# Concept Development for Lightweight Binary-Actuated Robotic Devices, with Application to Space Systems

by

Matthew D. Lichter

B.S. with Honors, Mechanical Engineering
The Pennsylvania State University, 1999

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

June 2001







Signature of Author .....
Department of Mechanical Engineering
May 11, 2001

Certified by .....
Steven Dubowsky
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by .....
Ain A. Sonin
Chairman, Department Committee on Graduate Students

Concept Development for Lightweight Binary-Actuated Robotic Devices, with Application to Space Systems

by
Matthew D. Lichter

Submitted to the Department of Mechanical Engineering
on May 11, 2001 in Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Mechanical Engineering

## ABSTRACT

Exploratory space missions of the future will require robotic systems to lead the way by negotiating and mapping very rough terrain, collecting samples, performing science tasks, and constructing facilities. These robots will need to be adaptable and reconfigurable in order to achieve a wide variety of objectives. Conventional designs using gears, motors, bearings, encoders, and many discrete components will be too complex, heavy, and failure-prone to allow highly-reconfigurable systems to be feasible.

This thesis develops new concepts that may potentially enable the design of self-transforming space explorers. The vision of this research is to integrate compliant bistable mechanisms with large numbers of binary-actuated embedded smart materials. Compliant mechanisms are lightweight and robust. Binary actuation is the idea of using an actuator in a discrete on/off manner rather than in a continuous manner. A binary actuator is easy to control and robust, and by using tens or hundreds of binary actuators, one can approximate a continuous system, much like a digital computer can approximate an analog system.

The first part of this thesis examines the fundamental planning issues involved with systems having large numbers of binary actuators. The notion of a workspace is described and applied to the optimization of a manipulator design. Methods for solving the forward and inverse kinematics are discussed in the context of this application. These methods are extended to the trajectory and locomotion planning problems. Methods for planning systems of substantial complexity are developed in the context of exploratory space robotics.

The second part of this thesis presents experimental demonstrations that examine elements of the concept. The results of several design prototypes are discussed.

Thesis Supervisor: Steven Dubowsky
Title: Professor of Mechanical Engineering

# ACKNOWLEDGEMENTS

I would like to thank the MIT Rosenblith Fellowship, the Department of Defense, the NASA Institute for Advanced Concepts, and the NASA Jet Propulsion Laboratory for their support of this research. I would like to thank collaborators on this research program, including Dr. Gregory Chirikjian, Dr. John Madden, Dr. Ian Hunter, Mr. Roy Kornbluh, and Dr. Paul Schenker. Special thanks go to all the members of the Field and Space Robotics Laboratory, whose help was greatly appreciated at one point or another – Chris for his programming help, Vivek for his pearls of wisdom, Moustapha for his collaboration, Ebraheem for his machining skills, and all the rest of the members for making the FSRL a stimulating and enjoyable research environment. And of course, thank you Dr. D for the opportunity and the mentorship to work on a great project.

Thanks especially to my family – Mom, Dad, Pete, Sharon, and Jim – whose support has always made things possible. Thanks to my friends for keeping it real and thanks to Brittany for keeping me (in)sane.

# CONTENTS

# FIGURES

CHAPTER

# 1

# INTRODUCTION

## 1.1 Introduction

This thesis presents preliminary concept development for a new lightweight, robust robotic design paradigm, with special emphasis on space applications. The work presented here represents the author's contribution to a collaborative research program being carried out at the MIT Field and Space Robotics Laboratory. This program is funded by the NASA Institute for Advanced Concepts (NIAC), an autonomous agency created to "provide an independent open forum for the external analysis and definition of space and aeronautics advanced concepts [and] complement the advanced concepts activities conducted within the NASA Enterprise" [NASA Institute for Advanced Concepts, 2001]. The NIAC is focused on supporting research that may dramatically impact the aerospace community on the 10 to 40 year time horizon and strongly seeks research programs that show promise for leapfrogging current technology development.

The design paradigm being explored and developed in this laboratory involves using compliant, elastic structures embedded with a large number of simple binary actuators. A binary actuator is one that is capable of robustly maintaining only two discrete states: an on or an off position. This is in contrast to conventional robotic devices, which use continuous actuators such as motors, hydraulics, pneumatics, etc. A binary actuator could be made of a lightweight smart material such as shape memory alloy (SMA), conducting polymer, electrostrictive polymer, piezo polymer, etc.

The use of a large number of binary actuators as opposed to a few continuous ones is analogous to the leap from analog to digital computing. By using tens, hundreds, or thousands of binary actuators, one can approximate a continuous system in dexterity and utility. Incorporating simple compliant mechanisms, rather than complex gear trains, bearings, and lubrication, vastly reduces the number of moving parts and hence increases robustness. Further design additions, such as the use of simple bistable mechanisms to reinforce the actuator commands, provide additional robustness.

While compliant mechanisms and binary actuators are not new concepts, their combination in large numbers to create robotic systems has not been studied. The goal of this study was to examine the fundamental issues and challenges involved with this compliance-based binary robotic paradigm. The research discussed here explores the planning and control issues involved with highly redundant binary systems as well as some of the preliminary challenges in building such devices.

## 1.2 Motivation

The exploration and development of the planets and moons in our solar system in the next 10 to 40 years are stated goals of NASA and the international space science community [NASA, 1998]. In order to do this, robotic systems will need to lead the way by scouting, collecting geologic samples, mapping, performing science tasks, and constructing facilities, while working in highly unstructured environments and negotiating very rough terrain [Huntsberger, et al, 2000] (see Figure 1.1).

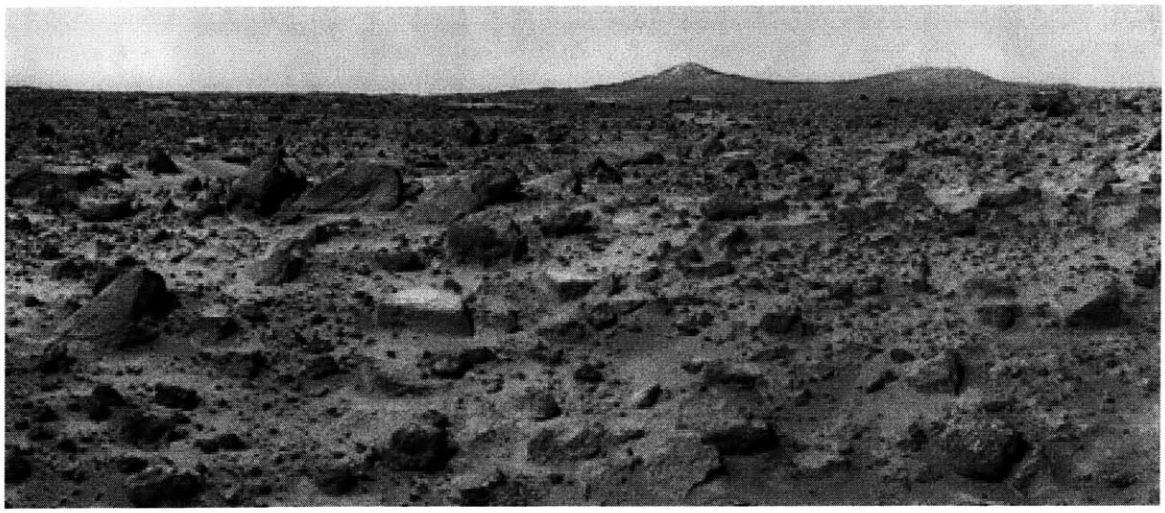

*Figure 1.1. Martian terrain as seen from the NASA Pathfinder lander, 1997. (Courtesy JPL.)*

Current planetary robotic systems do not have the capabilities to perform these missions. They are rovers or landers, built with a fixed configuration, capable of functioning in benign terrain and performing specific surveying and minor sample collection. They are composed of a large number of mechanical and electrical components such as gears, motors, bearings, encoders, and sensors. The latest planetary rover, Sojourner, and those under development are relatively conventional fixed-configuration vehicles carrying a simple mechanical manipulator [Bickler, 1992; Schenker, 1997] (see Figure 1.2). This technology, while well conceived for current and near-term science objectives, will not meet the demands of missions forecast for the new millennium. Even relatively small rocks the size of the rover itself present serious obstacles to current rovers (see Figure 1.3). They will not be able to explore rough terrain, such as cliff sides, deep ravines, and craters, where the most interesting scientific samples and information are probably located. Nor will they be able to perform even the simplest assembly or construction tasks.

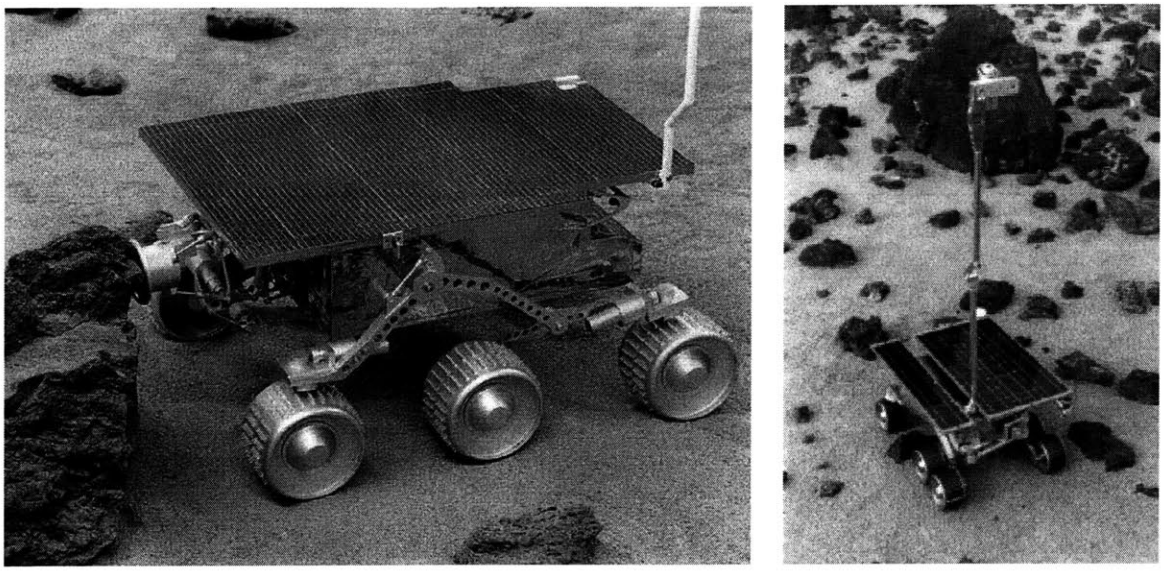

*Figure 1.2. (a) Sojourner rover that landed on Mars in 1997; (b) Rocky 7 test bed rover, one of many currently used to test technologies and concepts on Earth. (Courtesy JPL.)*

*Figure 1.3. Simulation of Sojourner, showing the difficulty in surmounting obstacles with conventional technologies. (Courtesy JPL.)*

Rather than implementing exploratory robots in a fixed-configuration manner with highly specified tasks, robots of the future will need to be adaptable, transforming themselves to meet a wide variety of objectives when needed (see Figure 1.4). A robot

that is optimized for adaptability, rather than a few specific tasks, will be much more valuable and cost-effective for missions of the next 10 to 40 years. Unfortunately, conventional components such as gear trains, motors, bearings, and encoders will render self-transformation unfeasible, due to their complexity, weight, and proneness to failure. New robot technology concepts as well as new paradigms for the design of space robots are required to meet the needs of future planetary exploration and development programs.

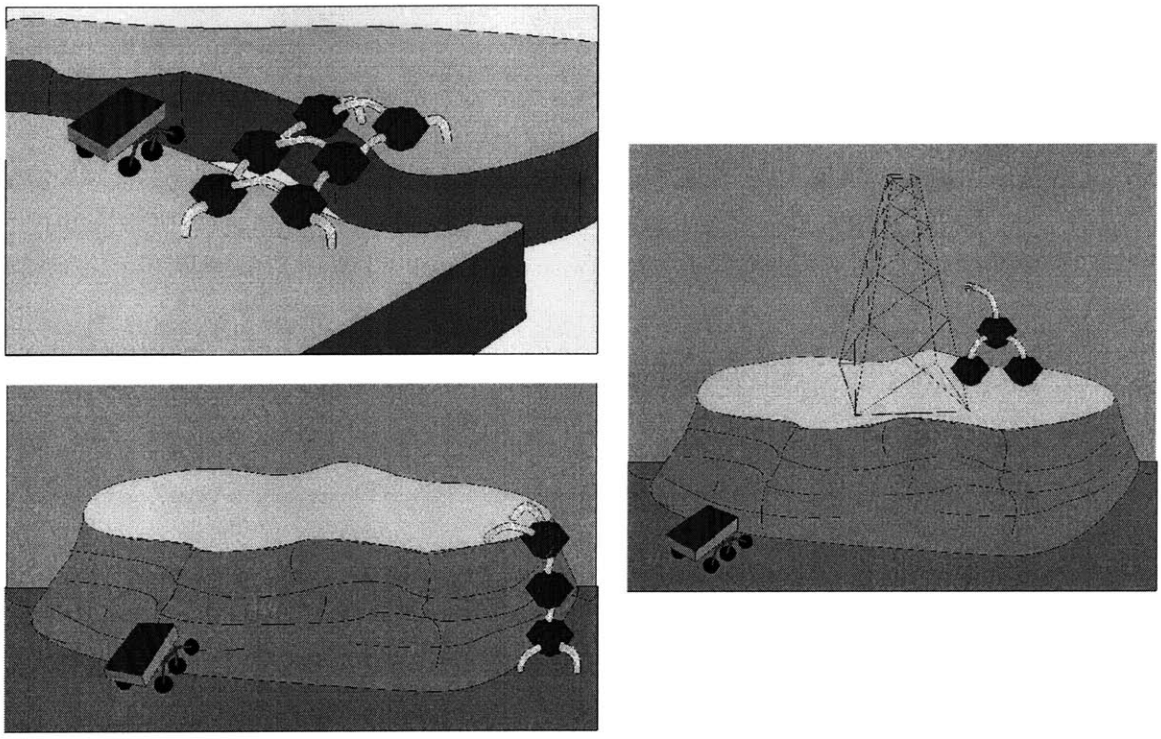

*Figure 1.4. The benefits of self-transformation. [Andrews, 2000]*

## 1.3 The Self-Transforming Explorer (STX) Concept

The thrust of the project being carried out by the MIT Field and Space Robotics Laboratory is to develop the fundamental planning and component technologies to enable the concept of self-transforming exploratory (STX) robots. Figure 1.5 shows such a robot, as envisioned by this research group, which is composed of modules interconnected by generic articulated elements [Andrews, 2000]. With such a structure, interconnections between different modules could be formed or broken to create a wide

variety of topologies, thus providing a diversity of capabilities. The articulated elements would not be composed of conventional gears and motors, but rather would be made from a compliant structure embedded with smart material actuators. The number of discrete parts would be substantially reduced, thus improving robustness in hostile environments and simplifying overall design and fabrication. Specifically, the vision laid out by this research group consists of four main points: embedded muscle-type actuators; polymer-based compliant mechanisms; binary actuation and control; and bistable joints and structures. These will be discussed in detail in the following subsections.

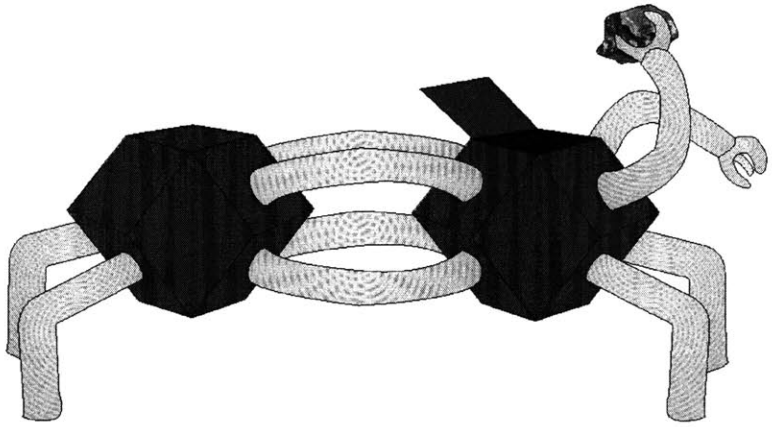

*Figure 1.5. Vision of a self-transforming explorer (STX) composed of modules and articulated elements. [Andrews, 2000]*

### 1.3.1 Embedded Muscle-Type Actuators

The first point of the STX vision is the concept of embedded muscle-type actuators. Researchers are presently developing a wide variety of smart muscle-like actuators, including conducting polymers, dielectric (electrostrictive) polymers, piezo polymers, shape memory alloys, polymer gels, and ferromagnetic polymers [Madden, et al, 2000; Pelrine, et al, 1998, 2000; Waram, 1993; Jolly, et al, 1996]. In the 10 to 40 year timeframe, many materials such as these will be developed into commercial engineering technologies. These technologies have the potential for revolutionizing a variety of mechanical systems, as they are lightweight, compact, and often lead to highly integrated designs due to their fundamental simplicity. In addition, these materials have the

potential for being very robust, as they typically consist of only a few components. Fewer components generally equates to a lower likelihood of system failure.

### 1.3.2 Polymer-Based Compliant Mechanisms

The second facet of the STX vision is the concept of replacing conventional bearings and sliding surfaces with flexures and compliant mechanisms (see Figure 1.6). The notion of compliance in mechanisms is not new, and much research has gone into developing methodologies for their design and optimization [Midha, et al, 1992; Ananthasuresh, et al, 1995; Frecker, et al, 1996, 1999]. They have seen limited use in robotic systems, however. While often not as stiff as ball bearings, they have the advantage of simplicity, light weight, low cost, zero friction, and no moving parts. Having a finite range of motion, they match nicely with muscle-type actuators, whose motions are also bounded. One can fabricate a large number of compliant joints into a single piece of material, reducing the assembly requirements for the device as well as the total part count. Methods are also being developed for embedding actuators during the fabrication of compliant mechanisms and structures [Madden, et al, 1995; Cham, et al, 1999]. Again, by their fundamental simplicity, compliant mechanisms can be very reliable, an important trait in the application of exploratory space systems.

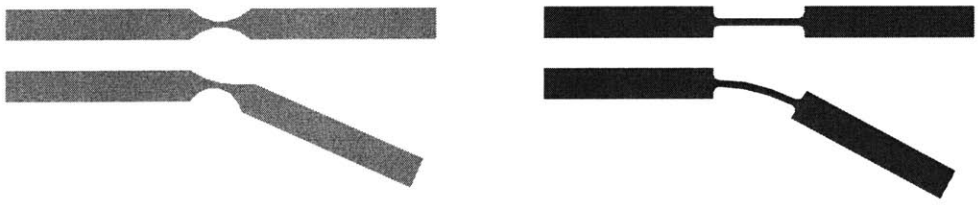

*Figure 1.6. Flexure-based elastic hinges: (a) notch flexure; (b) beam flexure.*

### 1.3.3 Binary Actuation and Control

The third component of the STX vision is the notion of binary actuation and control. Binary actuation refers to using actuators in a discrete manner, such that each actuator is capable of maintaining only two states – an on or an off position. Actuators can be designed to maintain these states reliably and precisely without the use of

feedback control. By using a large number of binary actuators, as opposed to a few continuous ones (motors, hydraulics, etc.), one can approximate a conventional system in dexterity and utility. This is analogous to the leap from analog to digital computing. Large networks of binary smart-material actuators could be embedded in compliant structures in a manner similar to the mass production of computer processors and circuit boards. In addition to their potential robustness, binary actuators lend themselves well to digital computation and control. A completely binary system avoids the complexity of analog-to-digital conversion, and control commands to the actuators are a simple 1 or 0. More importantly, control commands are insensitive to noise and precise signal commands are not necessary. That is, a binary system using digital logic will recognize any command signal above a certain threshold as a 1, and anything below as a 0. As long as command signals are kept well above or below thresholds, noise and imprecision in the actual command voltage have no affect on system behavior.

### 1.3.4 Bistable Joints and Structures

The fourth element of the STX vision is the idea of bistability in the mechanical structure and joints. A bistable mechanism is one that exhibits stability in two of its states, maintaining its configuration in the presence of disturbances [Iqbal, Pellegrino, 2000]. An example of this is a household light switch. In either the on or off position, the switch reliably maintains its mechanical position. An ideal bistable mechanism is unstable between its two stable states. The purpose of bistable mechanisms in binary robotic devices is to reinforce the desired state of the binary actuator and to increase the stiffness of the system when in a fixed configuration. When designed properly, a bistable mechanism in parallel with a binary actuator allows the controller to shut off power to the actuator, relying on the passive bistable mechanism rather than the actuator to reject disturbances. This is important for systems with a large number of actuators, when power is to be conserved.

The synergistic combination of embedded muscle-type actuators, compliant mechanisms, binary actuation and control, and bistable mechanisms provides a framework for a new design paradigm in lightweight robotic devices. This new approach

to space robotics may enable the design of increasingly complex exploratory robots, capable of adapting to a variety of mission requirements.

## 1.4 Background and Literature Review

Most of the elements of the STX vision are not new ideas and have been researched to varying degrees over the last ten to twenty years. However, to date the synergistic combination of all these elements has not been examined outside of the MIT Field and Space Robotics Laboratory [Oropeza, 1999; Dubowsky, 1999; Andrews, 2000; Lichter, et al, 2000; Sujan, et al, 2001].

The notion of using compliance to replace sliding parts is not at all new (consider a bow and arrow), but extensive research in this area over the last decade has produced very interesting results. The rapid growth of plastics use in this century has brought about a wide variety of implementations, such as tweezers, pincers, and mechanisms with a fixed (and often small) range of motion (see Figure 1.7). More recently, compliant mechanisms have been incorporated into MEMS devices, where alternative methods such as bearings are extremely difficult to build [Jensen, et al, 1997], and in surgical devices, where the simple monolithic nature of the device lends itself well to sterilization [Canfield, et al, 1999; Cappelleri, et al, 1999]. The notion of bistability in compliant and composite structures has also been explored [Jensen, 1998; Opdahl, et al, 1998; Iqbal, Pellegrino, 2000]. In recent years, automated design procedures have been developed to produce and optimize mechanism topologies and geometries based on given functional requirements [Midha, et al, 1992; Howell, et al, 1994; Murphy, et al, 1994; Ananthasuresh, et al, 1995; Frecker, et al, 1996, 1999; Saggere, et al, 1997; Hetrick, et al, 1998]. Design solutions generated by computer optimization have often led to very unusual and clever mechanisms.

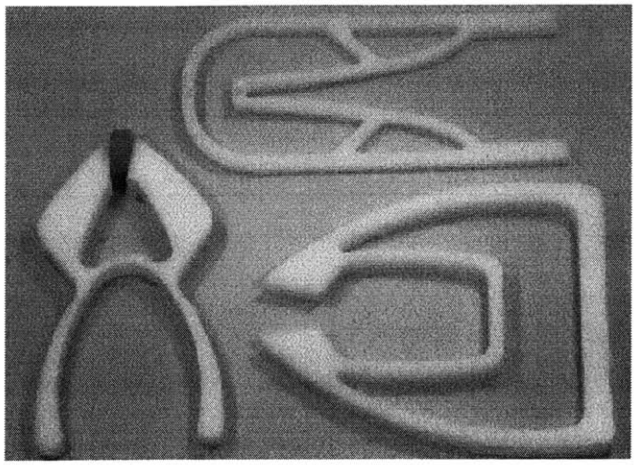

*Figure 1.7. Compliant pincer mechanisms. [Ananthasuresh]*

A very large area of research in the last ten years has been the area of artificial muscle technology. The preponderance of materials research groups around the world has led to a wide variety of potential actuation technologies. Conducting polymers and gels use the diffusion of ions into their molecular structure to change shape and volume [Madden, et al, 1995, 2000; Baughman, 1996]. Dielectric or electrostrictive polymers change shape under the electrostatic force of charge stored on parallel conductive plates [Kornbluh, et al, 1999; Pelrine, et al, 1998, 2000]. Ferromagnetic polymers change shape through the use of a magnetic field in conjunction with ferrous particles embedded in an elastomer matrix [Jolly, et al, 1996]. Shape memory alloys are metal alloys that change their crystalline structure, and thus their shape, under the application and removal of heat [Waram, 1993; Gilbertson, 1994; Youyi, Tu, 1994]. This is only a small sample of the many different materials that have been explored recently; even more will certainly follow.

The notion of embedded actuation and sensing is a somewhat new idea. Several groups have explored active damping and shape control of composite structures using embedded shape memory alloys [Baz, et al, 1994; Lee, Sun, 1995]. Similar studies have been performed with plastics and piezoelectric materials [Agrawal, et al, 1994]. The embedding of shape memory alloys and conducting polymer actuators for robotic and deployment applications has also been studied [Della Santa, et al, 1996; Wang, Shahinpoor, 1996; Huang, Pellegrino, 1996]. Methods of embedding discrete

components or actuators into plastic structures during manufacture have also been explored in a few places with interesting results [Madden, et al, 1995; Cham, et al, 1999].

Various groups have studied the concepts of modular reconfigurability, robotic cooperation, and autonomous organization in recent years. Extensive simulations and some experimental studies have demonstrated autonomous planning algorithms for general systems composed of tens or hundreds of "molecular" modules [Kawauchi, et al, 1992; Kotay, et al, 1998]. Self-assembly of systems has also been studied [Murata, et al, 1994]. Metrics for planning modular reconfiguration have been developed [Pamecha, et al, 1997]. Several groups have studied modular reconfiguration of manipulators, snake-like robots, and exploratory robots [Yim, 1995; Farritor, et al, 1996, 1997; Kotay, Rus, 1997; Hirose, 2000]. The architectures themselves of such modular systems have been a focus of considerable attention [Tesar, Butler, 1989; Farritor, 1998].

Of the different components of the STX vision, the area of binary robotics has seen the least amount of research. Some preliminary work in discretely actuated robots occurred many years ago [Pieper, 1968; Roth, et al, 1973; Koliskor 1986], but further research faded due to the lack of computation power available at the time. More recently, related research has been done in the areas of sensor-less manipulation and discrete command architectures [Erdmann, Mason, 1988; Goldberg, 1992; Canny, Goldberg, 1993]. The planning, control, and analysis of binary robotic systems has been studied primarily by a small number of research groups [Chirikjian, 1994, 1995, 1997; Ebert-Uphoff, et al, 1996; Lees, Chirikjian, 1996; Chirikjian, et al, 1998; Kyatkin, et al, 1999; Suthakorn, et al, 2000; Lichter, et al, 2000; Sujan, et al, 2001].

What has yet to be studied in great detail is the combination of all of these elements in large numbers to create a robotic device. Compliant mechanisms, embedded actuators, smart materials, bistable mechanisms, hyper-redundant binary robotics, and reconfigurable systems are all fairly well studied but have never been completely integrated by any one project. It is therefore the aim of this research program to examine systems combining all these elements.

## 1.5 Research Overview

Because the STX concept is intended for implementation in the 10 to 40 year time horizon (in accordance with the NIAC objective), it is impossible to create a complete example of the STX vision through the course of this research. However, one can demonstrate fundamental aspects of the planning, control, design, and fabrication of representative robotic subsystems. The goal of this research, then, is to use simulation studies to demonstrate feasibility for the planning and control of binary robotic systems, and to use preliminary laboratory experiments to examine issues involved in their design and fabrication. This work seeks to identify aspects of the vision that are achievable today, those that show promise in the near future, and those that will require further research and development.

## 1.6 Thesis Outline

This thesis is broken into two main chapters: one describing analytical and simulation work (Chapter 2) and the other describing preliminary experimental studies (Chapter 3). Chapter 4 presents the conclusions of the research and suggestions for future work. The appendices are used to present analytical and experimental details, which may bear value to some readers.

CHAPTER

# 2

# SYSTEM-LEVEL PLANNING, ANALYSIS, AND SIMULATION

## 2.1 Introduction

One of the fundamental challenges in implementing a binary-actuated robotic device is in its planning and control. Because binary robots consist of a large number of binary actuators, as opposed to a small number of continuous ones, many control and planning issues are fundamentally different than those of conventional robotics. Notions such as workspaces and forward kinematics have different meanings for binary-actuated robots. Dramatically different are the methods for solving the inverse kinematics and path planning problems. Instead of solving geometric equations to determine joint angles or link lengths, as one would do for a continuous robot, the inverse kinematics problem for a binary robot involves searching through a discrete set of configurations to find the one that best matches the desired state. Instead of using Jacobian matrices to compute actuator speeds for trajectory following, the problem for binary-actuated robots involves determining an acceptable sequence of actuations that achieves desired motions.

This chapter describes analysis and simulation studies performed to examine the feasibility of controlling and planning binary-actuated robotic systems in real time. In addition, this part of the study aims to develop planning algorithms that can be imported directly into physical implementations without major revision. With an understanding of the planning complexity of these systems, appropriate hardware demonstrations and physical systems can be designed.

In all simulations, graphical representations were made using OpenGL, a graphics library for use with the C/C++ programming language [Neider, et al, 1993].

## 2.2 Workspace Optimization

The workspace of a robot generally refers to the locus of all points that a robot's end-effector can reach [Craig, 1989]. With a continuous system, the workspace is usually a set of regions in continuous space (see Figure 2.1.a). In addition, many regions of the workspace are accessible in multiple orientations of the end-effector. Many continuous robots are able to achieve a continuous range of end-effector orientations for a given point in the workspace. Understanding the size of the workspace as well as the "orientability" of the end-effector within this workspace gives some measure of the ability of the robot to perform diverse manipulation tasks.

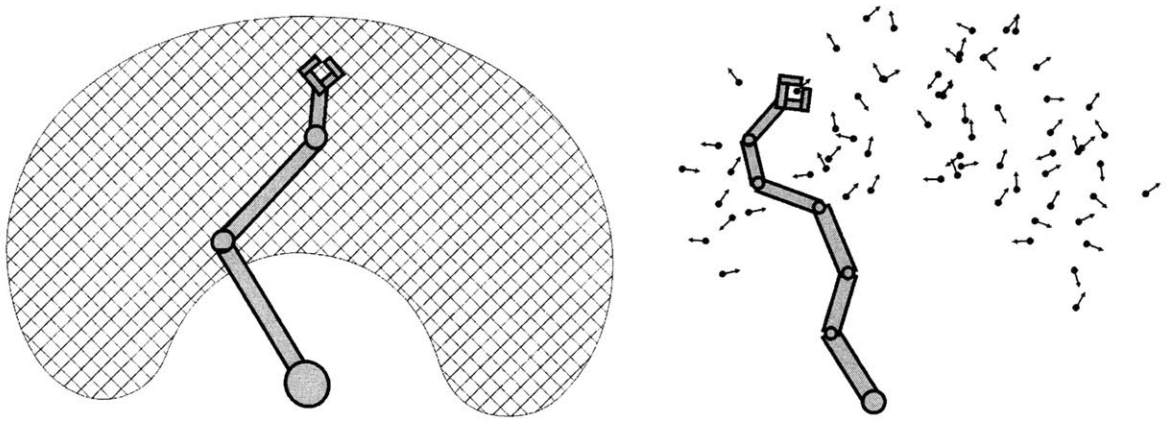

*Figure 2.1. (a) Continuous robot and workspace; (b) binary-actuated robot and discrete workspace.*

For binary-actuated robots the notion of a workspace takes on some subtle differences [Sen, et al, 1994]. For a binary system, the workspace in not a continuous volume but rather a finite set of points in space (see Figure 2.1.b). For each point there is an associated orientation of the end-effector, indicated by the arrows originating from each point in Figure 2.1.b. Thus for a given desired placement of the end-effector there is no guarantee that the robot will be able to achieve it; in fact the robot virtually never

achieves the desired placement exactly. However, there exists at least one binary configuration of the robot that achieves a minimum error of end-effector position and orientation. Thus for a binary robot, the density of the points within the workspace can be important, since a dense set of points will generally achieve small errors. The density of points increases as the number of actuators in the system increase, as each additional actuator doubles the total number of points composing the workspace. Therefore, increasing the number of actuators tends to increase the manipulability of the robot (but as will be seen in later sections, this also increases planning complexity).

With this in mind, one might want to optimize a binary robotic design so that its workspace is tailored to a given set of tasks. For example, a robot designed for a repeated pick and place task may perform best if the workspace has a great density of points in the specific pick and place locations, so that it has the greatest precision where critical. A more general robot whose tasks are quite varied might be better suited by a workspace whose points are evenly distributed throughout a large volume, thus giving it relatively uniform precision for a variety of tasks.

One aim of this study is to develop methods for optimizing binary robot designs to achieve a desired workspace distribution. In order to do this, it is important to quantify the notions of workspace uniformity and desired workspace distributions. With a quantification of how well a candidate design meets a desired workspace distribution, one can optimize the parameters of the robotic device using basic optimization methods.

For optimization and design purposes, it is useful to view a discrete workspace cloud from the perspective of a density map. For a planar robot, a density map represents the density of points (the z-axis) versus the Cartesian location in space (the x- and y-axes). With a discrete cloud, the density map appears as spikes of infinite density at each workspace point, with all other areas of the map having a value of zero density (see top of Figure 2.2). To a designer, this density representation has little information and visual value other than to depict the cloud itself. However, if one applies a low-pass filter to the density map, the spikes blend together and provide a continuous approximation of the density of the workspace (see Figure 2.2). With this continuous approximation, one can more easily create metrics for the uniformity and distribution of the workspace cloud.

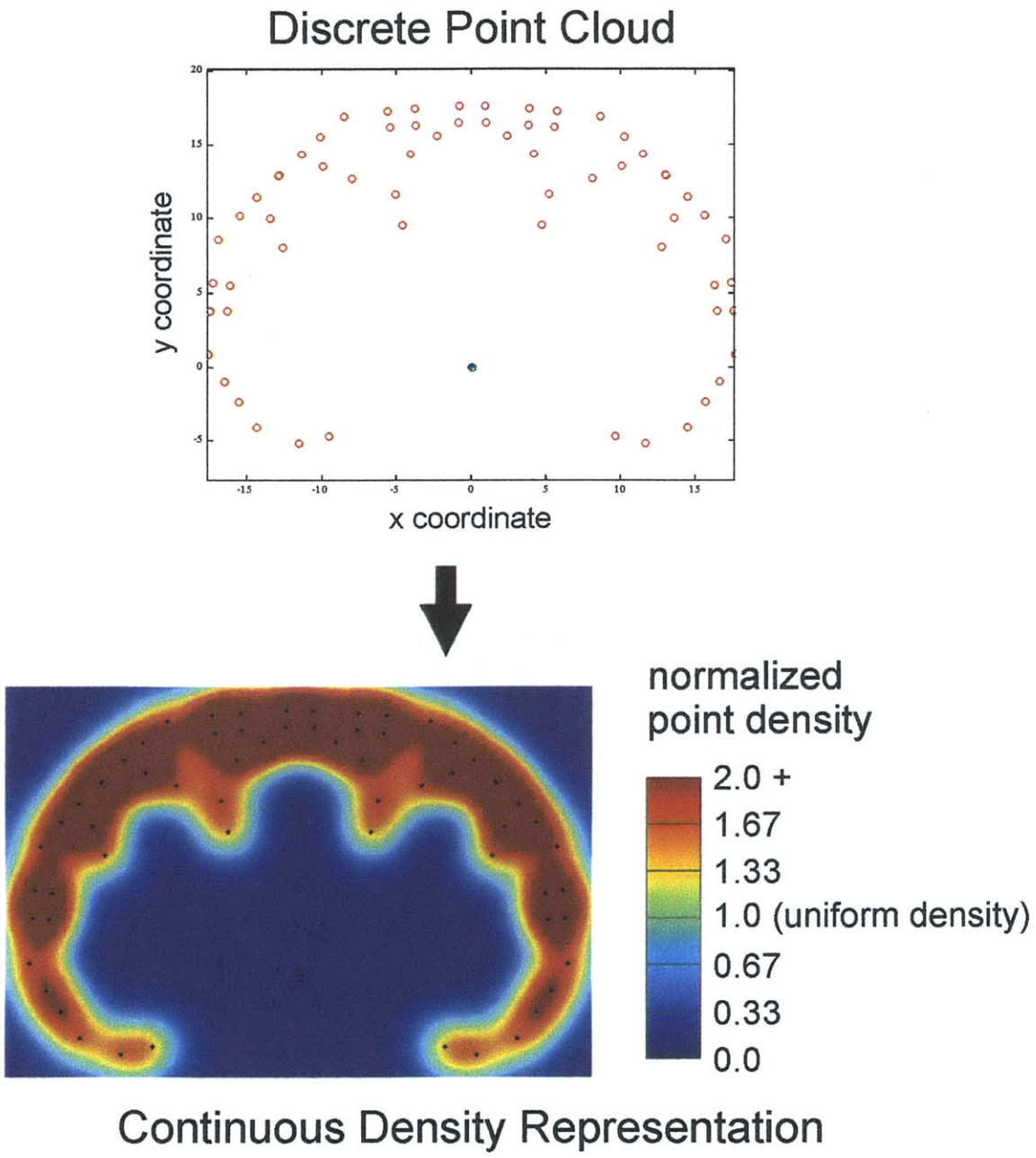


*Figure 2.2. Transformation of a discrete point cloud to a continuous density representation using a low-pass (Gaussian) filter.*

In the case studied here, the low-pass filter used was a spatial Gaussian filter, meaning that each point spike in the map was replaced by a bell-shaped peak that had the shape of a Gaussian (normal) distribution. The peak was normalized so that its height was exactly 1 unit, and the width of the peak (the standard deviation of the Gaussian) was

proportional to the square root of the workspace area divided by the total number of points. In this way, a uniformly distributed point cloud would have a continuous representation as a plateau of height 1 unit.

With a continuous density representation, one can quantify the uniformity of workspace distribution by taking the standard deviation of the z-values in the density map. A small standard deviation indicates a more uniform density distribution. This method for quantifying the distribution of the workspace can easily be extended to three-dimensional workspaces by adding a dimension to the density map. Information on the orientation of the workspace points can be included similarly by adding more dimensions to the density map. Of course this makes visualization of the map difficult, but nonetheless the notion of workspace distribution is still quantifiable.

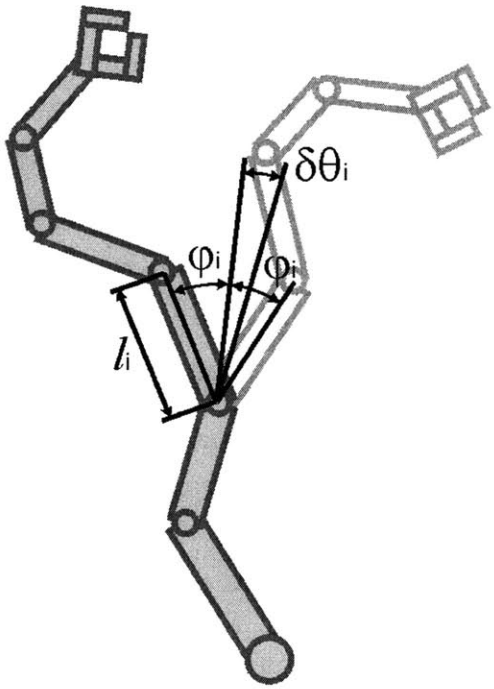


*Figure 2.3. A binary serial manipulator, showing the design variables $l_i$, $\delta\theta_i$, and $\varphi_i$ that can be optimized to provide uniform workspace density.*

With an optimization metric in place, an example case was studied to demonstrate the idea of optimizing a binary robot design to provide uniform workspace point density. For this case, a serial planar manipulator was examined, having between four and ten binary actuators (see Figure 2.3). The joint angles are operated in binary fashion,

meaning they deflect an angle of $\pm\varphi_i$ from a nominal angle $\delta\theta_i$. To be optimized were the lengths of each link, $l_i$, and the angles of deviation of each binary rotary joint, $\varphi_i$. (In this example, $\delta\theta_i$ was set equal to zero for all $i$, but this variable also could have been optimized.) This robotic design results in a planar workspace composed of $2^N$ points, where $N$ is the number of binary actuators.

A basic evolutionary algorithm was used to optimize the design variables of this design. This algorithm was not developed extensively, as it was used only to illustrate the idea of design optimization based on workspace qualities. The algorithm generated a random set of candidate designs and evaluated them based on their uniformity of workspace. It then selected the best candidates (those with the most uniform workspace densities) and mutated their variables by changing them slightly at random. This new generation of candidates was evaluated and compared to the previous generation. The best of the current and the previous generations were then kept and mutated again. The process was repeated hundreds of times until good solutions evolved. The result of one such optimization is shown in Figure 2.4. Note that the density map in this figure is much more uniform than the one shown in Figure 2.2. More sophisticated optimization methods could be developed to achieve faster convergence, and this would make an interesting study in the future.

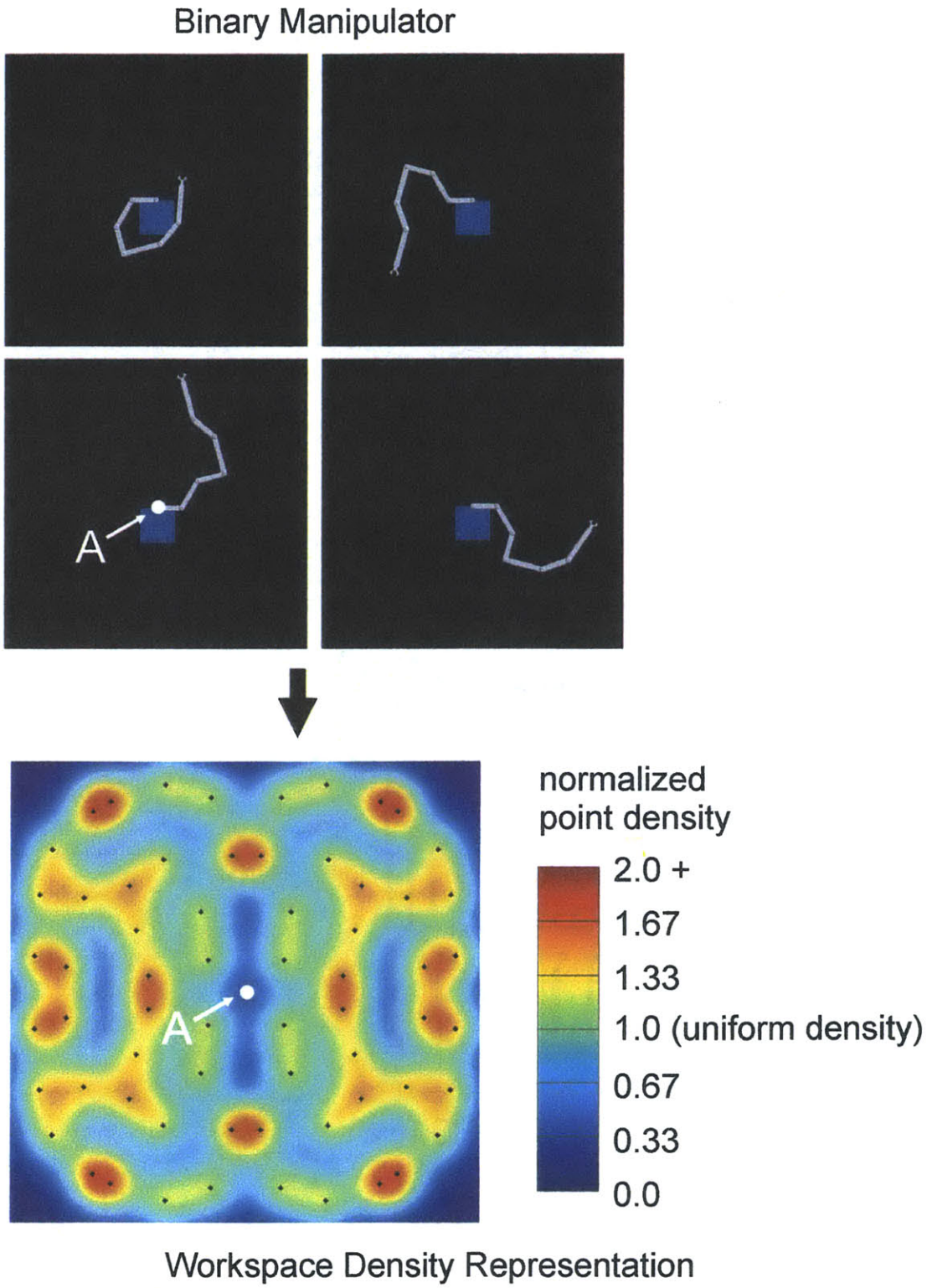


*Figure 2.4. A 6-DOF serial binary manipulator, optimized for uniform workspace density, showing a few binary configurations (top), and its workspace density map (bottom).*

## 2.3 Forward Kinematics

For binary robotic systems, it is sometimes convenient to formulate the forward kinematics using four-by-four homogeneous transformation matrices [Craig, 1989]. For example, the transformation matrix $A_{0,M}$ describing the position and orientation of the end-effector relative to the base can be viewed as the product of the $M$ intermediate transformations $A_{i-1,i}$ from module to module within the structure (see Figure 2.5). In other words,

$$A_{0,M} = A_{0,1} \cdot A_{1,2} \cdot \ldots \cdot A_{M-1,M} = \prod_{i=1}^{M} A_{i-1,i} \tag{2.1}$$

where $M$ is the number of intermediate modules. This method of solution decomposes the kinematics of a complex structure into a series of smaller, simpler structures that are easier and faster to solve [Lees, Chirikjian, 1996].

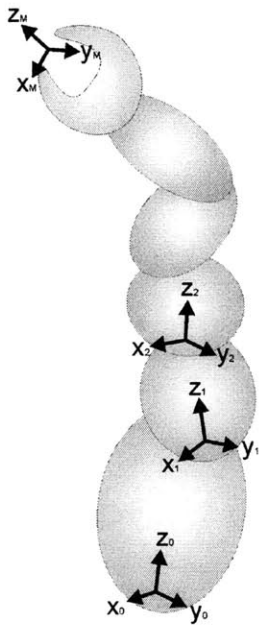


*Figure 2.5. Coordinate frames of each module within a binary device.*

Because of the discrete nature of binary devices, each term of the intermediate transformation $A_{i-1,i}$ can have only a finite number of possible values. If each module has

only a few binary degrees of freedom, one can quickly enumerate all the values that the terms of $A_{i-1,i}$ can possibly have. For example, if a module has three binary DOF, then the module has $2^3 = 8$ possible values for $A_{i-1,i}$ (notated by $A_{i-1,i}^{(1)}$, $A_{i-1,i}^{(2)}$, ..., $A_{i-1,i}^{(8)}$). The values can be computed once and stored into memory. Adding additional modules in a serial fashion will consequently increase the number of values stored in memory and thus memory requirements will grow linearly with an increasing number of modules [Lees, Chirikjian, 1996]. During run-time, forward kinematics computations need only know the binary state of the device to compute the transformation $A_{0,M}$ using Equation 2.1.

The forward kinematic computations can be simplified further in the case of a robot with similar modules. If the robotic device is composed of identical modules, each stage or module has the same kinematic characteristics and possesses the same values for $A_{i-1,i}^{(1)}$, $A_{i-1,i}^{(2)}$, ..., $A_{i-1,i}^{(S)}$. The number of computations performed and stored in memory is consequently reduced by a factor of $M$, where $M$ is the number of modules. One example of such a robot is the Binary Robotic Articulated Intelligent Device (BRAID), developed at the Field and Space Robotics Laboratory, which is a serial stack of identical parallel stages [Oropeza, 1999] (see Figures 2.6 and 2.7). Such a design could be used for manipulating instruments, collecting soil samples, or mating two cooperating robots, applications that require only moderate precision. See Appendix C for further discussion of the kinematics of the BRAID.

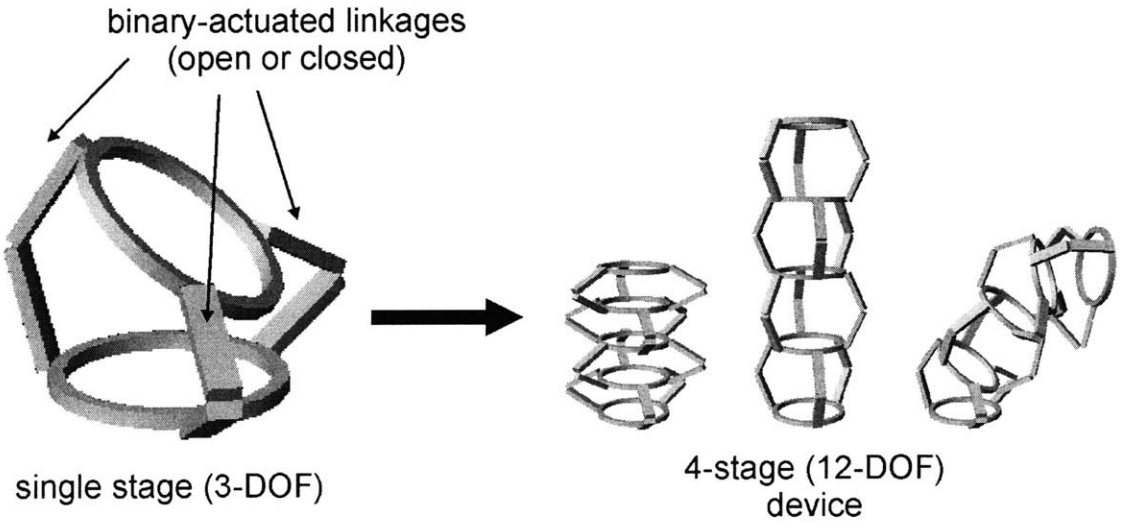


*Figure 2.6. BRAID: a serial chain of binary-actuated parallel stages.*

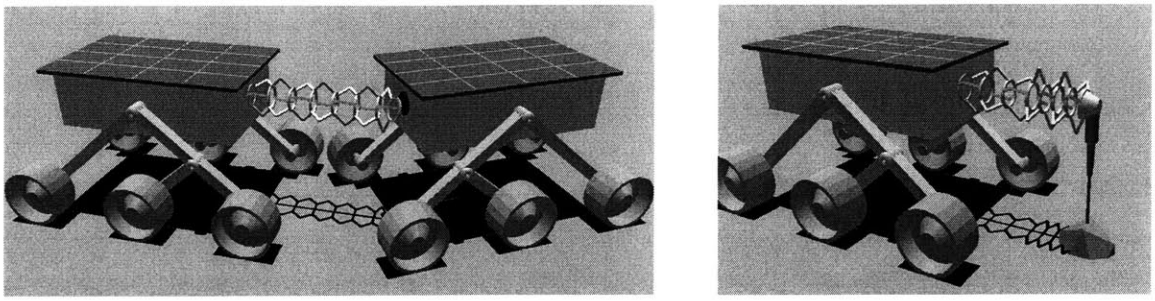

*Figure 2.7. Potential BRAID applications: (a) mating two rovers; (b) maneuvering an instrument.*

One interesting aspect of binary robotic devices is that forward kinematics computations can be done without repeated use of computationally costly transcendental functions during run-time [Lees, Chirikjian, 1996]. Whereas continuous robots have infinite solution spaces requiring complex geometric calculations be performed during run-time, binary-actuated robot geometries can be computed offline ahead of time since there are only a finite number of states involved. The solution of the module kinematics ($A_{i-1,i}^{(1)}$, $A_{i-1,i}^{(2)}$, ..., $A_{i-1,i}^{(S)}$) may of course require trigonometric or more complex mathematics, but these need only be solved once, possibly on a different computer than the one being used for real-time control. During run-time, forward kinematic computations are trivial linear algebra (Equation 2.1) based on values stored in memory. No computationally costly mathematics are required at run-time. This computation can be implemented with even the most basic processing capabilities. In fact, even these computations can be performed offline; however with high-DOF systems the memory requirements for this can become quite large. For example, a five-stage BRAID can be viewed as five identical 3-DOF modules, requiring 16 x $2^3$ = 128 floating point numbers of storage, or it can be viewed as one 15-DOF module, requiring 16 x $2^{15}$ = 524,288 floating point numbers of storage.

## 2.4 Inverse Kinematics

Most strikingly different between continuous and binary-actuated robotic devices is the solution of their inverse kinematics. Instead of using geometric relations to

compute infinitely variable joint angles or link lengths, the inverse kinematics are solved by searching through the configuration space of the binary robot to find the configuration that minimizes error between target and end-effector position and orientation. There are many ways to search through this configuration space, and particular methods are often suited to particular applications [Ebert-Uphoff, et al, 1996; Chirikjian, 1997; Sujan, et al, 2001]. One goal of this study was to examine several inverse kinematic solution methods to determine those appropriate for implementation in applications.

In all the methods discussed below, the goal is to minimize some error value, which is a function of both the positional and orientational errors. In the cases studied here, this error value was defined as

$$\begin{aligned} total\_error &= positional\_error^2 + K_{error} * angular\_error^2 \\ fitness &= \frac{1}{total\_error} \end{aligned} \tag{2.2}$$

where $K_{error}$ was a value around 100, angular error was specified in radians, and positional error was specified in percent of manipulator characteristic length (average of maximum and minimum possible lengths). Changing the value of $K_{error}$ simply shifts the weighting between positional and angular errors. The basis for this cost function was made by imagining elastic elements between the desired and actual end-effector position; total error as defined here approximates the energy stored in the elastic elements due to the error. Thus the goal for the inverse kinematics solution methods is to minimize total error (or maximize fitness).

### 2.4.1 Exhaustive Search

With modern computation speeds, an exhaustive search through the entire configuration space is possible for devices with low numbers of binary DOF. An exhaustive search algorithm typically computes the forward kinematics for the end-effector for each of the configurations, and stores this information in a look-up table in memory. At run-time, desired end-effector positions are compared to the information in

the look-up table to determine the configuration that minimizes error. With an exhaustive search, the resulting solution is globally optimal, as the entire configuration space is searched; no potential solutions are overlooked.

The problem with exhaustive searches is that they are infeasible for high-DOF systems. This is because the search space grows exponentially with the number of actuators, each additional binary DOF doubling the size of the search space. A large search space takes a long time to search through and requires a great deal of memory. For example, a 10-DOF system requires the search through and storage of $2^{10} = 1{,}024$ states, while a 20-DOF system has $2^{20} = 1{,}048{,}576$ states. In this case a system with only twice the physical complexity requires one thousand times the computation capability. In simulation studies, exhaustive searches were found to be quite effective for systems with less than fifteen DOF. Larger systems showed the exhaustive search to be too slow with modern computation for practical applications.

### 2.4.2 Combinatorial Search Algorithm

To deal with systems with larger numbers of binary actuators, a second algorithm was studied and compared to the exhaustive search method. This algorithm, denoted the *combinatorial search algorithm*, solves the inverse kinematics by changing the state of only a few actuators at a time [Lees, Chirikjian, 1996] (see Appendix A for example code). By limiting actuator state-changes to only a few at a time, rather than changing a large number of them all at once, the algorithm effectively reduces the search space to configurations that are close to the existing configuration in configuration space. This reduction in search space size over an exhaustive search is dramatic for high-DOF systems.

The first part of the search enumerates all the configurations that are within $K$ actuator state-changes (bit-flips) of the current configuration, where $K$ is a small number such as three. The algorithm then exhaustively searches these candidate configurations to find which one more closely matches the desired end-effector location. From this new configuration, the process is repeated to find an even better configuration. This process is repeated until a solution is converged upon. In this way, the combinatorial algorithm is

like a steepest-descent optimization routine, since it reaches a minimum error by taking small steps (in configuration space) in the direction that most strongly reduces error. In simulations, the number of iterations to convergence was found to be of the order of $N/3$, where $N$ is the number of DOF. The solution achieved with this algorithm is locally optimal; that is, there are no close configurations (in configuration space) that provide better results since all close configurations are exhaustively searched. While a globally optimal solution cannot be guaranteed like it can with the exhaustive search, for systems with many DOF the locally optimal solution was observed to be satisfactory for many applications.

The increase in search speed with this algorithm over the exhaustive search can be shown mathematically by examining the size of the search space. For an exhaustive search, the size of the search space is the size of the entire configuration space, which is exactly $2^N$, where $N$ is the number of binary DOF. With the combinatorial search algorithm, the search space size is exactly

$$C \cdot \left\{ \binom{N}{K} + \binom{N}{K-1} + \ldots + \binom{N}{1} + \binom{N}{0} \right\} \qquad \text{[Lees, Chirikjian, 1996]} \qquad (2.3)$$

where $C$ is the number of iterations through the algorithm (order $N/3$), $K$ is the number of bit-flips allowed per iteration, and the notation $\binom{N}{K}$ is the operator indicating the number of $K$ combinations among $N$ objects (without regard to order), defined mathematically as

$$\binom{N}{K} \equiv \frac{N!}{(N-K)!K!} \qquad (2.4)$$

Letting $C \approx N/3$, substituting Equation 2.4 into 2.3, and collecting the highest order terms, one can show that the search space size is of the order $N^{K+1}$. With small $K$, this is a dramatically smaller search space than the exhaustive search when the number of DOF is large. Therefore the combinatorial search space experiences only polynomial growth

with increasing numbers of DOF, as opposed to the exponential growth experienced by an exhaustive search.

### 2.4.3 Genetic Algorithm

A genetic algorithm was a third search method studied for solving the inverse kinematics problem [Sujan, et al, 2001] (see Appendix B for example code). Genetic algorithms have been widely used to solve unusual or difficult optimization problems [Goldberg D, 1989]. A genetic algorithm is a stochastic optimization process that often succeeds when deterministic methods are impractical or impossible. In essence, a genetic algorithm generates a random population of candidate solutions and evolves out an optimal or near-optimal solution by evaluating, selecting, crossbreeding, and mutating the individuals within the population. Depending on the application, the manner in which these processes are carried out can range from quite simple to very complex. In simulations of binary robotic devices done here, a very basic genetic algorithm was sufficient to provide good results.

One aspect of binary robots that makes the genetic algorithm very natural and convenient lies in the binary nature of the robot itself. A prerequisite for using genetic algorithms is that the optimization variables must somehow be represented in a binary DNA-like manner, so that crossbreeding and mutation can be performed on candidate solutions. With many engineering problems, the conversion from the real-world problem to a binary encoding can be complicated. With the binary inverse kinematics problem, though, these complexities do not exist. The optimization problem is already formulated in a binary code.

With a genetic algorithm, the main algorithm design variables are the population size, the number of generations of evolution, the mutation factor (likelihood of bit mutation), the crossover ratio (fraction of the selected population that is cross-bred), crossover method (1-point, n-point, uniform, etc.), and the fitness metric used for evaluating individuals in the population. In this study, many of these algorithm variables were chosen heuristically based on rules of thumb or general performance of the algorithm. In simulations, ten separately evolved populations were used, with the final

solution chosen from the best of all populations. Each of the ten populations had a size of one hundred individuals, and each population evolved over one hundred generations. These numbers were chosen because they gave reasonable performance and speed. For each new generation, a population of new individuals was selected based on the fitness of the individuals in the previous generation. The likelihood an individual was selected for the next generation was proportional to its fitness relative to the population (fitness being defined in Equation 2.2). This method of selection, the most common, is known as *proportional selection*. The crossover ratio was set to 0.5, meaning half of the selected population was crossbred, while the other half maintained its individuality. This ratio was recommended from literature as a good starting point for many applications [Goldberg D, 1989]. It provided good performance, and therefore was used. Uniform crossover was used to provide good diversity in populations (see Figure 2.8).

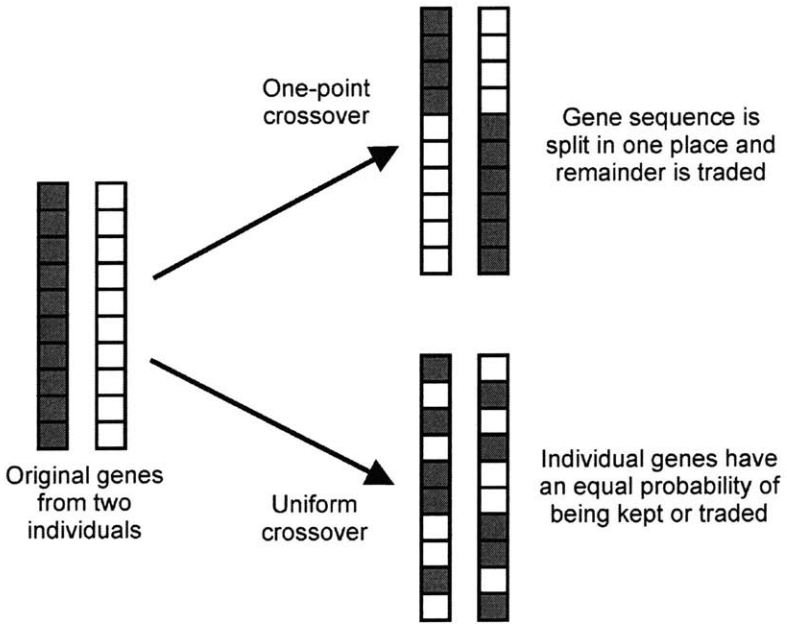


*Figure 2.8. Comparisons between basic crossover methods.*

In terms of cross-over and mutations, some modifications to the basic genetic algorithm were made that improved performance of the algorithm. The kinematic structure that was used in simulation studies was that of the BRAID (see Figure 2.6). This device is composed of a serial chain of parallel stages. Each parallel stage has three binary DOF. By the nature of its structure, individual *stages* rather than individual *bits*

should be viewed as the essential building block of the device. The genetic algorithm was seen to be more effective when working with the building blocks of the device (the stages) rather than pieces of the building blocks (the individual bits). Therefore, when the algorithm performs crossover with two individuals, it swaps stages (sets of three bits) rather than individual bits between the individuals (see Figure 2.9).

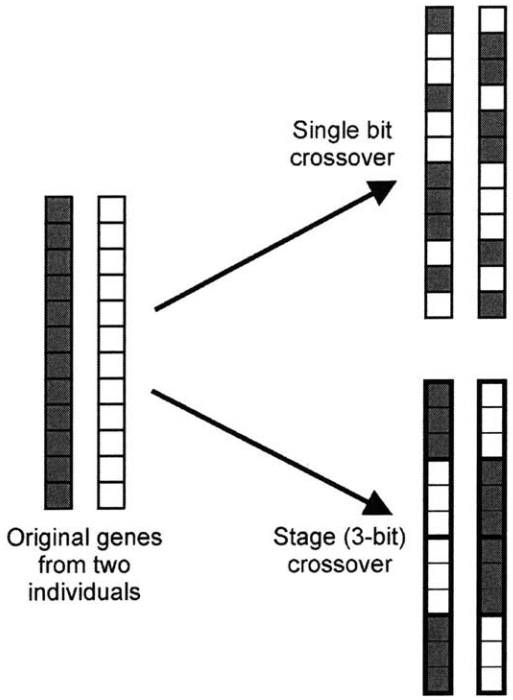


*Figure 2.9. Stage crossover vs. bit crossover.*

By the same reasoning, mutations were performed on stages rather than bits. In genetic algorithms, mutations provide random variations to individuals within the population, which helps ensure that the entire configuration space is represented. In this application, it appeared to be more effective to mutate stages (sets of three bits) rather than individual bits. The algorithm replaces the existing stage with one randomly chosen from the eight possible states the stage can have. While many phenomena of genetic algorithms are not easily explained, it is thought that the same reasoning used in the crossover discussion applies here; i.e. mutating the building blocks is more effective than mutating individual bits.

Comparing the genetic algorithm to the others discussed, the size of the search space explored by the genetic algorithm is given by

$$search\_space\_size = E \cdot G \cdot P \tag{2.5}$$

where $E$ is the number of populations separately evolved (in this case 10), $G$ is the number of generations for each population (100), and $P$ is the number of individuals within the population (100). In studies done here, $E$, $G$, and $P$ were kept constant relative to the number of degrees of freedom, $N$. For more advanced algorithm development, these values could be made a function of $N$, but this was not necessary for purposes here. The values of $E$, $G$, and $P$ used here were found to be effective for systems having as many as 150 binary DOF. In the case of constant algorithm parameters, the search space size stays constant relative to the number of DOF. Within the algorithm, several computations take place that are linearly proportional to $N$ (such as forward kinematics computations) and therefore computation time of the inverse kinematics using a genetic algorithm grows only linearly with the number of DOF of the system. As will be shown in the following section, this makes the genetic algorithm the fastest of those studied for systems having more than 40 DOF.

### 2.4.4 Algorithm Comparisons

Figure 2.10 shows the times for solving the inverse kinematics problem for each of the three algorithms described above. The times were computed from simulations based on the BRAID structure (see Figure 2.6), and were performed using a 600 MHz Pentium III processor. In these studies, the exhaustive search was observed to be the fastest for systems with less than 12 DOF, the combinatorial algorithm was the fastest for systems having between 12 and 40 DOF, and the genetic algorithm was the fastest for larger systems.

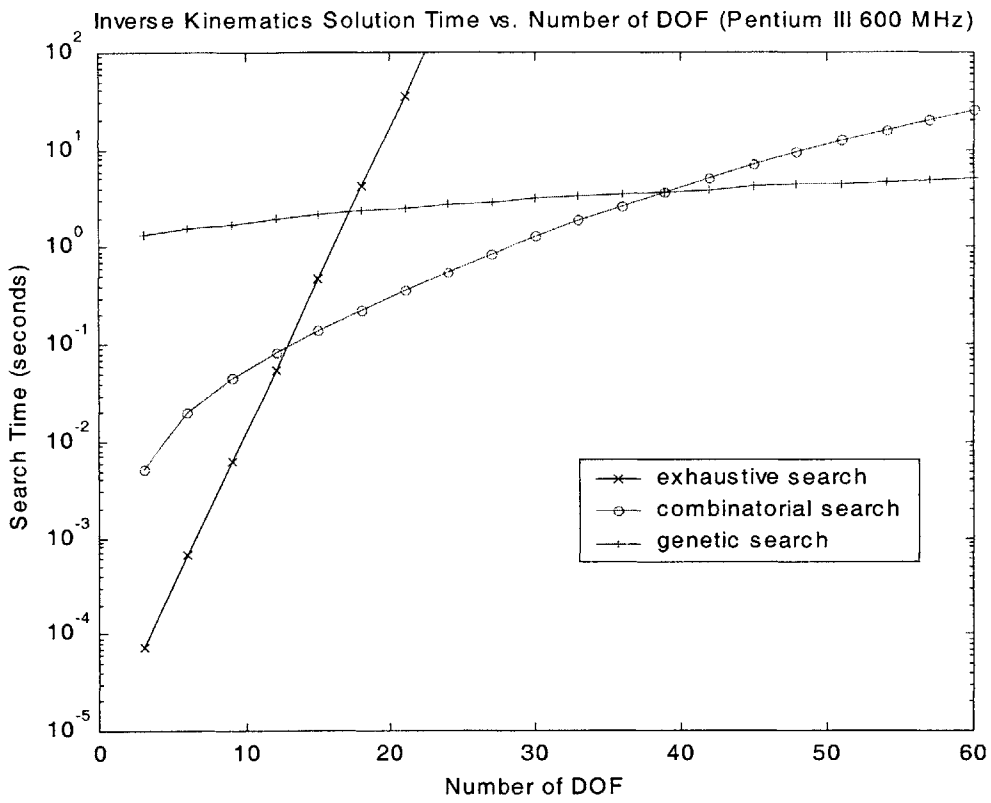


*Figure 2.10. Inverse kinematics solution times for various algorithms.*

### 4.4.5 Error Analysis of Binary Systems

Errors in position and orientation for the various algorithms can be quantified on a stochastic basis using a Monte Carlo method. The inverse kinematics for one thousand random target points were solved for and the errors were recorded and quantified on a statistical basis. The targets were chosen from within a *working workspace*, which was defined as a region roughly 90% of the radius of the actual point cloud (see Figure 2.11). The working workspace represents the region in which actual tasks would be carried out, having sufficient point density to allow good performance. (It is assumed that the periphery of the true workspace has low point density.) Each target was given a random orientation. Again, computations were made from simulations of the BRAID structure. The specific BRAID geometry analyzed here was chosen by trial and error as one that had good motion properties by inspection.

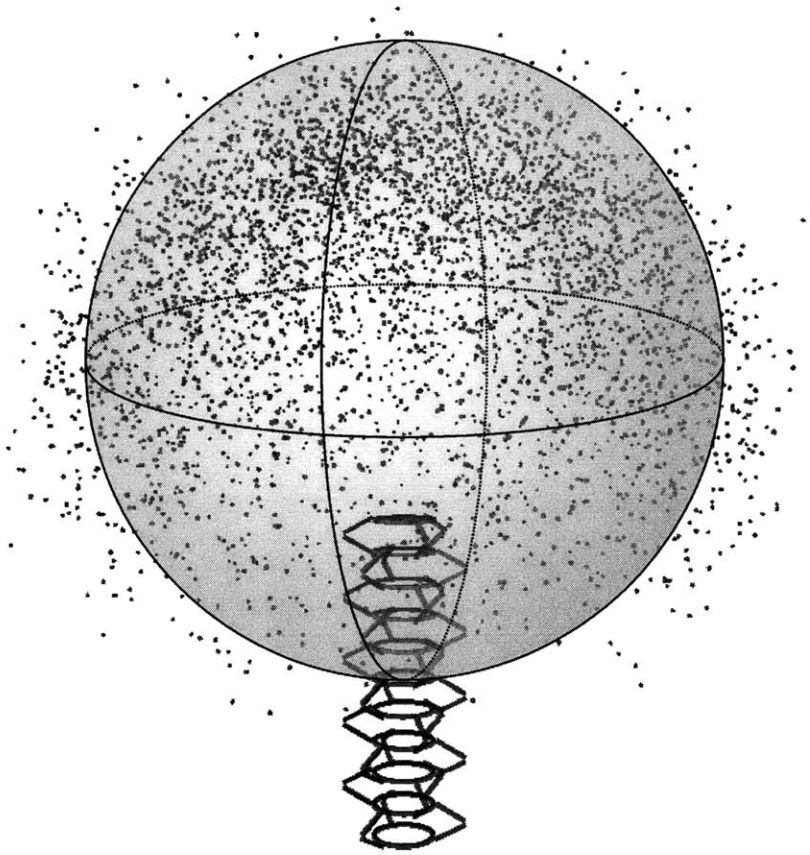

*Figure 2.11. Representative workspace of a 30-DOF BRAID. 1000 random target points were chosen from within the working workspace, a sphere whose size is slightly smaller than the actual workspace cloud.*

An example of the distribution of the errors is shown in Figure 2.12. These distributions are intended to give the designer an understanding for the size of errors encountered in binary systems and show that errors are not constant but still predictable statistically. For systems with 30 DOF (a ten-stage BRAID), displacement errors are generally within a few percent of the characteristic manipulator length and angular errors are within fifteen degrees. Such a system is unsuitable for precision work, but may be acceptable for such tasks as camera placement, crude instrument manipulation, and sample collection. For better angular precision (at the cost of displacement precision), one could simply increase the value of $K_{error}$ in the error cost function (Equation 2.2). One could also use a fine-motion stage to trim the errors involved with the binary system, although this was not a focus of this study. The shapes of the error distributions are very

similar for each of the algorithms, and most closely resemble a gamma distribution [NIST/Sematech, 2001]. The outliers are generally near the boundaries of the working workspace, and further refinement of the working workspace could eliminate these outliers and improve precision (at the cost of reducing the working envelope). Figure 2.13 shows how the median errors drop as a function of the number of DOF for the combinatorial and genetic search algorithms. Note that after about 30 DOF, there are only marginal improvements in performance with increasing number of DOF. One can also see that the genetic algorithm provides slightly better solutions than the combinatorial algorithm between 20 and 50 DOF. This is partly due to the fact that the genetic algorithm was heuristically tuned for roughly this range of DOF.

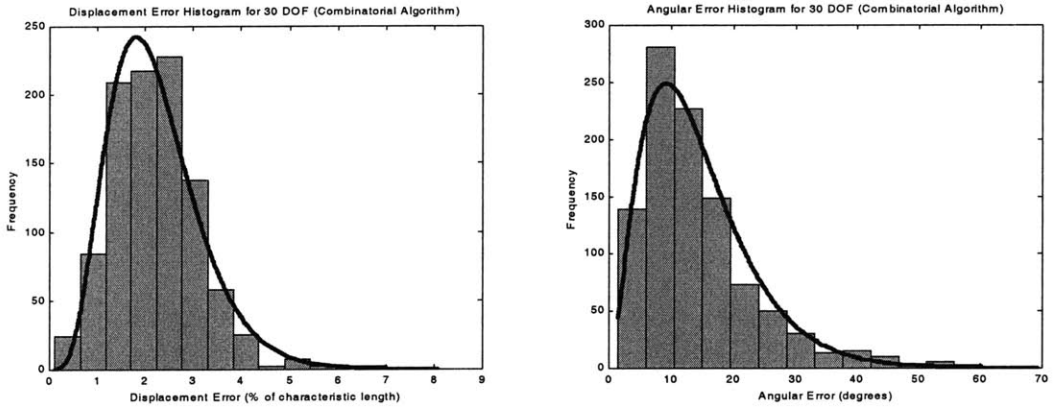


*Figure 2.12. Error distributions for a 30-DOF BRAID: displacement error (left); angular error (right). (1000 samples.)*

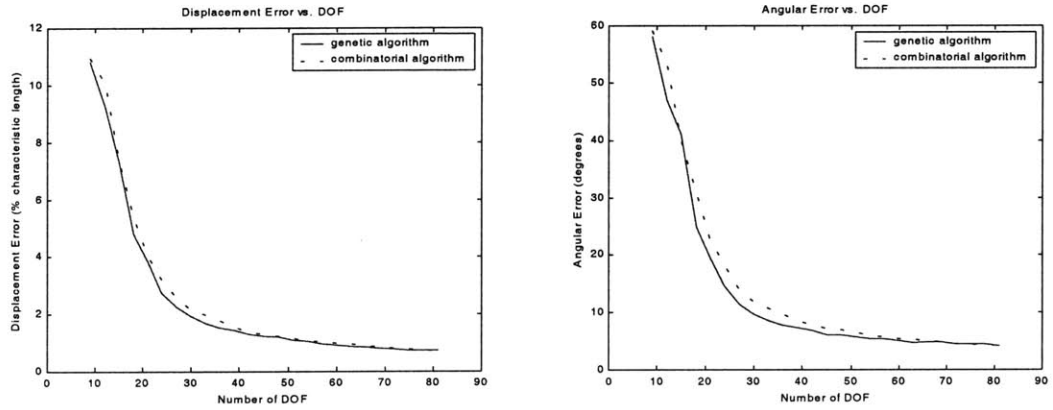


*Figure 2.13. Median errors vs. number of DOF for different algorithms: displacement error (left); angular error (right). (1000 samples per DOF.)*

These errors should not be viewed as the results of a thoroughly optimized system. Any "optimization" done here was made through qualitative inspection of systems having between 20 and 50 DOF. The same stage geometry was used for all the systems analyzed, regardless of DOF. In reality, an optimized stage geometry would be different for a 9-DOF system than a 90-DOF system. The geometry studied here was certainly not bad, but numerical optimization of all the geometric parameters would need to be carried out in order to determine more accurate limits on the errors of such systems. This is a challenging problem worthy of deeper study.

## 2.5 Trajectory Following

As was seen with the inverse kinematics problem, the trajectory following problem is strikingly different for binary devices than for continuous ones. Instead of computing Jacobian matrices based on infinitely variable geometries and using them to compute actuator velocities [Craig, 1989], the problem instead is a repeated search through the configuration space to find the configuration whose end-effector most closely matches a moving target [Lees, Chirikjian, 1996]. In this way, the trajectory following problem is very closely related to the inverse kinematics problem.

Since there are at least several ways to solve the inverse kinematics problem, it is straightforward to apply this to a moving target. With low-DOF systems, repeated use of the exhaustive search may prove to be the easiest and most robust method for trajectory following. For systems with higher DOF, one could use such methods as genetic algorithms or combinatorial searches as described above.

At first glance, a genetic algorithm may seem to be the best method for solving the trajectory following problem for systems with many binary states. It seems to be the fastest for large systems and provides good results. However, further examination shows the genetic algorithm as developed here is not well suited for the trajectory following task. A genetic algorithm is a stochastic search method; given the same target and the same initial conditions, it will produce different solutions by the fact that it is searching a subset of the configuration space based on a randomly selected initial population.

Because a binary system is very highly redundant, there are usually a large number of binary configurations that will produce nearly the same kinematic result of the end-effector, yet will have largely different configurations. That is, points neighboring in Cartesian space may be very distant in configuration space. Thus for the genetic algorithm developed here, a relatively smooth path can be planned in Cartesian space, but it may be quite erratic in configuration space (see Figure 2.14). The solution generated by the genetic algorithm developed here will likely generate a path in configuration space requiring large numbers of actuators to toggle between small time steps. For the sake of power consumption, reliability, and transient behavior, this is very undesirable. One could avoid this problem by incorporating smoothness of the configuration trajectory into the fitness metric. In such a case the fitness metric must be defined with great care so that appropriate performance and predictable algorithm behavior occurs. This is probably easier said than done, since this process would probably rely on heuristic tweaking and qualitative observations.

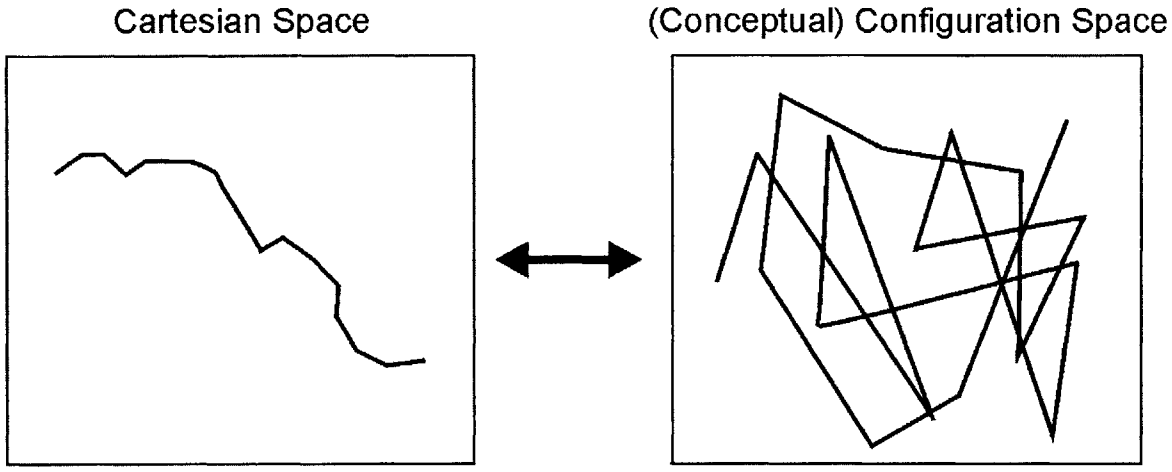


*Figure 2.14. A smooth trajectory in Cartesian space is not necessarily smooth in configuration space.*

Because of these issues, the combinatorial search algorithm was observed to be the most effective and straightforward method for solving the trajectory following problem, of the algorithms studied here. This was also observed by other researchers in a similar study [Lees, Chirikjian, 1996]. Because this algorithm searches only the subset of neighboring configurations, it intrinsically generates a path that is not only relatively

smooth in Cartesian space, but also in configuration space. Between time steps, only a few actuators will be actuated at a time (in the cases studied here, no more than three at a time). In addition, the combinatorial algorithm proves to be faster than the genetic algorithm up to one hundred DOF (see Figure 2.15). In the trajectory planning problem, the combinatorial algorithm is able to run much faster than in the pure inverse kinematics problem described in Section 2.4.2, since it only makes one iteration between time steps (as opposed to roughly $N/3$ iterations, where $N$ is the number of DOF). For its speed, reliability, and desirable behavior in configuration space, the combinatorial search algorithm was clearly the best of the algorithms studied for trajectory following.

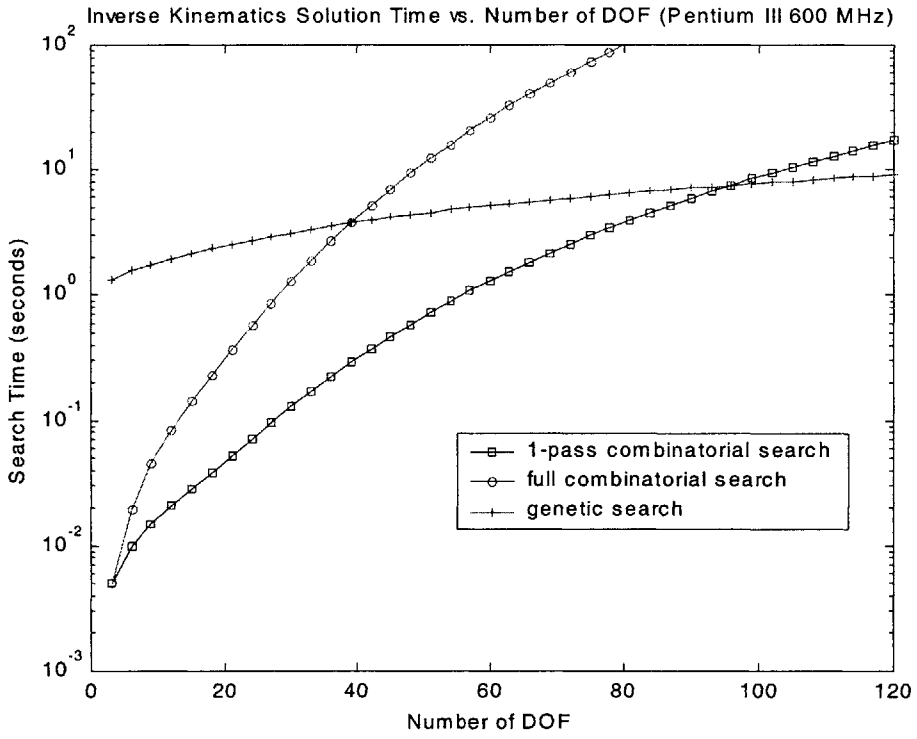


*Figure 2.15. Inverse kinematics solution times as they relate to the trajectory following problem.*

Simulations showed that the errors maintained during trajectory following were acceptable for a number of applications, such as maneuvering a camera or instrument or manipulating an object with low precision (see Figure 2.16). Typical errors during manipulation were found to be of roughly the same size as those discussed in Section 2.4.5 (see Figures 2.12 and 2.13).

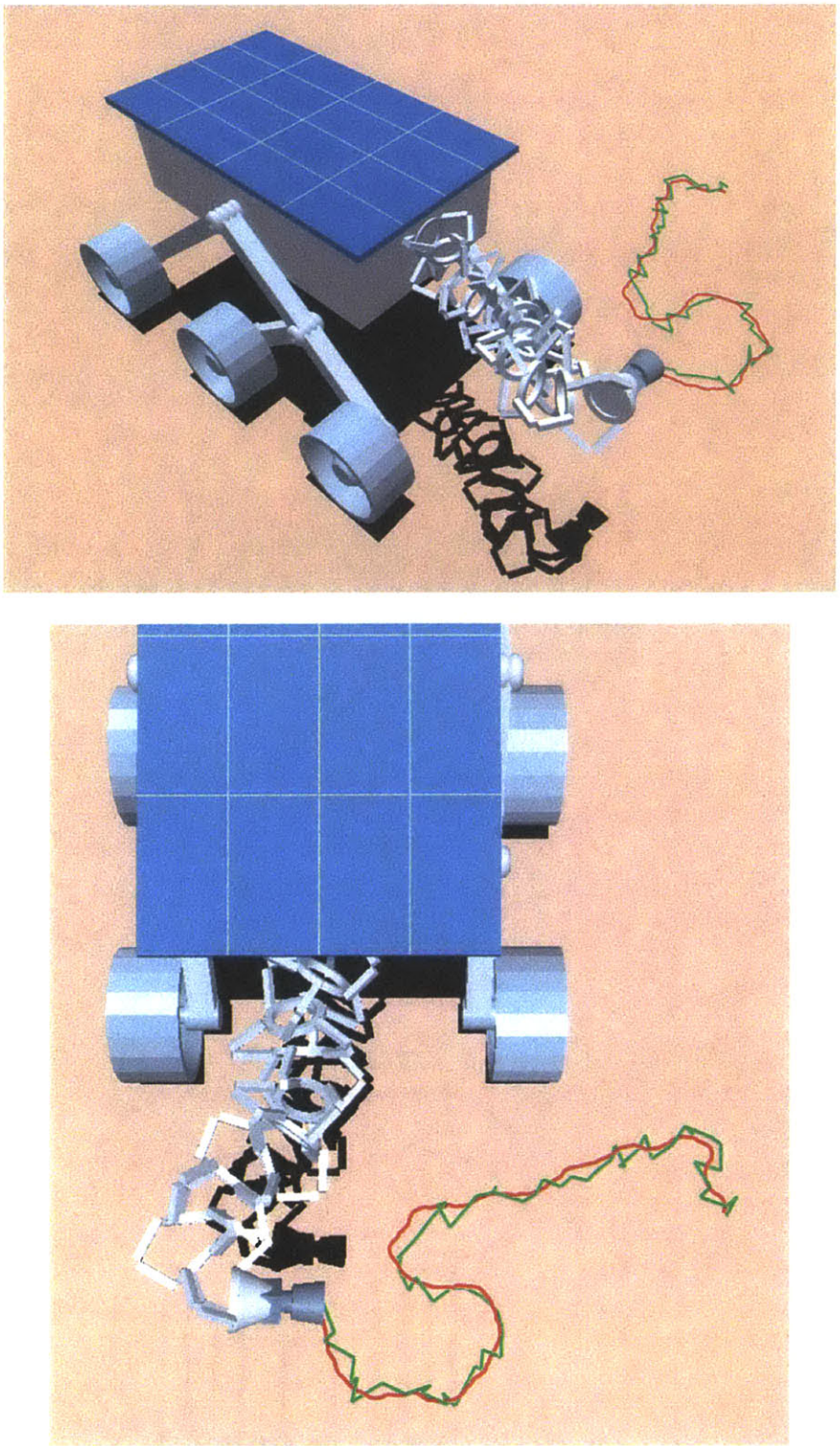

*Figure 2.16. Simulation of a camera maneuvering task using the trajectory-planning algorithm. Desired trajectory: red path; actual trajectory: green path.*

## 2.6 Locomotion Planning

The trajectory following problem can be extended to the locomotion planning problem, in the case of binary devices used as legs. Simulations were carried out to demonstrate the feasibility of planning actuator sequences in real-time for a robot having six binary-actuated legs. This locomotion scenario is a representative task of moderate computational difficulty that has foreseeable use in planetary explorers and other applications. The ability to plan such a task is prerequisite for demonstrating feasibility and utility of the binary robotic concept in real-world scenarios.

In simulation, each of the six legs was modeled as a BRAID and had 21 binary DOF (see Figure 2.17), yielding a total of 126 DOF for the system. Desired ground contact points for each leg were chosen based on the uneven terrain. Trajectory planning was performed on the legs to determine the configurations and actuator sequences required for motion of the robot.

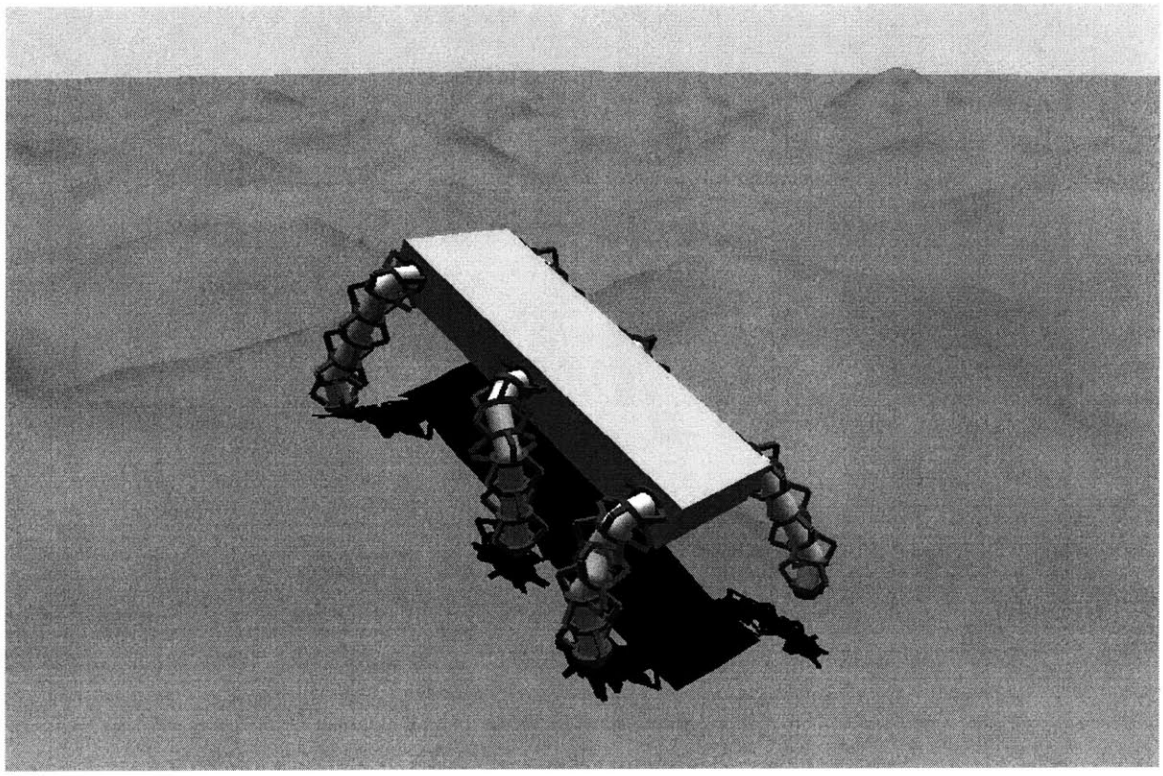

*Figure 2.17. Simulation of a 6x21-DOF walking robot composed of six BRAIDs for legs, walking in rough terrain.*

Several issues arise with a binary system such as this one. With multiple legs making contact with the ground, the robot forms a closed kinematic chain that is actually over constrained due to the discrete nature of the leg motions. If the ground contact points are rigidly held, it will be impossible for the contacting legs to change configurations, due to the incompatibilities between each leg's workspace (see Figure 2.18.a). Thus, with rigid ground contact, it is impossible to shift the weight of the body while keeping feet planted. This task is a prerequisite for walking.

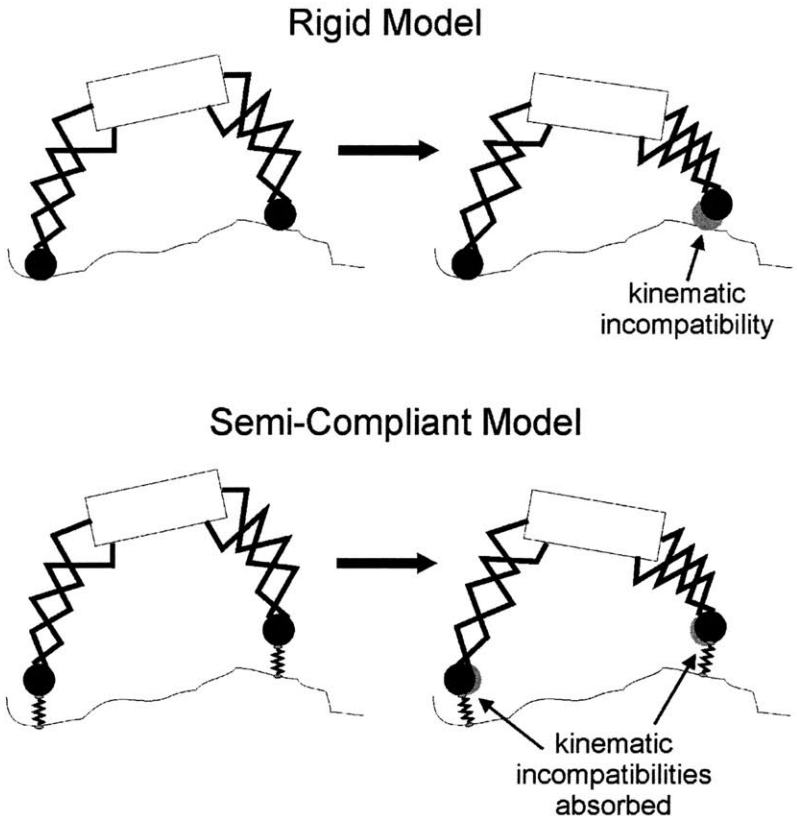


*Figure 2.18. Kinematic models in simulation: (a) rigid; (b) semi-compliant.*

One way to deal with this issue computationally is to assume a small amount of compliance in the ground contact and/or the robotic structure itself. When this is done, small incompatibilities between the workspaces of planted legs are absorbed (see Figure 2.18.b). In simulation, this seemed to be the most realistic and simple method for dealing with this kinematic issue. More discussion on this will follow.

A second major issue arises in the complexity of the inverse kinematics problem. In order to find the binary configuration of the legs that allows the body to move in a prescribed manner, one might be tempted to view the system as a single parallel system with 6 x 21 = 126 DOF. In order to avoid the over constraint problems described above, one must also consider the ground contact points to be continuous revolute joints with continuous displacement compliance. This now makes the problem hybrid in several ways. First, the walker having several legs in contact with the ground forms a macroscopically parallel structure. Each of its legs is a serial arrangement of parallel modules. This makes the walking robot a hybrid parallel-serial-parallel mechanism, a challenge in itself to analyze. Further, the robot is a hybrid system between discrete and continuous motions. With the requirement that trajectories are smooth in both Cartesian and configuration space, this highly hybridized system becomes very computationally difficult to solve quickly.

To get around this issue, the locomotion-planning problem was viewed from a slightly different perspective. Rather than viewing the entire system as a single entity, each leg was viewed as a single trajectory planning problem by itself. For computational purposes, a virtual body position was created, defined to be the desired position of the body for the next small time step. Each leg was then examined *as though it were attached to the virtual body*, and the inverse kinematics were solved using the one-pass combinatorial algorithm to make the leg move to the ground contact point (see Figure 2.19). Because the ground contact points move relative to the moving virtual body, the problem is identical to the trajectory planning problem described in the previous section.

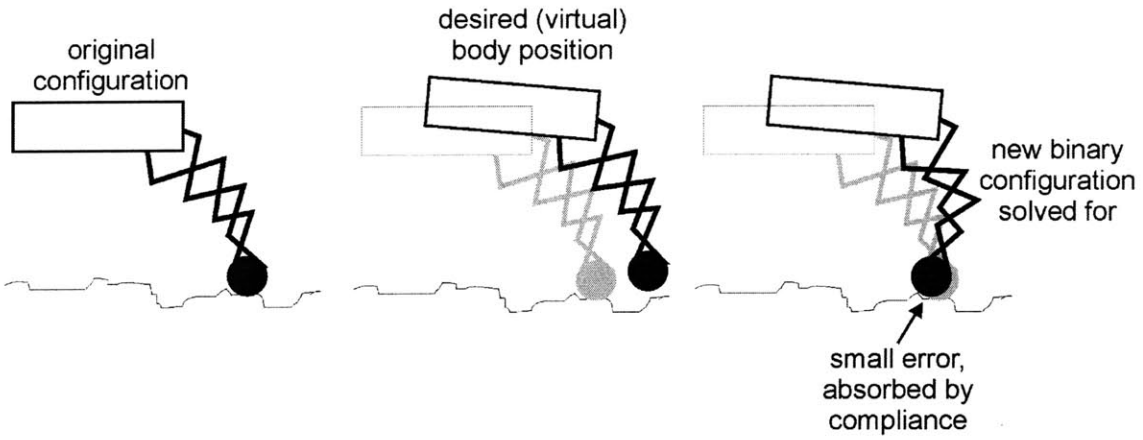


*Figure 2.19. Computational process for planning the trajectories for each leg.*

With the configuration of each leg being solved independently, the problem then switches to a second optimization routine so that graphical representation of the robot is realistic. The problem now becomes one of equilibrating two rigid bodies (the ground and the fixed-configuration robot) connected by compliant elements (see Figure 2.20). This problem can be solved by computing the potential energy stored in the compliance between the two bodies and minimizing it by adjusting the bodies. With complex three-dimensional geometries, deterministic optimization methods can become somewhat involved. Instead, a random trial-and-error method was used which randomly adjusted the robot a small amount relative to the ground until a near-minimum potential energy was found.

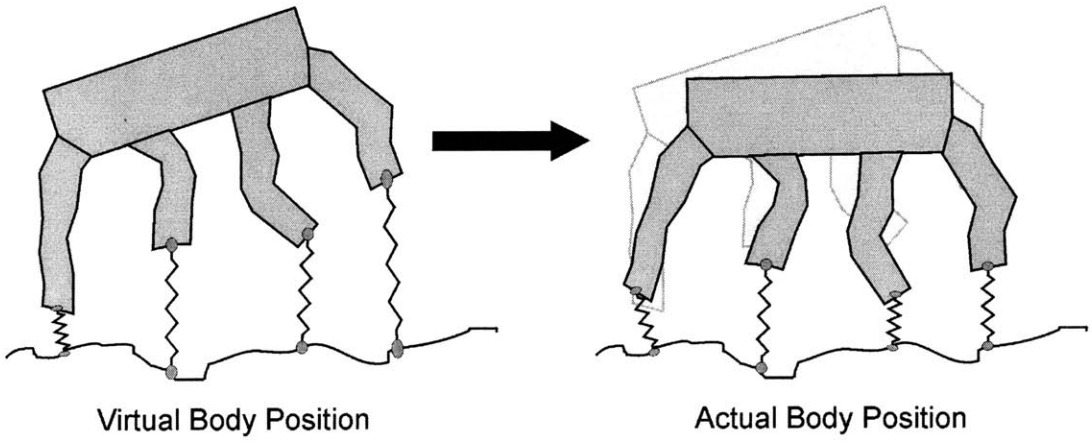


*Figure 2.20. Equilibrating the rigid robot to the ground in simulation: the potential energy in the springs is minimized.*

By the discrete nature of the system, the actual body position does not coincide exactly with the virtual (desired) body position, as can be seen in Figure 2.20. However, the error between the two is small, as it is roughly the size of the errors in the legs themselves, generally a few percent of the characteristic size (see Section 2.4.5). For most real-world scenarios, these errors would be acceptable, since they are of the same size as the measurement noise involved with detecting rough terrain from a mobile platform (the algorithm assumes the topography of local terrain is known).

This planning method was observed to be reliable, fast, and effective. Using a Pentium III 933 MHz processor, the simulated robot could plan and execute a stride nearly once per second. Kinematic incompatibilities between ground and robot were observed to be only a few percent of the leg length, a distance that would be naturally absorbed by compliance in the leg, ground, or deformable shoe at the end of the leg. The robot was able to execute side-stepping and turning motions in rough terrain with no problems observed. The walking robot was seen to maintain static stability at all times while walking on easy terrain (slopes less than 20 degrees) and lost static stability only a few percent of the time while ascending, descending, and traversing slopes of around 45 degrees (although the model assumed infinite static friction). The loss of stability represents problems not with the joint-level planning, but rather with the high-level guidance of the robot, which was merely the automated generation of an alternating tripod gait trajectory. This problem could be avoided simply by using more sophisticated guidance algorithms [Latombe, 1991]. Regardless, the goal of this part of the research was achieved: given a high-level guidance command, the solution of the joint-level command sequences was shown to be feasible at speeds necessary for real-world implementations.

## 2.7 Summary and Conclusions

This chapter presented the analytical challenges for the planning of binary robotic systems, and discussed solution methods for them. The notion of a binary workspace was described, and a method for optimizing robot designs based on uniformity of workspace

density was demonstrated via example. The forward kinematics of binary systems was discussed, and the computational simplicity of this operation relative to continuous systems was shown mathematically. The inverse kinematics and trajectory planning problems were contrasted to those of continuous systems, and several methods for solving these problems were described and compared in the context of space systems using simulations and mathematics. The small positioning errors of binary systems were also quantified in a probabilistic manner. Finally, the trajectory planning methodologies were adapted to the scenario of locomotion planning, which requires the coordinated planning of several binary devices, and is representative of the significant and real challenges to be encountered in future space explorers.

This chapter demonstrates the fundamental feasibility of planning binary robotic systems in representative exploratory tasks. Although exhaustive brute force methods are too cumbersome for implementation in most scenarios, other algorithms have been demonstrated that avoid these problems. The real-time planning of moderately complex systems has been shown to be feasible, even with today's limited computation speeds. One can only expect further growth of planning capabilities in the 10 to 40 year time frame as computation speeds increase.

CHAPTER

# 3

# FUNDAMENTAL HARDWARE DEMONSTRATIONS

## 3.1 Introduction

With the feasibility of many fundamental planning and control issues shown in simulation, it is important to develop experimental demonstrations to explore the physical issues involved with the compliance-based binary robotic concept. In this phase of the study, several key principles of the concept were studied using basic physical implementations. The goal was to provide a starting point for deeper experimental exploration in the future. While none of the following implementations were intended to demonstrate the complete self-transforming explorer (STX) vision, each explored individual aspects of the concept in a building-block fashion.

In physical implementation, a compliant binary robotic device consists of flexible elements to allow motions, binary actuators to provide the motions, and bi-stable members to hold the binary state and reject disturbances. Depending on the type of actuator, some mechanical amplification of the actuator motions might also be required. The following sections discuss preliminary studies of these fundamental building blocks.

## 3.2 Bistable Joints

Many existing binary actuation technologies require power to maintain a fixed state, even though there is no useful output of work. Uninsulated shape memory alloys, for example, require the constant generation of heat in order to maintain one of their

states. Capacitance-based polymer actuators such as conducting polymers and electrostrictive polymers lose charge (and thus their binary state) slowly through small trickle currents and therefore also require a small amount of power to maintain at least one of their states. This leads to poor energy efficiency in slow-moving binary systems, whose actuators spend most of their time in a fixed state. A system that spends a great deal of energy maintaining its rigidity is not acceptable for most applications in space, where power is a very limited resource.

In physical implementation, it is often advantageous to make use of bistable joints. A bistable joint is a mechanism that passively latches a binary robot into its discrete state without the use of actuators or power. This can be done in a variety of ways, such as through the use of detents or with an over-throw design. Bistable mechanisms will allow a robotic device to turn off power to those actuators that are simply maintaining a desired state. In addition, well-designed bistable mechanisms give a robotic device high repeatability, and the robot will maintain the desired binary configuration in the presence of disturbances and environmental variations.

To begin to examine the concept of bistable joints, several simple designs were examined in physical implementation. The goal was to get a qualitative understanding of the integration and potential simplicity of the design of these devices. Designs were conceived, drawn using AutoCAD, and rapidly manufactured from polymer materials using a laser cutter. This allowed a simple trial-and-error approach to the qualitative study.

Many mechanisms exhibit bistability. Light switches, shampoo bottle caps, bicycle kickstands, tape measures, and retractable pens are all household examples. After a general search of basic bistable mechanisms, at least two seemed appropriate for incorporation into a rotary joint-based binary device (see Figure 3.1). One was a spring-loaded over-throw hinge, the basic mechanism used in light switches, shampoo bottle caps, etc. The second was a detent-based latch, with small interlocking knobs that prevent rotation of the joint. The knob is pressed into the detent using the elasticity in the fingers that surround the joint. Of these two mechanisms, the second was more straightforward to realize in a low number of parts, and so was studied in further depth.

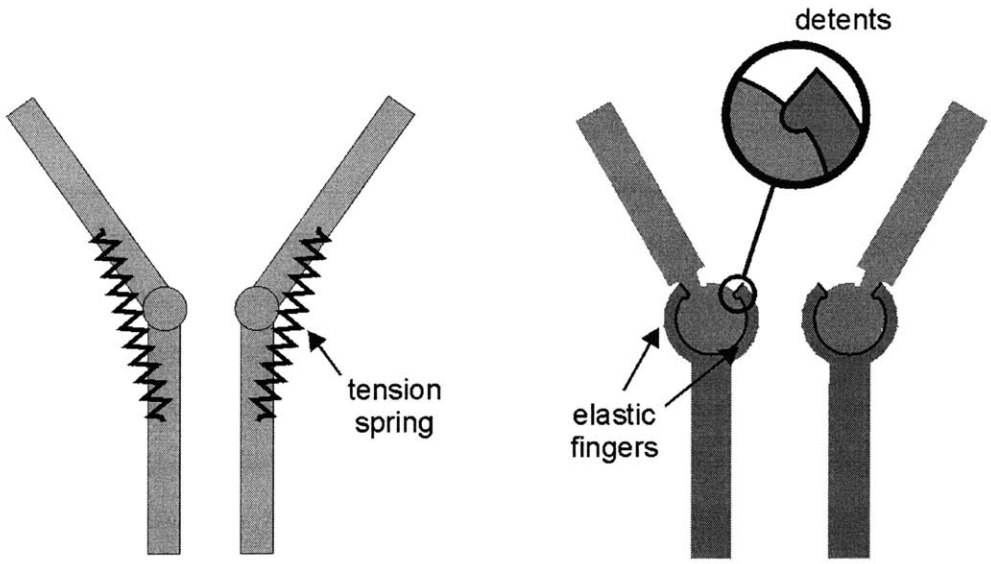


*Figure 3.1. Bistable rotary joint designs: (a) spring-loaded over-throw mechanism; (b) detent-based latching mechanism.*

Using a laser cutter, candidate designs were cut from plastic sheets having thicknesses between one and ten millimeters. Nylon, ultra-high-molecular-weight polyethylene, acrylic, polycarbonate, ABS (acrylonitrile butadiene styrene), and PETG (polyethylene terepthalate glycol) were all tested for compatibility with the laser cutter and quality of the part produced. In the end, PETG was found to have the best combination of cutting properties and mechanical properties for the given application. It cut very well on the laser cutter and had sufficient mechanical properties for most applications. Polycarbonate had superior fatigue strength and creep properties and fair cutting properties, and was also used in some prototypes after residue from the cutting operation was removed by hand. All other plastics studied either had inferior mechanical properties (acrylic, nylon, ABS), or inferior laser-cutting properties (polyethylene).

Several bistable rotary joints were designed and fabricated based on a sandwich of the detent-based mechanism and elastic flexure elements (see Figure 3.2). In these designs, the flexure elements added a great deal of out-of-plane rigidity to the joint and largely reduced slop in the bistable mechanism. Some prototypes also incorporated binary SMA actuators in a muscle-like fashion, although only poor actuation performance was realized in these early designs. The design was also reduced in size to the practical

limits of manufacture, with the entire mechanism being roughly 50 millimeters in length (see Figure 3.3). The results of this preliminary study show the simplicity of many bistable designs, and offer at least one interesting candidate that can be manufactured from a very small number of parts in a near-monolithic fashion.

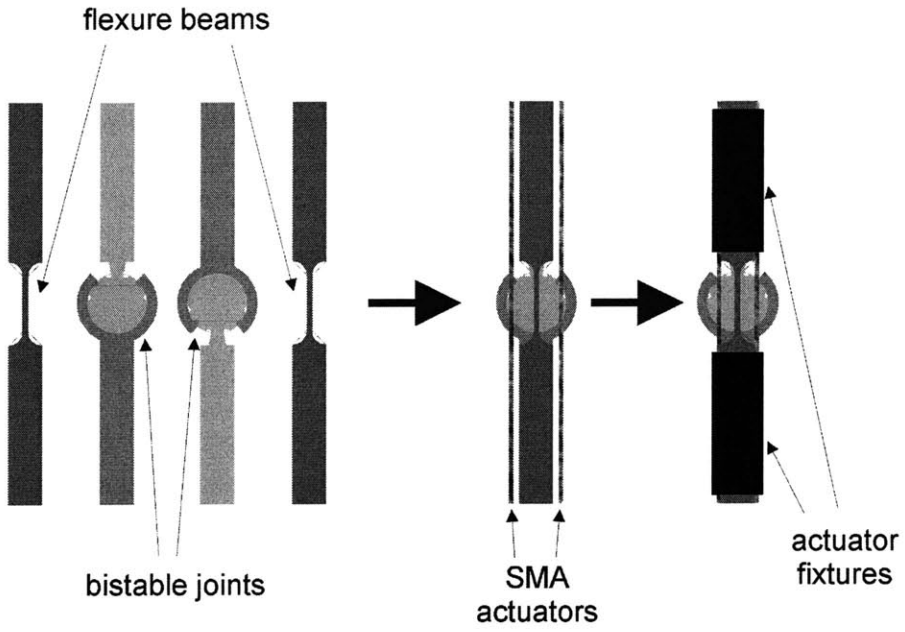


*Figure 3.2. Bistable joint: a sandwich of bistable members and elastic hinges.*

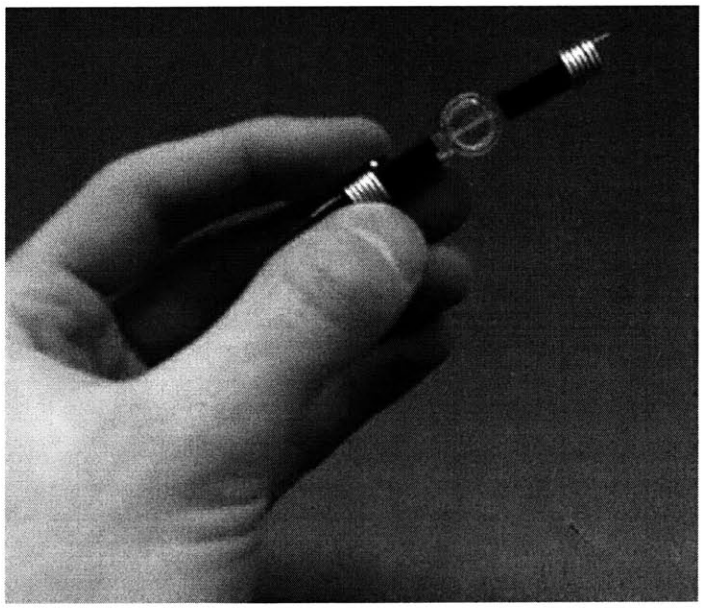

*Figure 3.3. Bistable joint prototype.*

## 3.3 Pantograph Mechanism

A second study was developed to study the concept of motion amplification in a monolithic design. Most muscle-type actuators are currently able to achieve only small motions. Shape memory alloy wires, for example, give usable elongations of only 3-5 percent of their length. Conducting polymer actuators presently can attain only 1-2 percent deformation, although much higher strains are predicted with time. For most robotic devices, larger motions than these are required, and thus some form of motion amplification is needed.

Many amplification methods were examined for the application of binary robotics (see Figure 3.4). Each was studied for its ease of realization in a monolithic design and its amplification properties. A simple lever can provide amplification of at least an order of magnitude, however it does not lend itself well to a compact design. More complex linkages were studied, but many of these were found to have very nonuniform amplification over their range of motion, which is undesirable in this application. One linkage that does provide relatively uniform motion amplification in a compact size is the pantograph mechanism [Nielson, et al, 1998]. The basic mechanism can be configured in a number of ways, including in a compound arrangement, where mechanisms are cascaded to achieve higher amplification.

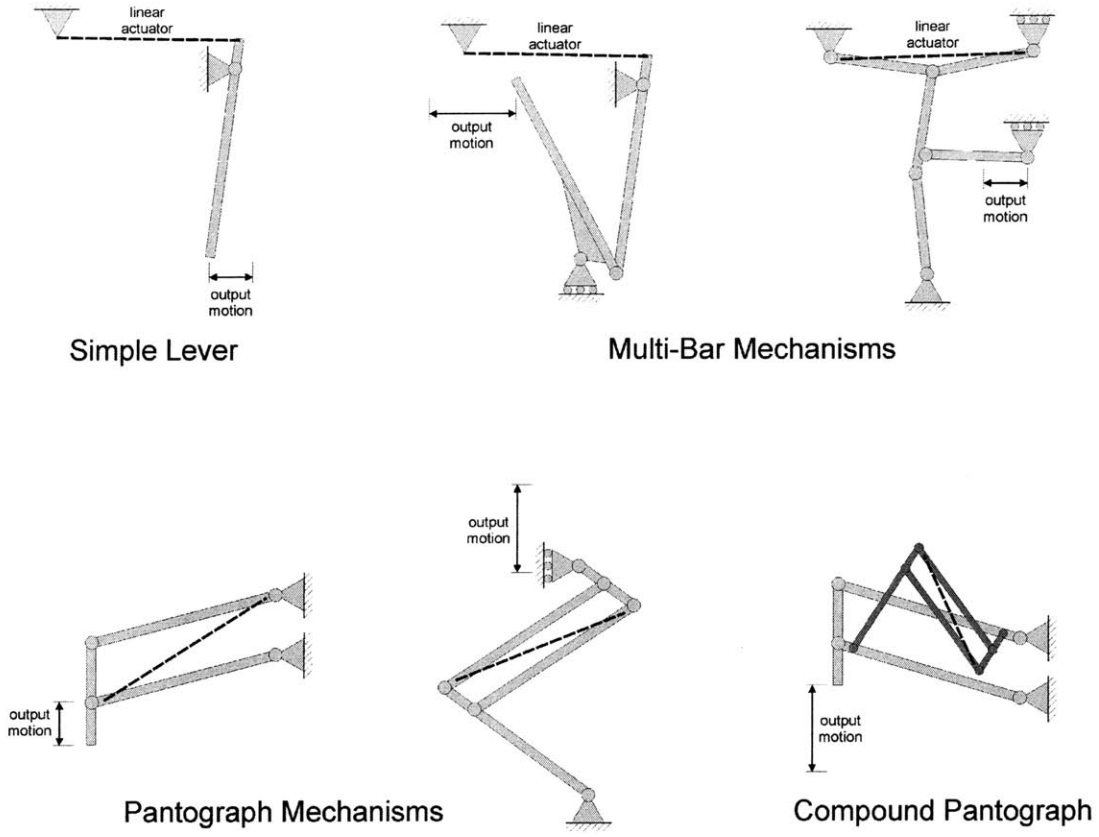


*Figure 3.4. Various amplification mechanisms.*

Because of its good amplification behavior, compact size, and simple structure, the pantograph was developed into a monolithic prototype (see Figures 3.5 and 3.6). In this design, the revolute joints were replaced by beam flexures and the entire structure (four bars and four flexures) was laser-cut as a single piece from 1/8-inch PETG sheet. An SMA wire actuator was affixed to the mechanism with nuts, washers, and bolts, which made adjustment and fine-tuning easier.

When current is supplied to the SMA, it heats up and contracts by 4.5 mm, raising the output link a distance of 29 mm. The elasticity of the flexures provides a bias spring that serves to return the mechanism to its original state when the SMA has been powered down. This mechanism achieves an amplification ratio of about 6.5, somewhat less than the amplification of 8.0 predicted from calculations. The discrepancy between these two values is mostly due to unmodeled compliance in the structure and imperfections in the hand-fabrication process.

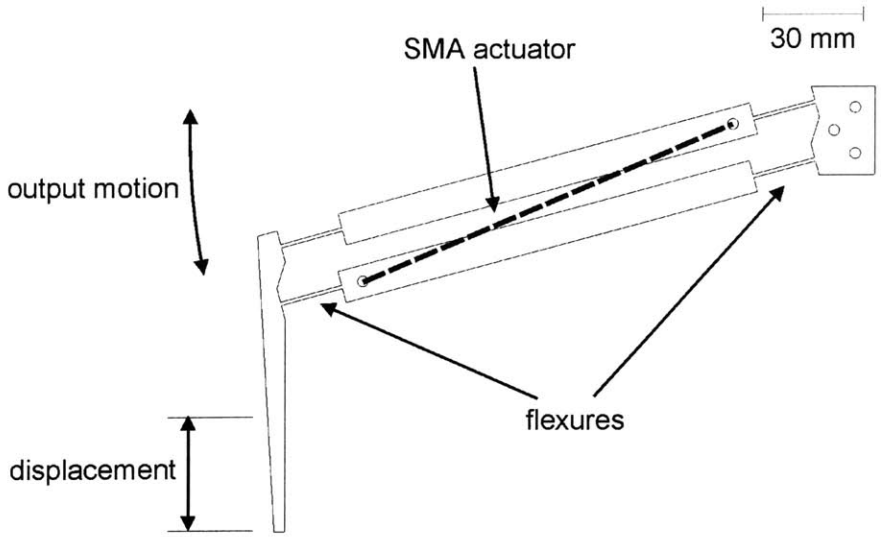


*Figure 3.5. Pantograph mechanism schematic.*

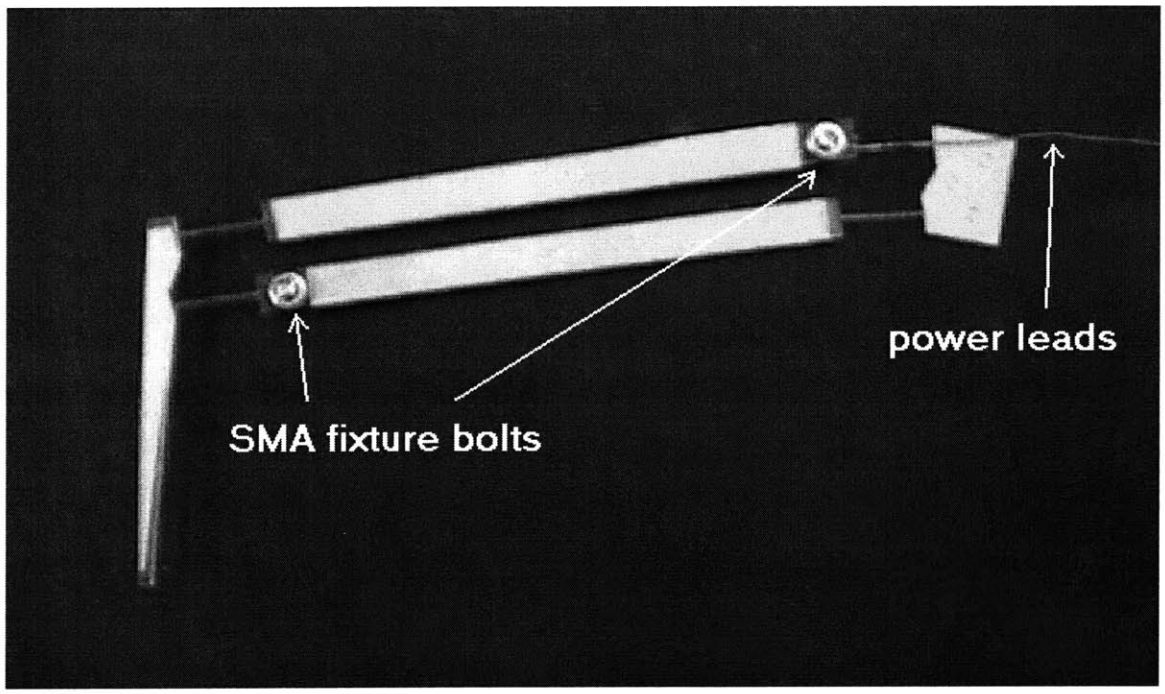


*Figure 3.6. Pantograph mechanism prototype.*

The pantograph clearly illustrates the simplicity of potential amplification mechanisms and the compliance-based binary robotic concept. Without the electronics, the device has a mass of only 10.5 grams. It is compact, having a length only slightly larger than that of the actuator itself. This type of device is easily fabricated from a

single piece of material. In this case, the prototype was laser-cut, however, many other methods could be used, such as water jet cutting, stamping, machining, or injection molding.

## 3.4 Binary Robotic Articulated Intelligent Device, Generation 2 (BRAID 2)

An area of ongoing work is the development of the Binary Robotic Articulated Intelligent Device (BRAID). This system is a serial arrangement of 3-DOF binary parallel stages (see Section 2.3, Figure 2.6). A first prototype was built in 1999 with interesting results [Oropeza, 1999; Lichter, et al, 2000; Sujan, et al, 2001]. This system was actuated using shape memory alloy wires, which opened the polyethylene linkages in a muscle-like manner (see Figure 3.7). The elasticity in the flexures was used to return the linkages to their closed state when power was removed from the actuators. This resulted in a very lightweight design capable of being stowed in a small volume (see Figure 3.8).

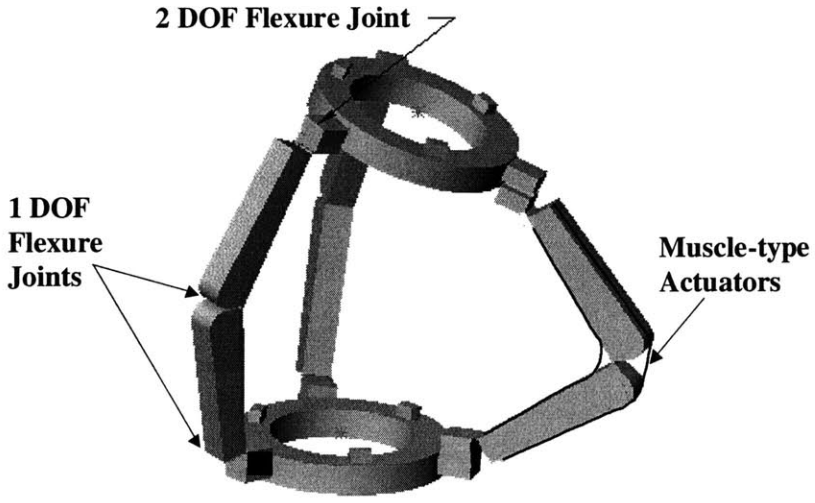


*Figure 3.7. Single stage of first BRAID prototype. [Sujan, et al, 2001]*

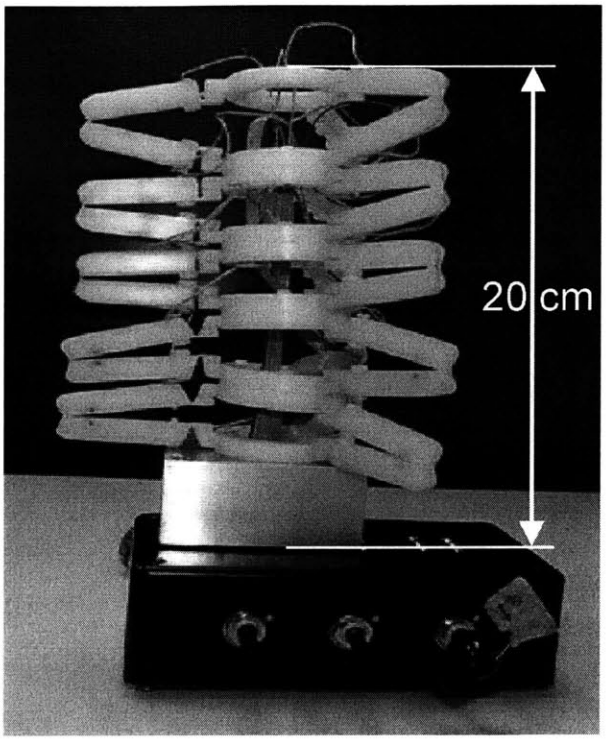


*Figure 3.8. First generation BRAID prototype. [Oropeza, 1999]*

The first prototype was able to achieve moderate motions and repeatability. This was very good for a first prototype, but future prototypes will require some improvement. Shape memory alloy actuators are not ideally suited to this application, as they contract only a few percent of their length and require amplification. Because of their small motions, the performance of the design is very sensitive to small fabrication imperfections, and the mounting products available at the time were very limited. This made the hand fabrication of the prototype difficult. Some difficulty was also encountered in the manufacture of the flexures. The dimensions of the hinges require a fair amount of precision that is difficult to achieve reliably with the manufacturing techniques available. Ultimately, water jet cutting proved to be the most reliable, but some of the linkages still exhibited a reduced fatigue life.

This research is now focusing on the development of the BRAID design as a way to further demonstrate the STX vision. Materials and fabrication techniques are being explored deeper to alleviate the fatigue life problems encountered in the first prototype. The design of the flexures themselves is also being explored further in order to improve

the fatigue life. Bistable mechanisms will be included in the next generation prototype to improve repeatability. Alternative actuator technologies are being surveyed, with the prime candidate currently electrostrictive polymers. To date, these actuators seem to have the best motion characteristics, light weight, and simplicity and appear most promising for this application. However, some development of this technology is still required for integration with the BRAID, so for a short time conventional voice-coil actuators will be used as a surrogate. This will allow the development of the flexures, manufacturing, and electronics to occur in parallel with the actuator technologies.

A tentative design is now under construction and is composed primarily of Delrin (acetal resin, polyoxymethylene), which has excellent fatigue, creep, friction, wear, and machining properties. Analysis of the BRAID structure shows that high angular deflections (±50 degrees) are required at the 3-DOF joint (see Appendix C, Figure C.3) in order to provide large motions of the device and good system performance. For this reason, spherical ball and socket joints are being considered to replace the flexural elements for this joint. A tentative design for a second generation BRAID is shown in Figure 3.9.

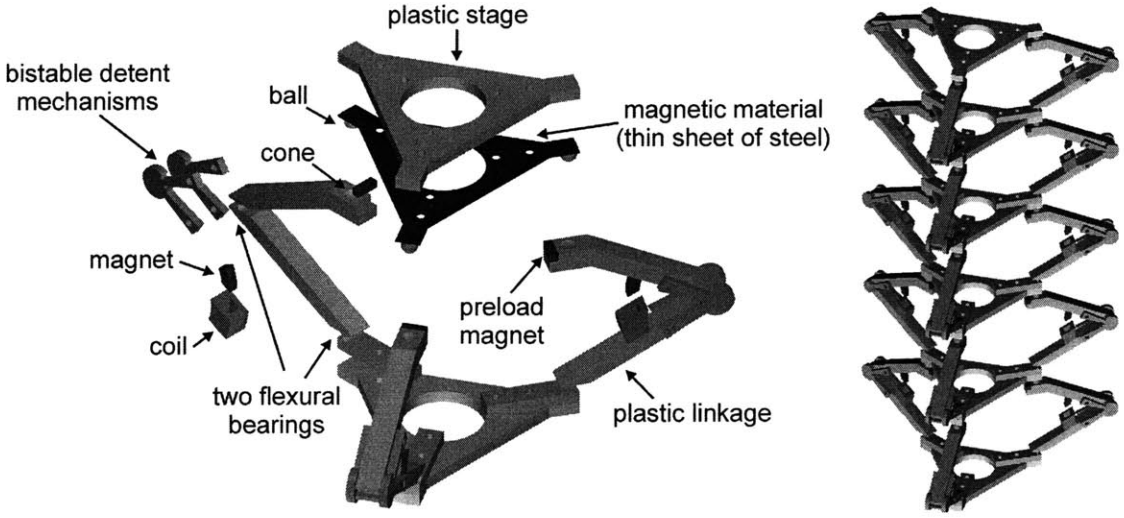


*Figure 3.9. Tentative design of a second generation BRAID. (Courtesy M. Hafez.)*

An unactuated single stage was fabricated based on this design using a waterjet cutter and hand assembly with epoxy (see Figure 3.10). This prototype was used to

demonstrate manufacturing potential and verify elements of the structural design and material properties. As can be seen in this prototype, cross flexural hinges were used to greatly improve fatigue life, range of motion, and out-of-plane stiffness over the notch flexures used in the first generation prototype. This type of flexure will likely see use in the final operational prototype.


*Figure 3.10. Test prototype of an unactuated single BRAID stage.*

## 3.5 Summary and Conclusions

This chapter has described several experimental studies carried out to understand practical limitations of the STX concept. Bistable mechanisms were explored via the fabrication of many prototypes, with several being of a size expected in a real STX design. These prototypes demonstrated their fundamental value in binary systems, as they improved repeatability, stiffness, and disturbance rejection significantly in rotary joints. A compliant pantograph was also built to demonstrate the simplicity of compliant amplification mechanisms. Work is now progressing in this area with the development of the BRAID system. The goal is to eventually embody all of the elements of embedded binary actuation and compliant bistable mechanisms in a single useful device.

It appears that at this time the major obstacle to development of the STX vision lies in the actuator technologies. Modern actuator technologies are presently hindered by high power requirements, low displacements, very slow response, low stiffness, and/or difficulty of fabrication or implementation. However, many of these problems will be resolved with time and research, and no doubt will enable a wide range of new engineering designs. Further developments in the areas of manufacturing and structural materials will also prove valuable to the design and production of more integrated, high performance STX-like systems.

CHAPTER

# 4

# CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

## 4.1 Contributions of this Work

This work has developed the concept of exploratory space robotics consisting of embedded binary actuators and compliant bistable mechanisms.

Chapter 2 presents methods for solving the kinematics and planning issues of complex binary systems with large numbers of actuators in real time. Existing methods for solving the inverse kinematics were applied to specific kinematic structures and developed in the context of the self-transforming explorer (STX) vision. Example methods for optimizing robotic designs based on given functional requirements were also demonstrated. Methods for dealing with the complexity of hybrid parallel-serial-parallel and discrete-continuous systems (i.e. a multi-legged binary walking robot) were created and demonstrated through simulations. Most importantly, the real-time joint-level planning of representative STX tasks was demonstrated in simulation.

Chapter 3 presents preliminary experimental work that demonstrates elements of the STX vision and will guide future studies. A qualitative exploration of bistable mechanisms was performed that included the fabrication of prototype devices. Several devices of a size appropriate to STX applications were realized using moderate precision manufacturing methods. An integrated, one-piece pantograph mechanism was also designed and fabricated to demonstrate the simplicity of compliant amplifier mechanisms. Work is continuing in this area with the development of the BRAID design,

a highly redundant articulated device, with the goal of eventually incorporating all aspects of the STX concept in a functional prototype.

## 4.2 Suggestions for Future Work

Over the course of this research, a diversity of concepts has been explored. Much of the emphasis has been placed on breadth rather than depth, especially in the case of the experimental studies. For this reason, there are a number of topics that would prove worthwhile to study in the future.

The optimization of robot topology and geometry based on workspace requirements warrants deeper study. The study done here was primarily used to illustrate the notion of such an optimization and showed the fundamental feasibility of the idea. However, it was limited to a serial planar robot, with the understanding that the underlying optimization problem is essentially the same for any robotic topology. The details of the optimization of a more complex device are an interesting and unexplored area.

Simulation studies could be taken further to demonstrate a higher level of autonomous planning. The organization of several binary modules to complete a random set of tasks has not been studied yet in this context. Communication methods and architecture could be studied in simulation, to analyze efficient transfer of information and command hierarchies. Concepts such as de-centralized and multi-agent control would be useful to study to further demonstrate feasibility of the STX vision.

Work should continue in the development of experimental prototypes to study the STX concept. Further integration of compliant bistable members and embedded binary actuation is needed to more fully explore the vision. Current work is focused on the redesign and development of the BRAID system.

As materials, computing, and manufacturing research progress, they will open up new avenues for the robotics paradigm described here. Actuator technologies are currently the biggest obstacle to the development of the STX vision. The introduction of new engineering materials and fabrication methods will also enable new types of designs.

Increased computation speeds and processor size reductions will give additional power to autonomous planning and control. Given time and research, each of the elements of the STX concept will reach a sufficient state of development to allow the full vision to be realized.

# REFERENCES


Agrawal S K, et al. "Control of Shapes of Elastic Plates Using Embedded Piezoelectric Actuators." *Proceedings of the SPIE Conference of Smart Structures and Materials: Smart Structures and Intelligent Systems*, Orlando, FL, Feb 1994.

Ananthasuresh G, Kota S. "Designing Compliant Mechanisms." *Mechanical Engineering*, Vol. 117, No. 11, 93-96, 1995.

Andrews K. "Elastic Elements with Embedded Actuation and Sensing for Use in Self-Transforming Robotic Planetary Explorers." M.S. Thesis, MIT, Cambridge, MA, 2000.

Baughman R. "Conducting Polymer Artificial Muscles." *Synthetic Metals*, No. 78, 339-353, 1996.

Baz A, et al. "Shape Control of Nitinol-Reinforced Composite Beams." *Proceedings of the SPIE Conference on Smart Structures and Materials: Smart Structures and Intelligent Systems*, Orlando, FL, Feb 1994.

Bickler D. "The New Family of JPL Planetary Surface Vehicles." *Proceedings of Missions, Technologies, and Design of Planetary Mobile Vehicles*, Toulouse, France, 1992.

Canfield S, Edinger, B, Frecker M, Koopmann G. "Design of Piezoelectric Inchworm Actuator and Compliant End Effector for Minimally Invasive Surgery." *Proceedings SPIE 6th International Symposium on Smart Materials and Structures*, Newport Beach, California, Paper 3668-78, March 1999.

Canny J, Goldberg K. "A RISC Paradigm for Industrial Robotics." *Technical Report ESCR 93-4/RAMP 93-2*, Engineering Systems Research Center, University of California at Berkeley, 1993.

Cappelleri D, Frecker M. "Optimal Design of Smart Tools for Minimally Invasive Surgery." *Proceedings Optimization in Industry*, Banff, Canada, June 1999.

Cham J G, Pruitt B L, Cutkosky M R, Binnard M, Weiss L E, Neplotnik, G. "Layered Manufacturing with Embedded Components: Process Planning Issues." *ASME Proceedings, DETC '99*, Las Vegas, Nevada, September 12-15, 1999.

Chirikjian G S, Ebert-Uphoff I. "Discretely Actuated Manipulator Workspace Generation Using Numerical Convolution on the Euclidean Group." *Proceedings of the IEEE International Conference on Robotics and Automation*, 742-749, 1998.

Chirikjian G S. "A Binary Paradigm for Robotic Manipulators." *Proceedings of the IEEE International Conference on Robotics and Automation*, 3063-3069, 1994.

Chirikjian G. "Inverse Kinematics of Binary Manipulators Using a Continuum Model." *Journal of Intelligent and Robotic Systems*, 19:5-22, 1997.

Chirikjian G. "Kinematic Synthesis of Mechanisms and Robotic Manipulators with Binary Actuators." *ASME Journal of Mechanical Design*, 117(Sept):573-580, 1995.

Craig J J. "Introduction to Robotics: Mechanics and Control." Second ed, Addison-Wesley, Reading, MA, 1989.

Della Santa A, et al. "Steerable Microcatheters Actuated by Embedded Conducting Polymer Structures." *Journal of Intelligent Material Systems and Structures*, Vol 7 May 1996.

Dubowsky S. *NASA Institute for Advanced Concepts Phase I Study of Self-Transforming Robotic Planetary Explorers: Final Report*, 1999. http://robots.mit.edu/research/NIAC/PhaseI.pdf

Ebert-Uphoff I, Chirikjian G S. "Inverse Kinematics of Discretely Actuated Hyper-Redundant Manipulators Using Workspace Densities." *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, 139-145, 1996.

Erdmann M A, Mason M T. "Exploration of Sensor-less Manipulation." *IEEE Journal of Robotics and Automation*, Vol. 4, 369-379, 1988.

Farritor S, Dubowsky S, Rutman N, Cole J. "A System Level Modular Design Approach to Field Robotics." *1996 IEEE International Conference on Robotics and Automation*, Minneapolis, 2890-5, 1996.

Farritor S, Dubowsky S. "A Self-Planning Methodology for Planetary Robotic Explorers." *1997 International Conference on Advanced Robotics*, Monterey, CA, 1997.

Farritor S. "On Modular Design and Planning for Field Robotic Systems." PhD Thesis, MIT, Cambridge, MA, 1998.

Frecker M, Kikuchi N, Kota S. "Optimal Synthesis of Compliant Mechanisms to Meet Structural and Kinematic Requirements – Preliminary Results." *Proceedings of the 1996 ASME Design Engineering Technical Conferences*, 96-DETC/DAC-1497, 1996.

Frecker M, Kikuchi N, Kota S. "Topology Optimization of Compliant Mechanisms With Multiple Outputs." *Structural Optimization*, 17: 269-278, 1999.

Gilbertson R G. *Muscle Wires Project Book*. 3rd Edition, Mondotronics Inc., San Anselmo, CA, 1994.

Goldberg D. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

Goldberg K. "Orienting Polygonal Parts Without Sensors." *Algorithmica*, Special Robotics Issue, 1992.

Hetrick J, Kota S. "Size and Shape Optimization of Compliant Mechanisms: An Efficiency Formulation." *Proceedings of 1998 ASME Design Engineering Technical Conference*, Atlanta, GA, DETC98/MECH-5943, 1998.

Hirose S. "Super Mechano-System: New Perspective for Versatile Robotic System." *Proceedings of the Seventh International Symposium on Experimental Robotics, ISER 00*, Dec 2000.

Howell L, Midha A. "The Development of Force-Deflection Relationships for Compliant Mechanisms." *Machine Elements and Machine Dynamics*, 1994.

Huang W, Pellegrino S. "Shape Memory Alloy Actuators for Deployable Structures." Cambridge University Engineering Department CUED/D-STRUCT/TR163, 1996.

Huntsberger T L, Rodriguez G, Schenker P S. "Robotics: Challenges for Robotic and Human Mars Exploration." *Proceedings of ROBOTICS2000*, Albuquerque, NM, Mar 2000.

Iqbal K, Pellegrino S. "Bi-stable Composite Shells." *Proceedings of the 41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, Atlanta, GA, April 2000.

Jensen B D. "Identification of Macro- and Micro-Compliant Mechanism Configurations Resulting in Bistable Behavior." M.S. Thesis, Brigham Young University, Provo, Utah, 1998.

Jensen B D, Howell L L, Gunyan DB, Salmon L G. "The Design and Analysis of Compliant MEMS Using the Pseudo-Rigid-Body Model." *Microelectomechanical Systems (MEMS) 1997*, presented at the 1997 ASME International Mechanical Engineering Congress and Exposition, November 16-21, 1997, Dallas, Texas, DSC-Vol. 62, pp. 119-126, 1997.

Jolly M R, et al. "The Magnetoviscoelastic Response of Elastomer Composites Consisting of Ferrous Particles Embedded In a Polymer Matrix." *Journal of Intelligent Material Systems and Structures*, Vol 7, November 1996.

Kawauchi Y, Inaba M, Fukuda T. "Self-Organising Intelligence for Cellular Robotic System (CEBOT) with Genetic Knowledge Production Algorithm." *Proceedings of IEEE International Conference on Robotics and Automation*, 813-8, 1992.

Koliskor A. "The l-coordinate Approach to the Industrial Robots Design." in Information Control Problems in Manufacturing Technology. *Proceedings of the 5th IFAC/IFIP/IMACS/IFORS Conference*, USSR, 1986.

Kornbluh R, Pelrine R, Joseph J, et al. "High-Field Electrostriction of Elastomeric Polymer Dielectrics for Actuation." *SPIE Conference on Electrostrictive Polymer Actuators and Devices*, Vol. 3669, 149-161, 1999.

Kotay K, Rus D, Vona M, McGray C. "The Self-Reconfiguring Robotic Molecule." *Proceedings of the IEEE Conference on Robotics and Automation*, Leuven, Belgium, 424-31, 1998.

Kotay K, Rus D. "Task Reconfigurable Robots: Navigators and Manipulators." *Proceedings of Intelligent Robots and Systems*, 1997.

Kyatkin, A B, Chirikjian, G S. "Synthesis of Binary Manipulators Using the Fourier Transform on the Euclidean Group." *Journal of Mechanical Design*, 121 (March):9-14, 1999.

Latombe J. "Robot Motion Planning." Kluwar Academic Publishers, Boston, MA 1991.

Lee C, Sun C. "The Nonlinear Frequency and Large Amplitude of Sandwich Composites with Embedded Shape Memory Alloys." *Journal of Reinforced Plastics and Composites*, Vol. 14, 1160-74, November 1995.

Lees D S, Chirikjian G S. "A Combinatorial Approach to Trajectory Planning for Binary Manipulators." *Proceedings of the IEEE International Conference on Robotics and Automation*, 2749-2754, 1996.

Lees D S, Chirikjian G S. "An Efficient Method for Computing the Forward Kinematics of Binary Manipulators." *Proceedings of the IEEE International Conference on Robotics and Automation*, 1012-1017, 1996.

Lichter M D, Sujan V A, Dubowsky S. "Experimental Demonstrations of a New Design Paradigm in Space Robotics." *Proceedings of the Seventh International Symposium on Experimental Robotics, ISER 00*, Dec 2000.

Madden J D, Cush R A, Kanigan T S, et al. "Fast-contracting Polypyrrole Actuators." *Synthetic Metals*, 113: 185-193, 2000.

Madden J, Lafontaine S, Hunter I. "Fabrication by Electrodeposition: Building 3D Structures and Polymer Actuators." *Proceedings of the Sixth Symposium on Micro Machine and Human Science*, 1995.

Midha A, et al. "A Methodology for Compliant Mechanism Design: Part I - Introduction and Large Deflection Analysis." *ASME Advances In Design Automation*, Vol. 2, 1992.

Murata S, Kurokawa H. Kokaji S. "Self Assembling Machine." *Proceedings IEEE International Conference on Robotics and Automation*, Vol. 1, 441-8, 1994.

Murphy M, et al. "The Topological Analysis of Compliant Mechanisms." *Machine Elements and Machine Dynamics*, 1994.

*NASA Strategic Plan.* 1998 NASA Policy Directive (NPD)-1000.1, http://www.hq.nasa.gov/office/nsp.

Neider J, Davis T, Woo M. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Release 1.* Addison-Wesley, Reading, MA, 1993.

*NASA Institute for Advanced Concepts*, 2001. http://www.niac.usra.edu/

Nielson A J, Howell, L L. "Compliant Pantographs via the Pseudo-Rigid-Body Model." *Proceedings of the 1998 ASME Design Engineering Technical Conferences*, DETC98/MECH-5930, 1998.

*NIST/Sematech Engineering Statistics Handbook*, 2001. http://www.itl.nist.gov/div898/handbook/eda/section3/eda366b.htm.

Opdahl P, Jensen B, Howell L. "An Investigation into Compliant Bistable Mechanisms." *ASME Biennial Mechanisms Conference*, Atlanta, GA, 1998.

Oropeza G. "The Design of Lightweight Deployable Structures for Space Applications." Thesis for the Bachelor of Science in Mechanical Engineering, Massachusetts Institute of Technology, May 1999.

Pamecha A, Ebert-Uphoff I, Chirikjian G. "Useful Metrics for Modular Robot Motion Planning." *IEEE Transactions on Robotics and Automation*, Vol. 13, No. 4, 531-45, 1997.

Pelrine R E, Kornbluh R D, Joseph J P. "Electrostiction of Polymer Dielectrics with Compliant Electrodes as a Means of Actuation." *Sensors and Actuators*, A 64, 77-85, 1998.

Pelrine R, Kornbluh R, Pei Q, et al. "High-speed Electrically Actuated Elastomers with Over 100% Strain." *Science*, Vol. 287, No. 5454, 836-839, 2000.

Pieper D L. "The Kinematics of Manipulators under Computer Control." PhD Thesis, Stanford University, 1968.

Roth B, Rastegar J, Scheinman V. "On the Design of Computer Controlled Manipulators." *First CISM-IFTMM Symposium on Theory and Practice of Robots and Manipulators*, 93-113, 1973.

Saggere L, Kota S. "Synthesis of Distributed Compliant Mechanisms for Adaptive Structures Applications: An Elasto-Kinematic Approach." *Proceedings of the 1997 ASME Design Engineering technical Conferences*, DETC97/DAC-3861.

Salamon B, Midha A. "An Introduction to Mechanical Advantage in Compliant Mechanisms." *18th ASME Design Automation Conference: Advances in Design Automation*, DE-Vol. 44-2, , 1992.

Schenker P. "Lightweight Rovers for Mars Science Exploration and Sample Return." *Intelligent Robots and Computer Vision XVI*, SPIE Proc. 3208, Pittsburgh, PA, October, 1997.

Sen D, Mruthyunjaya T S. "A Discrete State Perspective of Manipulator Workspaces." *Mech. Mach. Theory*, Vol. 29, No.4, 591-605, 1994.

Sujan, V A, Lichter M D, Dubowsky S. "Design and Analysis of Lightweight Hyper-redundant Binary Elements for Planetary Exploration Robots." *Journal of Mechanism and Machine Theory: The Scientific Journal of the International Federation for the Theory of Machines and Mechanisms*, Pergamon Press, 2001.

Sujan V A, Lichter M D, Dubowsky S. "Lightweight Hyper-redundant Binary Elements for Planetary Exploration Robots." *Proceedings of the 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '01)*, Como, Italy, July 2001.

Suthakorn J, Chirikjian G S. "A New Spatial Binary-Actuated Manipulator Design." *Proceedings of the Seventh International Symposium on Experimental Robotics, ISER 00*, 157-162, 2000.

Tesar D, Butler M. "A Generalized Modular Architecture for Robotic Structures." *Manufacturing Review*, Vol. 2, No. 2, 1989.

Wang G, Shahinpoor M. "A New Design for a Bending Muscle with an Embedded SMA Wire Actuator." *SPIE* Vol. 2715, 51-61, 1996.

Waram T. *Actuator Design Using Shape Memory Alloys.* 2nd Edition, T. C. Waram Publisher, Hamilton Ontario, Canada, 1993.

Yim M. "Locomotion with a Unit Modular Reconfigurable Robot." PhD Thesis, Stanford University, Palo Alto, 1995.

Youyi C, Tu H. "Shape Memory Materials '94." *Proceedings of the International Symposium of Shape Memory Materials*, Beijing, China, 1994.

APPENDIX

# A

# COMBINATORIAL SEARCH ALGORITHM SAMPLE CODE

```
//computes inverse kinematics of a binary mechanism using a combinatorial
//search algorithm

int i, j, k,              //counters representing configuration bits
    number_bits,          //total number of binary DOF
    best_bit_1,           //best bits to flip in configuration for better fitness
    best_bit_2,
    best_bit_3,
    configuration[MAX_BITS];     //array of 0's and 1's representing
                                 //binary configuration of robot

double  best_fitness,            //running tab of best fitness found
        current_fitness;         //fitness of current candidate configuration


do {
    //measure the fitness of the original configuration (baseline)
    best_fitness = compute_fitness(configuration);

    //mark the current configuration as the best for now
    best_bit_1=-1;
    best_bit_2=-1;
    best_bit_3=-1;

    for (i=0; i<number_bits; i++) {
        for (j=i+1; j<number_bits+1; j++) {
            for (k=j+1; k<number_bits+2; k++) {

                //flip three bits in the mechanism
                configuration[i]=!configuration[i];
                configuration[j]=!configuration[j];
                configuration[k]=!configuration[k];

                //compute the fitness of the new configuration
                current_fitness=compute_fitness(configuration);

                //see if this new configuration is better
                if (current_fitness > best_fitness) {
                    //update the best fitness value
                    best_fitness=current_fitness;
                    //record the configuration if it is better
                    best_bit_1=i;
                    best_bit_2=j;
```

```
                    best_bit_3=k;
                }

                //flip the bits back to their original state
                configuration[i]=!configuration[i];
                configuration[j]=!configuration[j];
                configuration[k]=!configuration[k];

            }   //(repeat for all 3-bit combinations)
        }
    }

    //if a better configuration was found, record it
    if (best_bit_1 > -1) {
        configuration[best_bit_1]=!configuration[best_bit_1];
        configuration[best_bit_2]=!configuration[best_bit_2];
        configuration[best_bit_3]=!configuration[best_bit_3];
    }
}
while (best_bit_1 > -1);

//search all 3-bit modifications again, starting from this new
//configuration, until a better solution cannot be found
```

APPENDIX

# B

# GENETIC ALGORITHM SAMPLE CODE

```
//computes inverse kinematics of a binary mechanism using a genetic algorithm

#define POPULATION_SIZE 100              //population size
#define NUMBER_GENERATIONS 100           //number of generations
#define NUMBER_SEPARATE_EVOLUTIONS 10    //# of populations separately evolved
#define STAGE_MUTATION_FACTOR 0.1        //probability of mutating a stage

double  fitness[POPULATION_SIZE],
        //an array describing the fitness of each individual in a generation

        fitsum,                  //total sum of the fitnesses of the individuals
        cumul_prob,
        //cumulative sum of the fitnesses of first i individuals in generation

        test,                        //temporary value
        best_fitness,                //fitness of best individual in generation
        overall_best_fitness=0.;     //fitness of best individual overall

int     configuration[POPULATION_SIZE][MAX_BITS],
        //an array of arrays representing the binary configurations of all the
        //individuals in a population for the current generation

        prev_configuration[POPULATION_SIZE][MAX_BITS],
        //representation of the previous generation

        overallbestconfig[MAX_BITS],
        //the configuration of the individual with the best fitness over all
        //generations and all separately evolved populations

        best_index,          //index of the most fit individual in a generation
        temp,                //temporary value
        i,j,k,l;             //counters


//run several separate evolutions with different initial populations
for (l=0; l<NUMBER_SEPARATE_EVOLUTIONS; l++) {

    //population initialization
    for (k=0; k<POPULATION_SIZE; k++) {
        for (i=0; i<number_bits; i++) {
            configuration[k][i]=int(random()+.5); //bit is a random 0 or 1
        }
    }

    //evolution of an initial population over a number of generations
    for (j=0; j<NUMBER_GENERATIONS; j++) {
```

```
    //evaluation
    fitsum=0.;
    best_fitness=0.;
    for (k=0; k<POPULATION_SIZE; k++) {
        //compute fitness of each individual in this generation
        fitness[k]=compute_fitness(configuration[k]);
        if (fitness[k] > best_fitness) {
            best_fitness=fitness[k];
            best_index=k;
        }
        fitsum+=fitness[k];
        //copy current configuration to previous configuration array
        copy_array(prev_configuration[k],configuration[k]);
    }

    //proportional selection
    for (k=0; k<POPULATION_SIZE; k++) {
        test=random()*fitsum;
        i=-1;
        cumul_prob=0.;
        while (test > cumul_prob) {
            i++;
            cumul_prob+=fitness[i];
        }
        //copy individual from previous generation to current generation
        copy_array(configuration[k],prev_configuration[i]);
    }

    //uniform stage crossover (half of selected population is cross-bred)
    for (k=0; k<POPULATION_SIZE/2; k+=2) {
        for (i=0; i<number_bits; i+=3) {
            if (random() > .5) {
                //trade first bit in stage…
                temp=configuration[k][i];
                configuration[k][i]=configuration[k+1][i];
                configuration[k+1][i]=temp;
                //then second bit…
                temp=configuration[k][i+1];
                configuration[k][i+1]=configuration[k+1][i+1];
                configuration[k+1][i+1]=temp;
                //then third bit
                temp=configuration[k][i+2];
                configuration[k][i+2]=configuration[k+1][i+2];
                configuration[k+1][i+2]=temp;
            }
        }
    }

    //stage mutation
    for (k=0; k<POPULATION_SIZE; k++) {
        for (i=0; i<number_bits; i+=3) {
            if (random() < STAGE_MUTATION_FACTOR) {
                configuration[k][i]=int(random()+.5);   //mutate bit 1…
                configuration[k][i+1]=int(random()+.5); //then bit 2…
                configuration[k][i+2]=int(random()+.5); //then bit 3
            }
        }
    }
}
```

```
    //determination of best individual from one population after full evolution
    best_fitness=0.;
    for (k=0; k<POPULATION_SIZE; k++) {
        test=compute_fitness(configuration[k]);
        if (test > best_fitness) {
            best_fitness=test;
            best_index=k;
        }
    }

    //determination of best individual from all populations separately evolved
    if (best_fitness > overall_best_fitness) {
        overall_best_fitness=best_fitness;
        //copy best individual configuration to overall best individual config
        copy_array(overallbestconfig,configuration[best_index]);
    }
}
```

APPENDIX

# C

# FORWARD KINEMATICS OF THE BRAID

This appendix details the kinematics of the BRAID structure, including homogeneous transformations and angles of deflection in the flexure hinges. A BRAID device is composed of a serial stack of parallel stages (see Figure 2.6). Each of these stages has three binary-actuated linkages, thus yielding three binary DOF per stage. This results in $2^3 = 8$ different binary configurations for a single stage (see Figure C.1). Because of the discrete nature of the binary robotic concept, many of the complex geometric computations need only be done once to be stored into memory for later use by the real-time planner and controller.

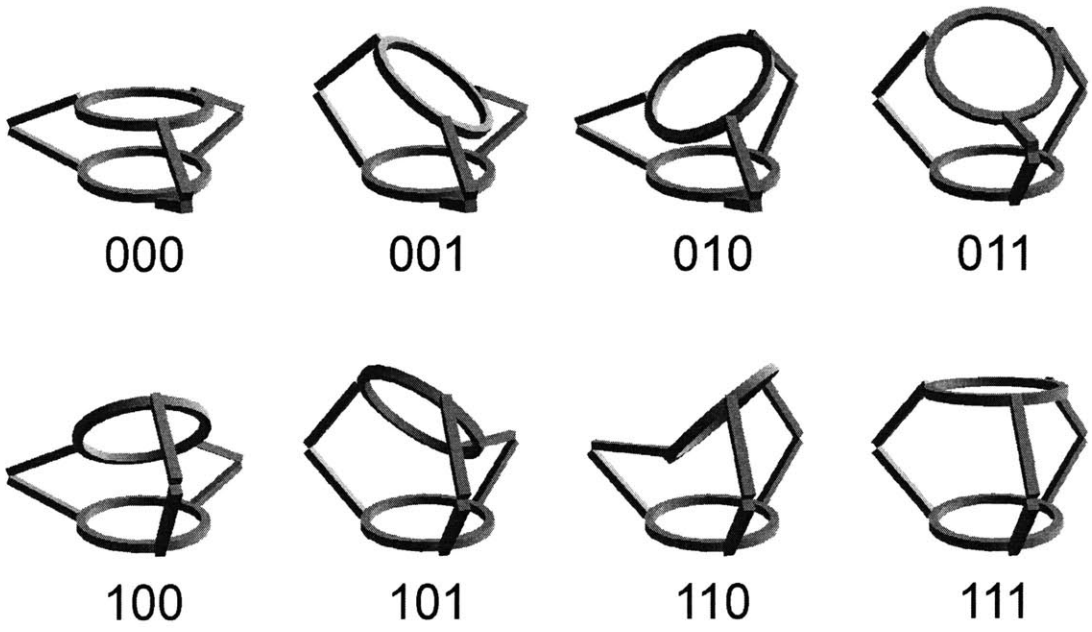


*Figure C.1. One stage of the BRAID, showing its eight binary configurations.*

As described in Section 2.3, it is convenient to use homogeneous transformation matrices that describe the position and orientation of the top ring of the stage (the end frame) relative to the bottom ring (the base frame). By precomputing all the possible transformations based on the binary configuration of the stage, it is easy to compute the forward kinematics of an *M*-staged device. To do so, one simply multiplies the transformations of each stage from the base to the end-effector of the device (see Equation 2.1, Section 2.3).

The convention used in the following equations implies that a base frame is situated at the center of the base ring (see Figure C.2). The x-y plane is coplanar with the base ring, with the z-axis normal to the ring. The end frame is situated at the center of the top ring, with the top ring in the x'-y' plane and the z'-axis normal to the top ring. The binary-actuated links are numbered in counterclockwise order when viewed from above, with link 1 being the link in the y-z plane.

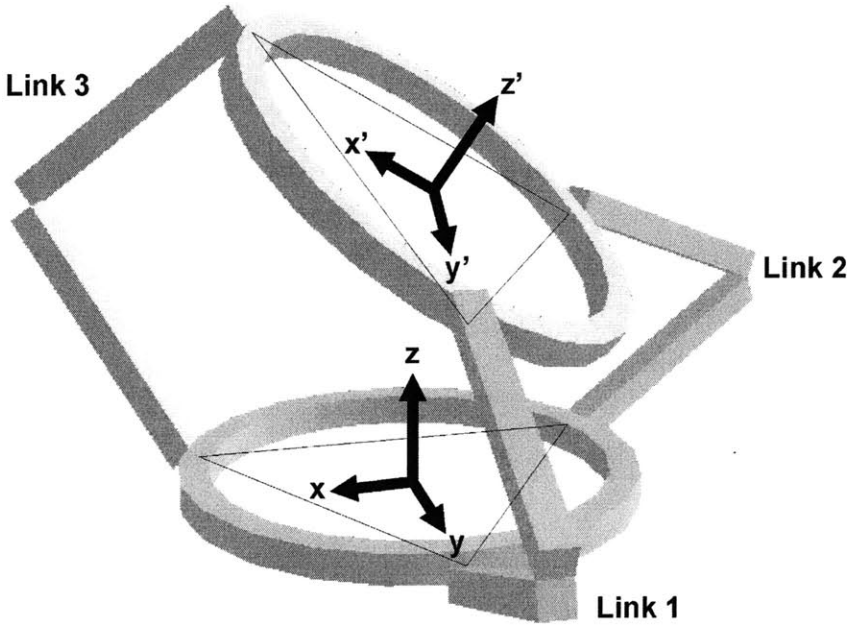


*Figure C.2. Coordinate frames for a single stage.*

In a single BRAID stage, there are 15 rotational hinges (five at each link). The axes of revolution of the five hinges in a single link are shown in Figure C.3.a. Axes 1, 2,

and 3 are parallel to the x-axis, with axis 1 being in the x-y plane. Axis 5 is parallel to the z-axis. Axis 4 is coincident with the y'-axis.

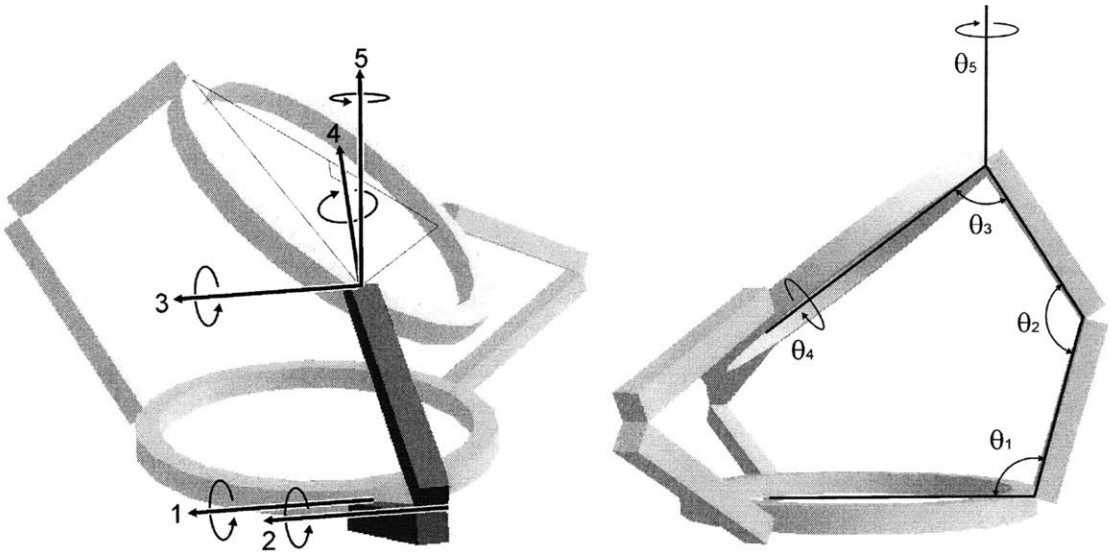


*Figure C.3. (a) Axes of revolution of the five flexural hinges in one link of a stage; (b) hinge angle definitions.*

## C.1 State (000)

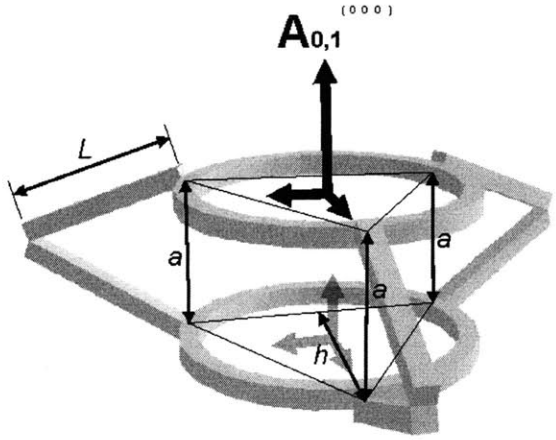


*Figure C.4. Geometry of stage in state 000.*

In state 000, all three links are in their closed position. The homogeneous transformation matrix describing the top ring relative to the base can be described as a translation in the z-direction as follows

$$A_{0,1}^{(000)} = translation(0,0,a) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The angles of each of the hinge joints can be computed as follows

$$\theta_1^{(000)} = \theta_3^{(000)} = \frac{\pi}{2} + \cos^{-1}\left(\frac{a}{2L}\right)$$

$$\theta_2^{(000)} = \pi - 2\cos^{-1}\left(\frac{a}{2L}\right)$$

$$\theta_4^{(000)} = \theta_5^{(000)} = 0$$

## C.2 State (111)

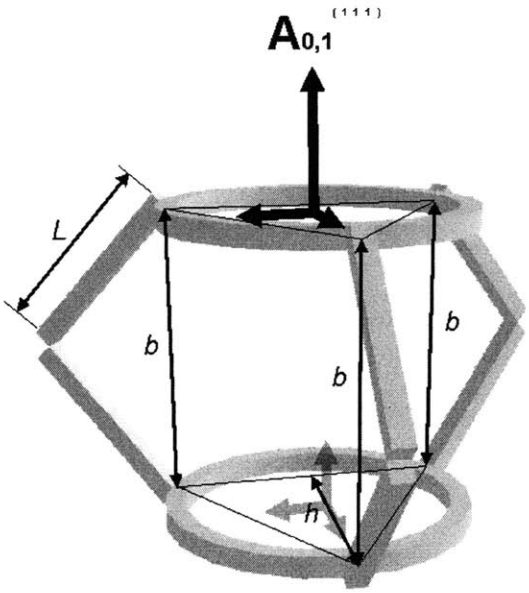


*Figure C.5. Geometry of stage in state 111.*

In state 111, all three links are in their open position. The homogeneous transformation matrix describing the top ring relative to the base can be described as a translation in the z-direction as follows

$$A_{0,1}^{(111)} = translation(0,0,b) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & b \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The angles of each of the hinge joints can be computed as follows

$$\theta_1^{(111)} = \theta_3^{(111)} = \frac{\pi}{2} + \cos^{-1}\left(\frac{b}{2L}\right)$$

$$\theta_2^{(111)} = \pi - 2\cos^{-1}\left(\frac{b}{2L}\right)$$

$$\theta_4^{(111)} = \theta_5^{(111)} = 0$$

## C.3 State (100)

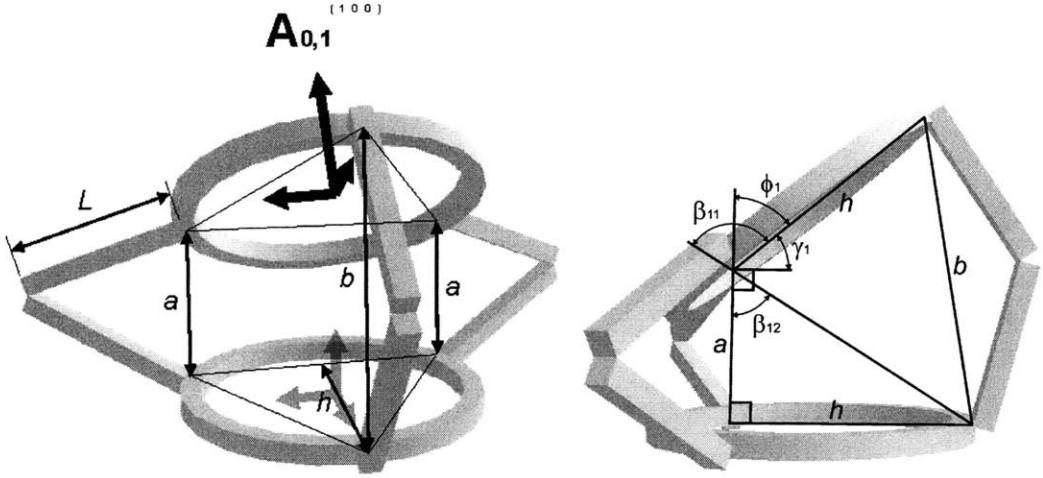


*Figure C.6. Geometry of stage in state 100.*

In state 100, link one is open and the others are closed. The homogeneous transformation matrix describing the top ring relative to the base can be described as a translation in the y- and z-directions followed by a rotation about the x'-axis as follows

$$A_{0,1}^{(100)} = translation\left(0, \frac{h}{3}(\sin\phi_1 - 1), a + \frac{h}{3}\cos\phi_1\right) * rotation(\gamma_1, 0, 0)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{h}{3}(\sin\phi_1 - 1) \\ 0 & 0 & 1 & \left(a + \frac{h}{3}\cos\phi_1\right) \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\gamma_1) & -\sin(\gamma_1) & 0 \\ 0 & \sin(\gamma_1) & \cos(\gamma_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where

$$\phi_1 = \beta_{11} - \beta_{12} = \cos^{-1}\left(\frac{b^2 - a^2 - 2h^2}{2h\sqrt{h^2 + a^2}}\right) - \tan^{-1}\left(\frac{h}{a}\right)$$

$$\gamma_1 \equiv \frac{\pi}{2} - \phi_1$$

The angles of each of the hinge joints in link 1 can be computed as follows

$$\theta_1^{(100)} = \tan^{-1}\left(\frac{a}{h}\right) + \cos^{-1}\left(\frac{b^2 + a^2}{2b\sqrt{a^2 + h^2}}\right) + \cos^{-1}\left(\frac{b}{2L}\right)$$

$$\theta_2^{(100)} = \theta_2^{(111)}$$

$$\theta_3^{(100)} = \cos^{-1}\left(\frac{b^2 - a^2}{2bh}\right) + \cos^{-1}\left(\frac{b}{2L}\right)$$

$$\theta_4^{(100)} = \theta_5^{(100)} = 0$$

## C.4 State (010)

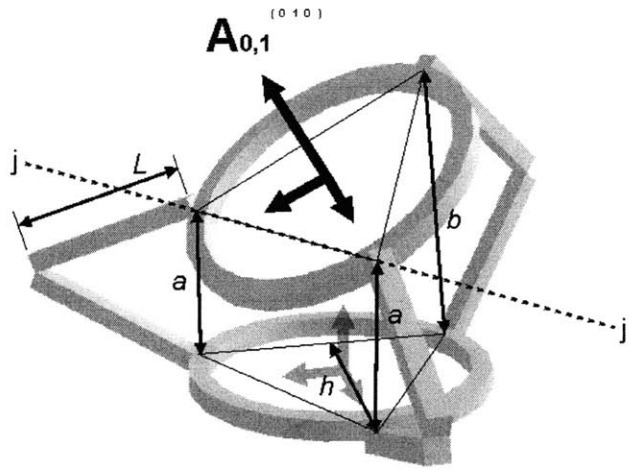


*Figure C.7. Geometry of stage in state 010.*

In state 010, link two is open and the others are closed. The homogeneous transformation matrix describing the top ring relative to the base is derived the same way as that for state 100, only this time the axes are rotated about the z axis first, then rotated back about the z' axis. That is

$$A_{0,1}^{(010)} = rotation\left(0,0,\frac{2\pi}{3}\right) * A_{0,1}^{(100)} * rotation\left(0,0,\frac{-2\pi}{3}\right)$$

$$= \begin{bmatrix} \cos\left(\frac{2\pi}{3}\right) & -\sin\left(\frac{2\pi}{3}\right) & 0 & 0 \\ \sin\left(\frac{2\pi}{3}\right) & \cos\left(\frac{2\pi}{3}\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & & 1 \end{bmatrix} * A_{0,1}^{(100)} * \begin{bmatrix} \cos\left(\frac{-2\pi}{3}\right) & -\sin\left(\frac{-2\pi}{3}\right) & 0 & 0 \\ \sin\left(\frac{-2\pi}{3}\right) & \cos\left(\frac{-2\pi}{3}\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & & 1 \end{bmatrix}$$

The transition from state 000 to 010 is pure rotation about axis *j-j* an angle $\gamma_1$ (see Figure C.6 for a depiction of $\gamma_1$). The angles of each of the hinge joints in link 1 can be computed as follows

$$\theta_1^{(010)} = \theta_1^{(000)}$$
$$\theta_2^{(010)} = \theta_2^{(000)}$$
$$\theta_3^{(010)} = \theta_3^{(000)} + \sin^{-1}\left(\tfrac{1}{2}\sin(\gamma_1)\right)$$
$$\theta_4^{(010)} = -\sin^{-1}\left(\tfrac{\sqrt{3}}{2}\sin(\gamma_1)\right)$$
$$\theta_5^{(010)} = \frac{\pi}{6} - \tan^{-1}\left(\tfrac{1}{\sqrt{3}}\sin(\gamma_1)\right)$$
$$\gamma_1 \equiv \frac{\pi}{2} - \phi_1 = \tan^{-1}\left(\frac{h}{a}\right) + \cos^{-1}\left(\frac{a^2 + 2h^2 - b^2}{2h\sqrt{a^2 + h^2}}\right) - \frac{\pi}{2}$$

## C.5 State (001)

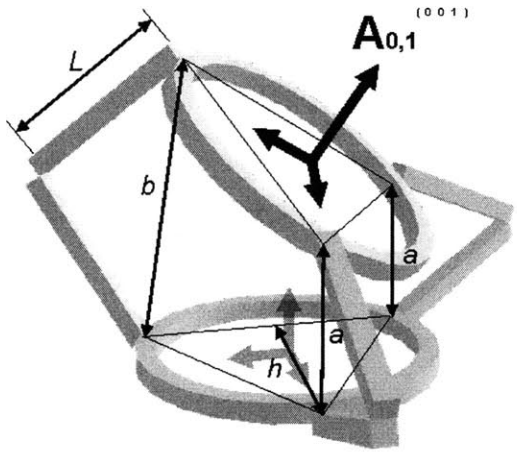


*Figure C.8. Geometry of stage in state 001.*

In state 001, link three is open and the others are closed. Again, the homogeneous transformation matrix describing the top ring relative to the base is derived the same way as that for state 100, and again the axes are rotated about the z axis first, then rotated back about the z' axis. That is

$$A_{0,1}^{(001)} = rotation\left(0,0,\frac{-2\pi}{3}\right) * A_{0,1}^{(100)} * rotation\left(0,0,\frac{2\pi}{3}\right)$$

$$= \begin{bmatrix} \cos\left(\frac{-2\pi}{3}\right) & -\sin\left(\frac{-2\pi}{3}\right) & 0 & 0 \\ \sin\left(\frac{-2\pi}{3}\right) & \cos\left(\frac{-2\pi}{3}\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & & 1 \end{bmatrix} * A_{0,1}^{(100)} * \begin{bmatrix} \cos\left(\frac{2\pi}{3}\right) & -\sin\left(\frac{2\pi}{3}\right) & 0 & 0 \\ \sin\left(\frac{2\pi}{3}\right) & \cos\left(\frac{2\pi}{3}\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & & 1 \end{bmatrix}$$

The angles of each of the hinge joints in link 1 are derived similarly to those for state 010 and are thus

$$\theta_1^{(001)} = \theta_1^{(010)}$$
$$\theta_2^{(001)} = \theta_2^{(010)}$$
$$\theta_3^{(001)} = \theta_3^{(010)}$$
$$\theta_4^{(001)} = -\theta_4^{(010)}$$
$$\theta_5^{(001)} = -\theta_5^{(010)}$$

## C.6 State (011)

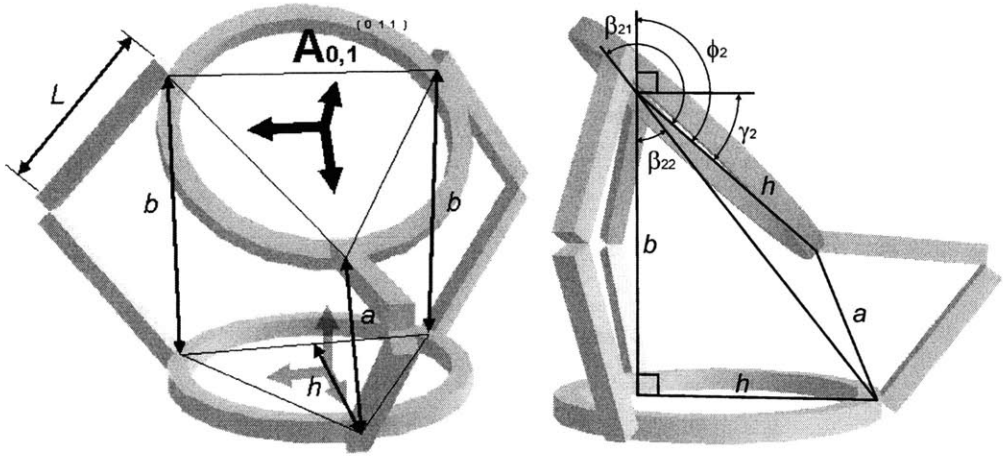


*Figure C.9. Geometry of stage in state 011.*

In state 011, link one is closed and the others are open. The homogeneous transformation matrix describing the top ring relative to the base can be described as a translation in the y- and z-directions followed by a rotation about the x'-axis as follows

$$A_{0,1}^{(011)} = translation\left(0, \frac{h}{3}(\sin\phi_2 - 1), b + \frac{h}{3}\cos\phi_2\right) * rotation(\gamma_2, 0, 0)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{h}{3}(\sin\phi_2 - 1) \\ 0 & 0 & 1 & \left(b + \frac{h}{3}\cos\phi_2\right) \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\gamma_2) & -\sin(\gamma_2) & 0 \\ 0 & \sin(\gamma_2) & \cos(\gamma_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where

$$\phi_2 = \beta_{21} - \beta_{22} = \cos^{-1}\left(\frac{a^2 - b^2 - 2h^2}{2h\sqrt{h^2 + b^2}}\right) - \tan^{-1}\left(\frac{h}{b}\right)$$

$$\gamma_2 \equiv \frac{\pi}{2} - \phi_2$$

The angles of each of the hinge joints in link 1 can be computed as follows

$$\theta_1^{(011)} = \tan^{-1}\left(\frac{b}{h}\right) + \cos^{-1}\left(\frac{a^2 + b^2}{2a\sqrt{b^2 + h^2}}\right) + \cos^{-1}\left(\frac{a}{2L}\right)$$

$$\theta_2^{(011)} = \theta_2^{(000)}$$

$$\theta_3^{(011)} = \cos^{-1}\left(\frac{a^2 - b^2}{2ah}\right) + \cos^{-1}\left(\frac{a}{2L}\right)$$

$$\theta_4^{(011)} = \theta_5^{(011)} = 0$$

## C.7 State (101)

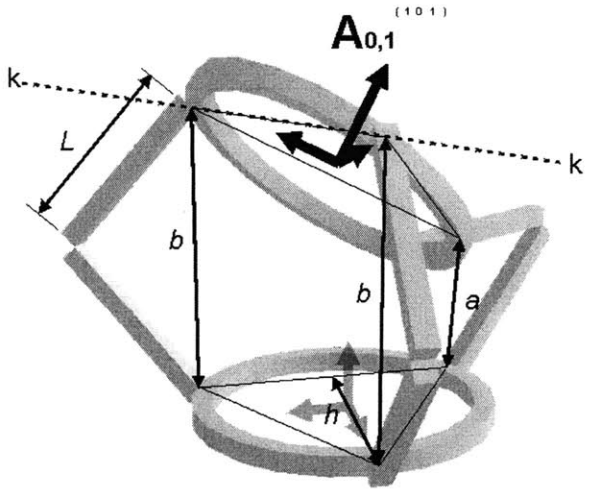


*Figure C.10. Geometry of stage in state 101.*

In state 101, link two is closed and the others are open. The homogeneous transformation matrix describing the top ring relative to the base is derived the same way as that for state 011, only this time the axes are rotated about the z axis first, then rotated back about the z' axis. That is

$$A_{0,1}^{(101)} = rotation\left(0,0,\frac{2\pi}{3}\right) * A_{0,1}^{(011)} * rotation\left(0,0,\frac{-2\pi}{3}\right)$$

$$= \begin{bmatrix} \cos\left(\frac{2\pi}{3}\right) & -\sin\left(\frac{2\pi}{3}\right) & 0 & 0 \\ \sin\left(\frac{2\pi}{3}\right) & \cos\left(\frac{2\pi}{3}\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & & 1 \end{bmatrix} * A_{0,1}^{(011)} * \begin{bmatrix} \cos\left(\frac{-2\pi}{3}\right) & -\sin\left(\frac{-2\pi}{3}\right) & 0 & 0 \\ \sin\left(\frac{-2\pi}{3}\right) & \cos\left(\frac{-2\pi}{3}\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & & 1 \end{bmatrix}$$

The transition from state 111 to 101 is pure rotation about axis *k-k* an angle $\gamma_2$ (see Figure C.9 for a depiction of $\gamma_2$). The angles of each of the hinge joints in link 1 can be computed as follows

$$\theta_1^{(101)} = \theta_1^{(111)}$$
$$\theta_2^{(101)} = \theta_2^{(111)}$$
$$\theta_3^{(101)} = \theta_3^{(111)} + \sin^{-1}\left(\tfrac{1}{2}\sin(\gamma_2)\right)$$
$$\theta_4^{(101)} = -\sin^{-1}\left(\tfrac{\sqrt{3}}{2}\sin(\gamma_2)\right)$$
$$\theta_5^{(101)} = \frac{\pi}{6} - \tan^{-1}\left(\tfrac{1}{\sqrt{3}}\sin(\gamma_2)\right)$$
$$\gamma_2 \equiv \frac{\pi}{2} - \phi_2 = \tan^{-1}\left(\frac{h}{b}\right) + \cos^{-1}\left(\frac{b^2 + 2h^2 - a^2}{2h\sqrt{b^2 + h^2}}\right) - \frac{\pi}{2}$$

## C.8 State (110)

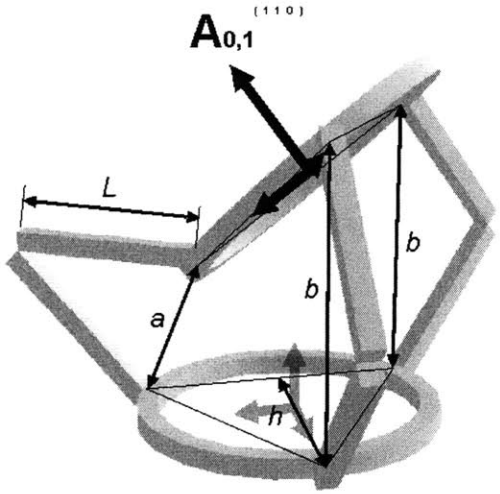


*Figure C.11. Geometry of stage in state 110.*

In state 110, link three is closed and the others are open. Again, the homogeneous transformation matrix describing the top ring relative to the base is derived the same way as that for state 011, and again the axes are rotated about the z axis first, then rotated back about the z' axis. That is

$$A_{0,1}^{(110)} = rotation\left(0,0,\frac{-2\pi}{3}\right) * A_{0,1}^{(011)} * rotation\left(0,0,\frac{2\pi}{3}\right)$$

$$= \begin{bmatrix} \cos\left(\frac{-2\pi}{3}\right) & -\sin\left(\frac{-2\pi}{3}\right) & 0 & 0 \\ \sin\left(\frac{-2\pi}{3}\right) & \cos\left(\frac{-2\pi}{3}\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & & 1 \end{bmatrix} * A_{0,1}^{(011)} * \begin{bmatrix} \cos\left(\frac{2\pi}{3}\right) & -\sin\left(\frac{2\pi}{3}\right) & 0 & 0 \\ \sin\left(\frac{2\pi}{3}\right) & \cos\left(\frac{2\pi}{3}\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & & 1 \end{bmatrix}$$

The angles of each of the hinge joints in link 1 are derived similarly to those for state 101 and are thus

$$\theta_1^{(110)} = \theta_1^{(101)}$$
$$\theta_2^{(110)} = \theta_2^{(101)}$$
$$\theta_3^{(110)} = \theta_3^{(101)}$$
$$\theta_4^{(110)} = -\theta_4^{(101)}$$
$$\theta_5^{(110)} = -\theta_5^{(101)}$$