper 90-1164-CD.

[65] Yang, D.C., and T.W. Lee, "Feasibility Study of a Platform Type of Robotic Manipulator from a Kinematic Viewpoint," *Transactions of the ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 106, June 1984, pp. 191-198.

[66] *Environmental Restoration and Waste Management Robotics Technology Development Program, Robotics Five-Year Plan*, DOE/CE-0007T, Vol. 1-3, Unioted States Department of Energy, Washington D.C., 1990.

Appendix A

Derivation of Cubic Coefficient Matrix

As indicated in the main body of the text, a spatial cubic curve can be represented by the following vector equation:

$$\vec{C}(u) = \vec{A}_0 + \vec{A}_1 u + \vec{A}_2 u^2 + \vec{A}_3 u^3. \tag{A.1}$$

Differentiating this with respect to u yields,

$$\vec{C}'(u) = \vec{A}_1 + 2\vec{A}_2 u + 3\vec{A}_3 u^2. \tag{A.2}$$

If these two equations are evaluated for the beginning and end of the curve, at $u = 0$ and $u = 1$, the four following simultaneous equations result.

$$\begin{aligned} \vec{C}(0) &= \vec{A}_0 \\ \vec{C}(1) &= \vec{A}_0 + \vec{A}_1 + \vec{A}_2 + \vec{A}_3 \\ \vec{C}'(0) &= \vec{A}_1 \\ \vec{C}'(1) &= \vec{A}_1 + 2\vec{A}_2 + 3\vec{A}_3 \end{aligned} \tag{A.3}$$

These equations can be represented in matrix form as;

$$\begin{bmatrix} \vec{C}(0) \\ \vec{C}(1) \\ \vec{C}'(0) \\ \vec{C}'(1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} \vec{A}_0 \\ \vec{A}_1 \\ \vec{A}_2 \\ \vec{A}_3 \end{bmatrix} \tag{A.4}$$

Now the cubic coefficients $\vec{A}_0, \vec{A}_1, \vec{A}_2$, and $\vec{A}_3$ can be evaluated in terms of the known parameters $\vec{C}(0), \vec{C}(1), \vec{C}'(0)$, and $\vec{C}'(1)$. Pre-multiplying both sides of the equation by the inverted coefficient matrix yields:

$$\begin{bmatrix} \vec{A}_0 \\ \vec{A}_1 \\ \vec{A}_2 \\ \vec{A}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix}^{-1} \begin{bmatrix} \vec{C}(0) \\ \vec{C}(1) \\ \vec{C}'(0) \\ \vec{C}'(1) \end{bmatrix}; \tag{A.5}$$

which reduces to,

$$\begin{bmatrix} \vec{A}_0 \\ \vec{A}_1 \\ \vec{A}_2 \\ \vec{A}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} \vec{C}(0) \\ \vec{C}(1) \\ \vec{C}'(0) \\ \vec{C}'(1) \end{bmatrix}, \tag{A.6}$$

or, by taking the transpose:

$$\begin{bmatrix} \vec{A}_0 & \vec{A}_1 & \vec{A}_2 & \vec{A}_3 \end{bmatrix} = \begin{bmatrix} \vec{C}(0) & \vec{C}(1) & \vec{C}'(0) & \vec{C}'(1) \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \tag{A.7}$$

Thus, any point on the curve, within the interval $u[0,1]$ can be represented as follows:

$$\vec{C}(u) = \begin{bmatrix} \vec{C}(0) & \vec{C}(1) & \vec{C}'(0) & \vec{C}'(1) \end{bmatrix} [\mathbf{F}] \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix}. \tag{A.8}$$

where,

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix}. \tag{A.9}$$

Vita

Robert J. Salerno was born in Rockville, Maryland on July 13, 1965. He earned a bachelors degree in Mechanical Engineering from Virginia Polytechnic Institute and State University (VPI&SU) in June of 1987. During his four years as an undergraduate, Robert worked summers and all other academic breaks at the National Bureau of Standards (now the National Institute of Standards and Technology), where he worked for the Sensory Interactive Robotics Group on developing several robotic systems.

Robert returned to VPI&SU in the Fall of 1987 to pursue a Masters of Science degree. The focus of his master's thesis was on developing inverse kinematic algorithms for the control of continuously articulated, long-chain VGT manipulators. This was the first published work on shape control concepts. The initial investigations were funded with support from the National Science Foundation and the National Aeronuatics and Space Administration.

Robert plans to complete the requirements for the degree of Doctor of Philosophy in Mechanical Engineering in November of 1993. After finishing this degree, Robert intends to remain at VPI&SU to continue post-doctoral research relating to the application of long-reach, variable geometry truss manipulators for the environmental restoration of nuclear storage facilities. This research will be funded as a result of the interest generated by this doctoral dissertation.