THESIS REPORT
Master's Degree

SYSTEMS
RESEARCH
CENTER

Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), the University of Maryland, Harvard University, and Industry

Analysis of Robotic Grasp with Application to the Modular Dextrous Hand

by M.E. DeSilets
Advisor: P.S. Krishnaprasad

M.S. 92-11

ABSTRACT

Title of Thesis: ANALYSIS OF ROBOTIC GRASP
WITH APPLICATION TO
THE MODULAR DEXTROUS HAND

Name of degree candidate: Mark E. DeSilets

Degree and Year: Master of Science, 1992

Thesis directed by: P. S. Krishnaprasad
Professor
Electrical Engineering

We present an analysis of robotic grasping and apply it to the specific problem of using the Modular Dextrous Hand (MDH), developed at the Systems Research Center. The goal is to provide the capabilty to automatically select, analyze and implement a firm grasp for arbitrarily shaped objects. A geometrical grasp selection algorithm, based on the notion of form closure is presented and shown to have good properties. A method for analyzing grasps is presented and a method for determining finger torques and verifying grasps against task induced stability requirements is described. An algorithm to implement path planning to drive the hand to a grasp configuration while avoiding finger self-collision is presented. These capabilities are combined to provide a complete description of the grasping process. In addition, a description of the hardware and software architecture built to support the MDH is provided.

ANALYSIS OF ROBOTIC GRASP

WITH APPLICATION TO THE MODULAR DEXTROUS HAND

by

Mark E. DeSilets

Thesis submitted to the Faculty of the Graduate School
of The University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1992

Advisory Committee:

Professor P. S. Krishnaprasad, Chairman/Advisor
Assistant Research Scientist J. Lončarić/Coadvisor
Associate Professor W. P. Dayawansa

DEDICATION

This thesis is dedicated foremost to my lovely wife, Lenore, without whose help and support this effort would not be possible. I also dedicate this work to my parents, Rex and Carol who taught me the beauty of the analytical and inquisitive life.

ACKNOWLEDGEMENTS

I am indebted to both my research advisors, Dr. P.S. Krishnaprasad and Dr. Lončarić, for their continuous support, encouragement and contribution throughout the course of this work. I am especially thankful to Dr. Krishnaprasad for his advice for attacking several of the problems herein. I have yet to find a subject on which he is not an expert.

I would like to thank Dr. W. P. Dayawansa for serving as a member of my thesis advisory committee, my colleague Yannis Salmatjidis for his electronic design expertise in implementing the prototype hand control hardware.

This research was supported in part by the National Science Foundation's Engineering Research Centers Program: NSFD CDR8803012.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The problems associated with finding grasping strategies for robotic hands has occupied considerable research effort in recent years. Attempts to provide truly general approaches to grasp kinematics and grasp synthesis have been provided in such examples as [1], [2],[3], and [4]. A great deal of attention to the various issues surrounding grasp stability are exhibited in (among others) [5],[6], [7], [8], [9], [10], [11], [12], [13], and [14]. While we rely on some of these results to formulate a general approach to robotic grasping, we will focus our work on the more specific goal of solving the problems addressed in these works in the context of providing a grasping strategy for a particular device, the Modular Dextrous Hand.

This document describes a full methodology for applying the Modular Dextrous Hand to the problem of grasping an object. The goal is to present the complete story of how a grasp is selected and how it is implemented. Most importantly, we show that grasps obtained by these methods have desirable characteristics. Namely that the grasps are, in some respect, stable under quantifiable loading conditions. Included here are the algorithms which implement the grasp selection strategy and which position the fingers of the hand in such a way that they may attain the desired grasp configuration, as well as analysis which allows determination of whether the grasp will allow the grasp to be maintained despite quasi-static loads being applied to the object.

In order to provide the reader with a sense of what the Modular Dextrous Hand is, and the way in which it is controlled, Chapter 2 provides a description of the hand itself and of the control architecture we are building for it. A prototype implementation of the hand control has been built and tested, providing confirmation of the validity of the concepts employed. The information in this chapter is provided to allow us some liberties in later chapters, with knowledge that the physical system we are working with has been explained to the extent necessary for understanding its effect on grasp selection, implementation and analysis.

Given an object, the first step in grasping it is to detemine at which points of the object's surface the fingers of the hand must be placed in order to provide a secure grasp. In Chapter 3, we describe a method for selection of grasps, given a characterization of the object's surface. Following the lead of Markenscoff, Li and Papadimitriou [1], we demonstrate that the contact points selected provide a *form closure grasp* of the object and describe what that means for the security of the grasp. Out algorithm is based on the idea of immobilizing the maximal inscribed circle of an object boundary, which stems from [1], as well as having been mentioned in [6], in which explicit mention of the stability properties of such grasps is made. We present several examples of the application of the algorithm to the selection of grasp points for a variety of objects and show the grasp points thus selected. We also briefly describe how the grasp selection algorithm is implemented.

The fact that we can determine a set of feasible contact points and place the hand in a configuration which achieves the contacts does not ensure that we will actually be able to pick up the object, nor that we will be able to subject it to arbitrary accelerations as we move it, nor that we can subject it to arbitrary forces as we attempt to perform some task. It is incumbent on us, then, to determine whether the grasp can be maintained in the presence of disturbing forces and torques, provided, for example by static and dynamic loading, and by the use of the grasped object in a task specific enironment (such as screwing a nut onto a bolt). These notions require us to consider the stability of the grasp we have selected. In Chapter 4, we present some important notions of what it means to have a stable grasp. Since our grasp selection algorithm makes use of a two-dimensional analysis of the object cross-section, it lends itself to the analysis of grasp stability from a two-dimensional perspective. We begin by considering grasp stability in this reduced domain, which contains many of the interesting issues, and then see how we can generalize these results to the three-dimensional case. We note that Baker, Fortune and Grosse [6] address the problem of grasp stability for what they call the "triangular grip" and "parallel grip", which correspond to the types of grasps selected by our maximal inscribed circle algorithm. They do not deal, however, with issues arising from the potential non-normality at the contact point of forces applied by the fingers, which we must address given the physical limitations of our device.

In Chapter 5, we turn to the problem of actually achieving the grasp we have selected. Obtaining the grasp is complicated by the fact that there may be several configurations of the hand which allow the same contact points to be used and by the fact that in attaining the grasp configuration, we have to concern ourselves with ensuring that the fingers do not collide with themselves, or with the object. The set of possible hand configurations that can achieve a particular grasp is determined using inverse kinematics. A description of this process and of its implementation is presented as well as some heuristics for selecting the best from among the set. The problem of self-collision avoidance deals with the issue of maneuvering the fingers of the hand into a pre-grasp configuration. Since the fingertips may actually collide over a wide range of their configuration space, we must provide an efficient algorithm for steering the fingers from an arbitrary initial configuration to an arbitrary goal position.

Chapter 2

The MDH and Its Control Architecture

The purpose of this chapter is to present a description of the Modular Dextrous Hand (MDH). We will first describe the physical properties of the MDH, itself. We next describe the electronics we built for obtaining data, coordinating transfer to and from the host computer, synchronizing events and sending commands to the hand. While not a significant part of the theoretical aspects of this project, this architecture development process comprised a large portion of its efforts. This chapter will help document the design work so that future enhancements can be naturally added. Finally, we will describe the software built for control of and interface with the MDH. This software embodies the solutions to a number of challenging engineering problems, thus meriting its presentation.

2.1 MDH Physical Characteristics

The Modular Dextrous Hand (MDH) is the general purpose robotic hand developed at the University of Maryland, first described in [15]. It is referred to as *modular* because the tasks of gripping an object, and of manipulating an object are apportioned to two distinct stages of the device. The primary stage of interest for this thesis is the gripping stage, which is composed of three one degree-of-freedom fingers actuated through revolute joints whose axes are perpendicular to the plane of the palm. The secondary stage, when complete, will consist of a Stewart Platform, or similar parallel platform, which will provide full 6 degree-of-freedom fine manipulation capability. While the skillful manipulation of grasped objects will undoubtedly occupy other researchers for some time, the use of the gripper stage for the efficient grasp of objects of almost arbitrary shape is our focus.

2.1.1 MDH Fingers

Figure 2.1 provides a diagram of the MDH indicating its major features. The fingers are composed of hollow aluminum tubes of 0.5 in. diameter, carefully radiused in such a way that the distal end of the fingers are parallel to their axes of rotation. The proximal portion of the finger tubes provide a lever arm of length 3.0 in. The vertical portion of the finger tubes extends 4.25 in from the

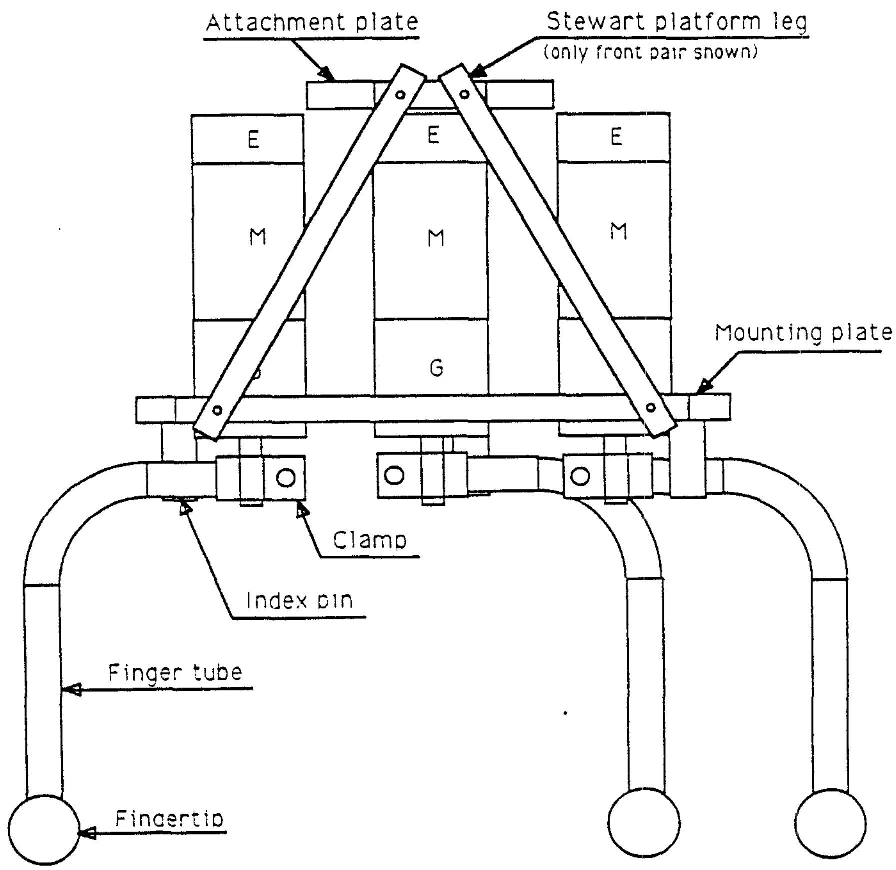


Figure 2.1: MDH Diagram

top of the top surface of base tube. Figure 2.2 is an engineering diagram of the finger tube assembly, indicating these as well as other important dimensions.

The fingers are mounted at the vertices of an equilateral triangle of side length roughly $3\sqrt{3} \approx 5.2$ inches, such that the distance from the axis of rotation of the finger to the centroid of this triangle is 3 in. Driving the fingers are Pittman 9433 motors with planetary gearheads with 24:1 ratio and HDES encoders with 500 lines per revolution. The approximate maximum output torque for these motors is 3.5Nm. This results in the maximum force which can be

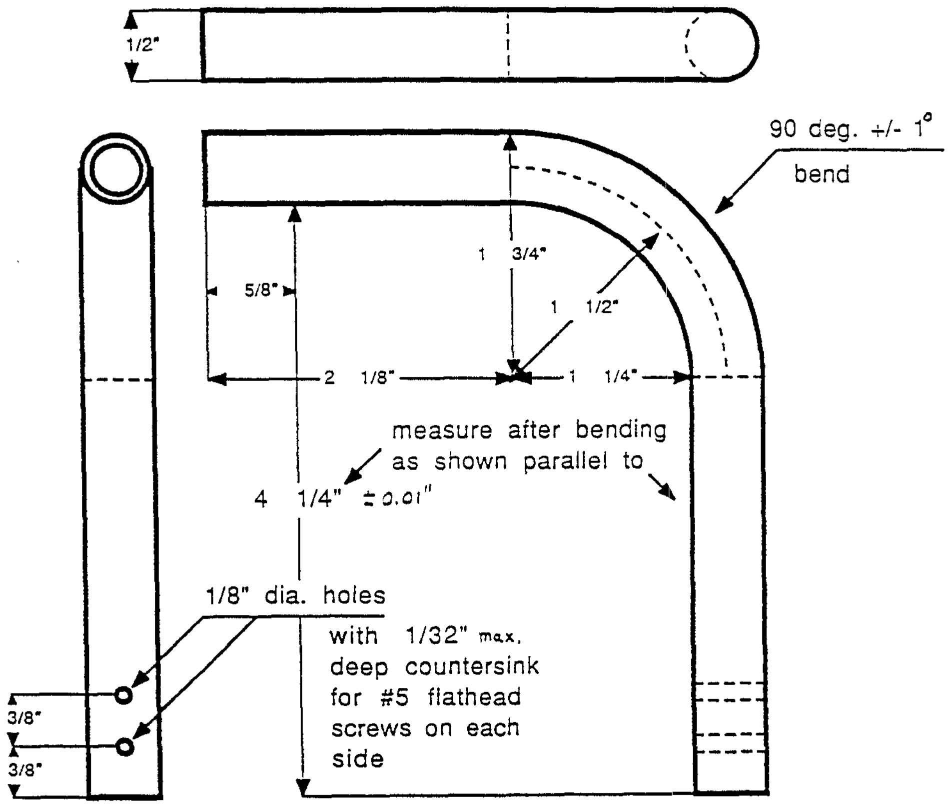


Figure 2.2: MDH Finger Tube Engineering Diagram

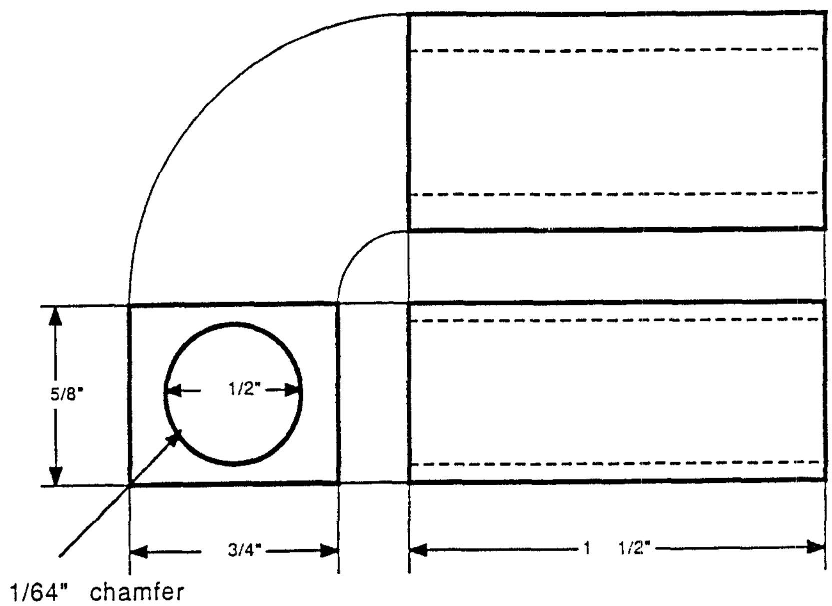


Figure 2.3: MDH Finger Tube Clamp Engineering Diagram

applied at the fingertips being on the order of 45.93N. The fingers are coupled to the motor shaft by the clamp mechanism indicated in Figure 2.3, which is mounted to the palmar base plate by four mounting screws. The base plate, itself is machined from 0.375 in. aluminum, which, for the purposes of the analysis of the hands compliance, can be considered fully rigid.

The fingers are kept from rotating through a full 360 degrees by index pins mounted near the outer edges of the palmar base plate. The presence of these pins provides a convenient mechanism for absolute position registration at power up time, and as needed to compensate for accumulating drift errors during operation. On the down side, however, is the fact that there is only one possible trajectory for each finger to achieve a specified position, which gives rise to the need for more involved path planning algorithms, as is explained in Chapter 4.

2.1.2 MDH Fingertips

The MDH fingertips provide an active feedback of grasping forces. Embedded in each finger tip assembly is a pressure sensitive sensor. The tips themselves are hollow rubber spheroids held in place by the ingeneous mechanism depicted in Figure 2.4. Since the fingertips are airtight, deformations of the surface of the fingertip spheroids result in a change in their internal volume, giving rise to a change in internal pressure. This pressure reading is fed through analog channels to the upstream processing hardware for feedback evaluation of the grasp characteristics.

2.2 MDH Measurement and Control Electronics

The measurement and control electronics system for the Modular Dextrous Hand is under construction as of this writing. The system will be built to specifications described in this section, with many of the major components already complete. A complete prototype breadboard implementation of the control system has verified the concepts we employ in the design of the actual system, so we speak with confidence of the capabilities of the actual implementation.

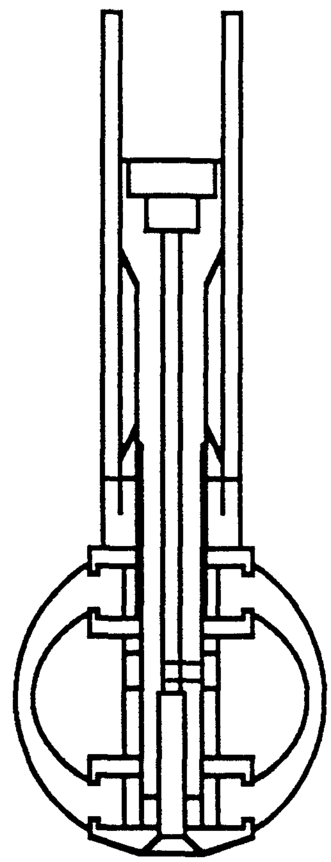


Figure 2.4: MDH Finger Tip Engineering Diagram

The measurement and control electronics system to be built for the Modular Dextrous Hand is a multicomponent hybrid network, comprised of commercially available as well as custom built components. Figure 2.5 presents a high-level diagram indicating the major system components, and can be used as a reference to the description that follows. At the highest level of abstraction, there are three components – the MDH, itself, the VXI bus subsystem and the network of Hewlett Packard HP 9000 workstations.

The latter of these components supports "high-level" computing needs. The primary station, labeled HP 835 TSRX (skylab) on Figure 2.5, acts as the user's command station. It is here that the user provides task level input to the system, initiates simulations of the various required subtasks, installs task specific control schemes and monitors the progress and state of the system. The remaining workstations on the network are used for remote processing, e.g. path planning, to computationally off-load the primary station. The workstation network interacts with the VXI bus subsystem by downloading subtask information to it and receiving measurement data, error information and status from it.

The VXI bus subsystem is the heart of the control and measurement system. VXI an extension to the VME standard (VXI stands for *V*ME e*X*tension for *I*nstrumentation). It provides an internal high speed data transfer bus, as well as a number of dedicated lines for simultaneous triggering of multiple devices and handling of interrupts. Installed in the first slot of the VXI subsystem

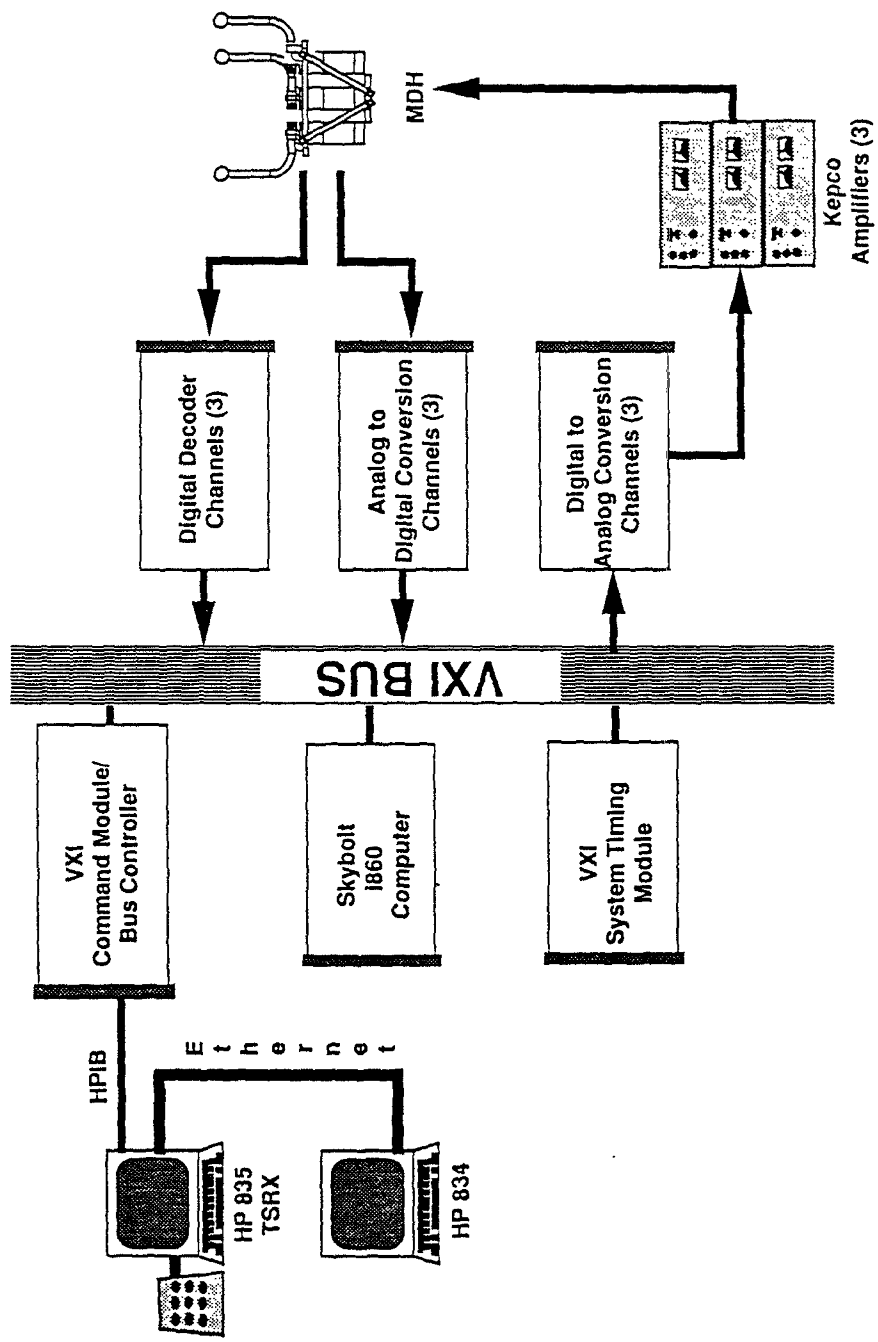


Figure 2.5: MDH Electronics: Top-Level Diagram

is a VXI command module, whose responsibility is to oversee communication between the VXI subsystem and the host computer, as well as to act as bus master. In its former role, the Command Module implements a full IEEE-488 standard bus port for direct VXI-to-host communication. In its latter role, it fulfills the standard requirements of granting bus access, arbitrating bus conflicts and handling interrupts.

The most important device in the VXI bus subsystem is the Skybolt [1] computer. The Skybolt is an extremely fast computer, based on the Intel i860 and i960 processors. With the 960 processor handling the data transfer tasks, and the 860 processor reserved for computationally intensive tasks, the Skybolt can run at a peak computational rate of 80 MFLOPS, and, given the computational environment determined by MDH control requirements, we estimate a sustained rate of about 8 MFLOPS. As a component of the MDH system, the Skybolt is responsible for performing the low level control functions. That is, it gets control code downloaded through the Command Module from the host HP 9000, gets measurement data from the custom circuits in the VXI subsystem and uses these to determine the necessary commands to be sent, via the D/A channels to the MDH. In addition, the Skybolt performs all necessary history maintanance and periodically transmits reports of progress to the host, for use in task monitoring.

The remaining components of the electronics suite are custom-built circuits.

[1] Skybolt is a registered trademark of Sky Computers, Inc.

These include a digital sampling board, whose function it is to store and synchronously sample the outputs of the shaft encoders for measurement of finger position. Also included are paired channels of D/A and A/D conversion circuitry. The A/D channel can be used either to capture motor back-EMF data, or to capture force information from the analog pressure sensors installed in the MDH finger tips. The D/A channels provide a gateway to the amplifiers whose outputs directly drive the finger motors. Additional custom circuitry includes a clock generator/synchronous trigger circuit and an address translation module which maps the 24 bit addresses used by the VXI Command Module to the 32 bit address format required by the Skybolt computer.

2.3 MDH Software for Control and Interface

2.3.1 Overview

In this section we will describe, at a superficial level, the overall architecture of the software designed for use with the MDH. Although not the focus of this project, it was necessary to develop this high-level architecture to provide a view of the way in which the grasping software would interact with the production system. The individual software components we have built have been designed with system interface issues clearly defined for ease of integration when the full system is implemented. Prototype designs for major components of the overall system have been implemented and successfully tested, validating the design concepts herein described. Final implementation of the system software is an ongoing project.

Figure 2.6 provides a high level module interaction diagram which may be a useful reference for this discussion. There are a number of goals for this architecture which need to be understood in order that the architecure makes sense. They are:

1. The architecture should be general enough to support the variety of robotic devices envisioned for current and future lab experiments.
2. The architecture should provide reliable control of the MDH.
3. The architecture should support networked operation, and be flexible about the precise configuration of the underlying hardware.
4. The architecture should be modular, so that support programs may be replaced to suit particular user needs. Thus the definition of support program interfaces is an architectural concern.
5. The architecture should support user configurable display. That is, there should be a suite of display types from which a user may select.

While these goals are fairly broad, they have pronounced effect on many of the decisions made about the MDH software.

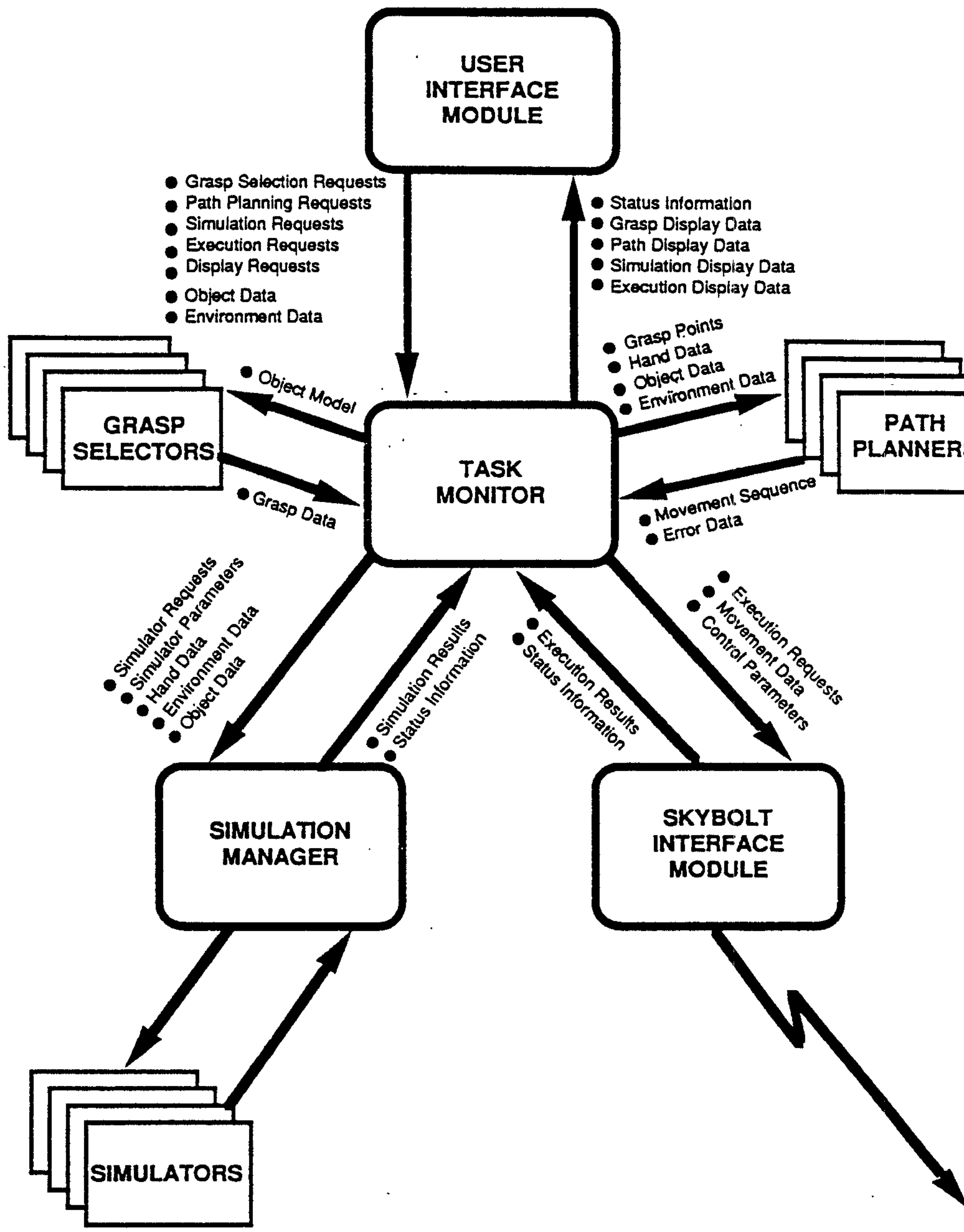


Figure 2.6: MDH Software Architecture: Top-Level Diagram

The first goal is really a requirement, although exactly what is meant by "reliable" control is ambiguous. Since the precise control scheme is determined by the user, there is no assurance that the control will be "good". What is meant is that the appropriate measurements will be reliably taken at required times and that command outputs will be initiated at required intervals. This allows the control code designer to assume this level of integrity, which should vastly simplify the control task. The architecture will specify a set of interface requirements for the control code, including how measurements may be taken, how commands may be issued and how status information may be returned to the higher levels of the system and, eventually, to the user. For the most part, these functions will be encapsulated in functions made available as an object library for rapid development of control programs.

The goal of providing networked operation of the MDH software is important, since the complete system is too computationally intensive for all of the software to run on one workstation. Of course, this is due, in part, to the fact the we are designing a general system for control experimentation. A production controller would be much simpler, and therefore much less computationally intensive. Ideally, for use in experimentation, a large network of workstations would exist, allowing each program to run on its own CPU. Since this is not always feasible, the architecture must be flexible enough to allow the software to make use of as much computing resource as possible, allowing for the possibility that this might be a dynamically variable quantity. This requires building into the software a degree of configuration independence, which makes the system design more complex, but greatly simplifies the job of the support program designer and of the end-user.

Modularity is essential for the MDH system. This is due to the fact that this software is meant to be a general purpose control system exploration package. That is, many users will wish to experiment with different control programs, different grasp selection programs, different path planning algorithms, different simulators and, in fact, different types of programs not needed for the MDH, but which may be useful for other devices to which this system may be adapted. This modularity is achieved by encapsulating interface communication and data requirements so that, although two programs may perform quite differently, they plug into the same "slot" in the architecture and, from a system perspective, appear to be a "black box."

There is simply too much information to have it all displayed on the screen at once. This gives rise to the need to have the user select the types of information that he is interested in and which are pertinent to the type of experimentation he is attempting. There will, therefore, be a specific set of display types, with specific data requirements, which will allow different types of information to be displayed in the appropriate ways. For example, the user might wish to see an animated replay of a simulation, requiring a real-time animated display screen. Other types of informational displays might include real-time graphs of finger position, or of position error. The specific displays are not what is important here. What is important is to note that there will be a number of display types that will be required. The job of providing a flexible user interface architecture is to allow easy incorporation of these constructs into support programs.

2.3.2 The Task Monitor

Referring, again, to Figure 2.6, we see that the central module in the MDH software is the Task Monitor, or TM. The TM is responsible for coordinating requests from other modules and ensuring there completion. This requires the TM to be in continuous, real-time contact with all of the other modules. It is also responsible for facilitating data transfer amongst modules. Since these modules may be dispersed on the network, this requires the TM to maintain either information about, or direct communication links to, all other modules in the system. Finally, the TM is responsible for maintaining the system's "world model". That is, the TM maintains a database in which are recorded all known information about the MDH, itself, about the object to be grasped, and about any pertinent environmental factors such as other objects which might interfere with the grasp attempt or with the object's manipulation.

2.3.3 The User Interface Module

This module, known as UIM, is responsible for displaying on the user screen, all the information selected by the user. It is also responsible for obtaining input from the user and transferring it to the TM, as appropriate. Such information includes configuration requests, such as requests to connect specific support programs. It also might include requests for specific parameter settings. For each of the displays that the UIM is actively maintaining, there must be an open channel with the TM for the approprate data to flow through. The creation and maintenance of these channels is the responsibility of the UIM.

2.3.4 The Simulation Manager

The Simulation Manager (SM) exists to provide a common interface for simulators. That is, since all of the simulators for the MDH provide a subset of the possible data, by providing a common interface, the TM can treat all simulators the same way. That is, the TM connects a simulator by requesting its connection through the SM. Otherwise, it deals with all simulators in the same way, sending input parameters and returning simulation results. The SM, on the other hand, needs to deal with formatting data in the particular form needed for each simulator and for formatting the output from each simulator in a uniform way.

2.3.5 The Skybolt Interface Module

The Skybolt Interface Module (SIM) provides a reliable interface to the Skybolt computer resident in the VXI bus subsystem (see Section SECNO). This task includes the downloading of software into the Skybolt's memory and sending parameter values and other data to the Skybolt for processing. It also included retrieving execution history data from the Skybolt and returning it to the TM.

2.3.6 The Support Programs

The Simulators, Grasp Selectors and Path Planners are all referred to as support programs, because they are considered replaceable from a system standpoint. Later in this thesis, we shall describe a specific Grasp Selector and a specific Path Planner developed as part of this work. Some Simulators have already been developed by our colleagues. It is envisioned that these support programs will be interchangeable components, allowing the user to select which of them he wants active in the configuration at any particular time. The specific requirements for writing suitible support programs are documented elsewhere.

Chapter 3

Grasp Selection

3.1 Geometric Notions of Grasp

We turn, now, to the issue of how we automate the process of selecting a "good" grasp. Before we can be successful in this endeavor, we must determine how to characterize a grasp as being "good", or at least "good enough" to accomplish a given task. In this chapter, we address this issue from a simple, geometric perspective. We will establish a geometric argument that a grasp which achieves *form closure* will be good in the sense that the object is constrained to immobility if large enough forces can be generated at the contact points. The notion of form closure is attributed to Reuleaux [16], but we will generally follow the perspective of Markenscoff et al [1] in developing the general theory, and then discuss its applicabiltity to selecting a grasp to be implemented with the MDH. We note here, that some simplifying assumptions will be made to present this argument clearly. The goal for this chapter is to present an algorithm based on achieving form closure grasp and to argue that this is probably a "good" grasp in some sense. When we turn to a more mathematical discussion of grasp stability in Chapter 4, we will be able to show that the grasps selected by our algorithm have some good stability properties and indicate quantitative conditions under which stability is obtained.

In selecting a grasp by a three-fingered hand on an arbitrarily shaped object, we are required to select three points of contact on the object's surface and three generalized forces to be applied via the contact to the object to hold it fixed with respect to the hand. We solve this problem reductively. If we assume that arbitrary forces may be applied, then the problem of selecting a grasp is reduced to simply choosing the contact locations. Since the three contact points will uniquely specify a plane, we could reduce the problem to selecting three grasp points for a two dimensional object assuming we know how to select the plane which should be chosen to contain the grasp.

Thus we have reduced the problem from that of achieving a stable grasp for an arbitrary three-dimensional object to that of selecting a form closure grasp for a two dimensional object. We will then synthesize a grasp procedure for three-dimensional objects with the two dimensional grasp as a starting point. We will start by assuming that we already know which plane contains the grasp points. We present an algorithm for selecting a form closure grasp in two dimensions, which assumes contacts of lower order than are actually supplied by the MDH fingertips. We will demonstrate that form closure grasp in two dimensions has the capability to completely restrain a two dimensional object. We will then argue that the higher order contacts provide true three dimensional restraint under specific geometric conditions local to the grasp points. These conditions will provide us insight to solving the problem of selecting the grasp plane and we shall present a heuristic method for its selection, although detailed investigation of this process is beyond the scope of this work. We will then investigate the grasps selected by our method in the context of the forces which may be actually transmitted by the MDH, which are of limited magnitude and constrained in direction.

3.2 Selecting a Two-Dimensional Form Closure Grasp

For the purposes of this section, let us assume that we are simply looking for a firm grasp of a two dimensional object. By object, we mean a closed, bounded, non-degenerate, two dimensional body, with a boundary that is connected and piecewise smooth. Non-degenerate refers to the condition that

$$\overline{\mathring{O}} = O \tag{3.1}$$

where O is the object, $\overset{\circ}{O}$ denotes the object interior and $\overline{\overset{\circ}{O}}$ denotes the closure of the interior.

What this implies is the absence of one-dimensional "horns". By insisting that the boundary be piecewise smooth, we require that there be only finitely many boundary derivative discontinuities, allowing us to treat any that exist in a sequential fashion, as exceptions to the general local boundary geometry. The reasons for requiring closure, boundedness and connectedness are obvious.

We will need to use a generalized notion of a force, good for describing both translational forces and torques in a uniform way. The notation used is that of a *wrench*, based on the descriptions in [17]. A wrench in two dimensions is vector in $\Re^3$ whose components are the force x and y components and moment (about the z axis). Conceptually, a wrench consists of a force with a fixed line of action, or as a pure moment in those cases where there is no net translational force being applied. We shall later have need of the three-dimensional analogue, which is a vector in $\Re^6$, with the first three components indicating the force projections on the coordinate axes, and the last three indicating the moments about the axes. We call a wrench a *unit wrench* if the first two components form a unit vector, i.e. the magnitude of the force is 1 newton.

We break, here, with the convention described in [1], and denote by $N(O)$ the set of unit normal wrenchs defined at the points of ∂O and directed towards the interior of O. That is $p \in \partial O \Rightarrow N(p) \in N(O)$. This definition is unambiguous at all but finitely many points of ∂O, due to our insistance that ∂O be piecewise smooth. At points whose neighborhoods on ∂O are smooth, the unit normal wrench is simply a force of magnitude one, located at the point whose direction is perpendicular to the tangent of ∂O at the point. To complete the description of $N(O)$, we need only consider the finitely many derivative discontinuities on ∂O. If p is a point of concave discontinuity, then a point obstacle at p can exert forces anywhere in the convex cone generated by the normals immediately "left" and "right" of p [1]. Consequently, the set $N(p)$ consists of all of the unit wrenches in this convex cone. If p is a point of convex discontinuity, the only possible directions for force application are those of the left and right normals. Therefore, $N(p)$ would consist of these two unit normal wrenches. We note here in passing that points of convex discontinuity would never be selected by the algorithm we present, but we include them in the discussion for the sake of completeness. Intuitively, one may think of $N(O)$ as being the set of all unit wrenches that may be applied by idealized frictionless point contacts at points in the boundary of O.

Since our grasp is to be realized by forces directed towards the interior of O, we require that the wrench intensities be positive, otherwise contact will be broken. Intuitively, the grasped object will be constrained to immobility if any force or moment on the object can be countered by the appropriate application

[1] We will make use of the abused notation "left" and "right" normals of a point to denote the limiting normals as the point is approached from the clockwise, and counter-clockwise directions, respectively.

of grasp forces by the fingers of the hand. Mathematically, this implies that the fingers can generate wrenches of arbitrary magnitude, direction and moment, using only wrenches of positive intensity. That is, if we think of two dimensional wrenches as vectors in $\Re^3$, the condition needed is that any vector can be expressed as the positive linear combination of some of the vectors in $N(O)$. A set of vectors satisfying this condition is said to positively span the space ($\Re^3$ in this case), and is known as a positive spanning set. We note, in passing, the trivial fact that the dimension of a positive spanning space must be greater than the dimesion of the space, itself. We call a subset $G(O) \in N(O)$ a form closure grasp if its members positively span $\Re^3$. The determination of one or more candidate sets $G(O)$ is the two dimensional form closure selection task we now turn to. In the two dimensional case, at least, it should be clear why form closure grasp constitues a grasp that is "good".

In the remainder of this section we intend to show that, for a two dimensional object that is not a circle, a form closure grasp can always be attained with a maximum of four normal wrenches. We wish to draw attention to the role played in the proof by the maximal inscribed circle for establishing the complete restraint of the grasped object. It is this role which sparked the idea of using the maximal inscribed circle as a tool for selecting an initial set of grasp points. Also remaining in this section is the task of showing how the achievement of form closure grasp, requiring as many as four frictionless point contacts is possible with three (or fewer) soft finger contacts, as are used on the MDH.

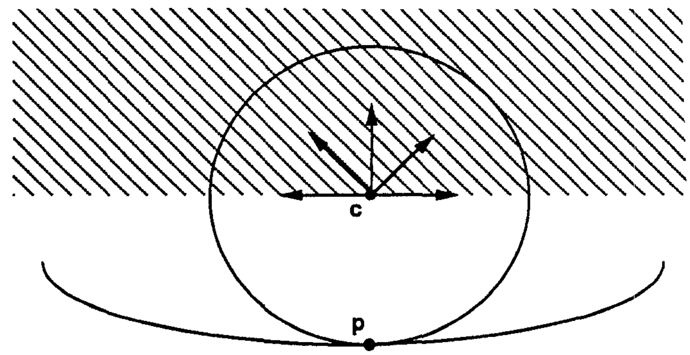


Figure 3.1: Movement of c, Constrained to an Open Half Plane by Contact

The main result we wish to show is:

Theorem 1 *(Markenscoff, Li and Papadimitriou) For any two dimensional object, O, that is not a circle there is a set of four unit normal wrenches in $N(O)$ that is a form closure.*

Thr crux of the proof comes from considering circles inscribed in the object boundary. Consider a circle C, inscribed in O, with radius r and center c, which contacts the boundary at a point p. Now think of the point p as a point obstacle, preventing the unconstrained movement of the circle C (see Figure 3.1). We wish to consider the instantaneous motions that are permitted C in light of this constraint. We do this by examining the feasible motions of the center c. Clearly the permissible directions are determined by the direction of the line $\overline{pc}$.

In fact, the permissible motions will be those which tend to increase the distance from c to p in the direction of $\overline{pc}$. These directions constitute an open half plane, whose boundary is $\overline{pc}_c^{\perp}$ (the line defined by direction $\overline{pc}^{\perp}$ which passes through the point c). See Figure 3.1 for an illustration of this.

We next establish the connection between half planes defined in this manner and the notion of a positive spanning set, which we have already seen is intimately tied to the notion of form closure.

Lemma 1 *Let e_1, e_2, e_3 be a set of vectors in $\Re^2$. Then the following conditions are equivalent:*

1. *e_1, e_2, e_3 are positively dependent, i.e. $\nexists\ a_i,\ a_j \in \Re$, s.t. $a_i e_i + a_j e_j = 0, i \neq j$, but $\exists b_1, b_2, b_3 \in \Re^+$ (i.e.positive) s.t. $b_1 e_1 + b_2 e_2 + b_3 e_3 \ = \ 0$*

2. *e_1, e_2, e_3 positively span $\Re^2$.*

3. *There is no half-plane through the origin that leaves all three vectors on the same closed side (i.e. the angles between any two of e_i and e_j is less than π radians).*

The equivalence of 1 and 2 is obvious, and the equivalence of 1 and 3 comes from considering the fact that if any vector can be expressed as a positive linear combination of the e_i, then every such vector must lie in the convex cone generated by two of the e_i. The way in which we shall use this result is the following.

Form closure by a set of unit normal wrenches, $n_i \in N(O)$, is equivalent to the condition that this subset positively spans $\Re^3$. It is a necessary condition that the first two components of the n_i, considered as 2-vectors, positively span $\Re^2$. Denote e_i as the 2-vector associated with n_i in the positive spanning set. What we wish to show, the reason for which we required the lemma above, is the fact that the half planes defined in the above by the contact points of the n_i have an empty intersection. That is, there is no translational direction in which the center, c, is free to move. This will be true whenever the angle between any two of the e_i is less than π radians. The we see that the result is intimately tied to the lemma, above. We present the result formally.

Lemma 2 *Every two dimensional object boundary ∂O contains a maximal inscribed circle and for any maximal inscribed circle in ∂O, we have*

$$\cap_{b \in C \cap \partial O} M(b) = \emptyset \tag{3.2}$$

where $M(b)$ is the half plane of directions that the center c can move despite the existence of a point obstacle at b on $C \cap \partial O$.

PROOF: Existence of a maximal inscribed circle follows from the closure and boundedness of O. To prove the second part of the lemma, assume that there is some vector $\vec{r} \in \cap_{b \in C \cap \partial O} M(b)$. Then. since $\vec{r}$ is an unconstrained direction of motion, we can move the center, c, of our circle a small way along $\vec{r}$, so that it breaks contact with the boundary. This comes from the fact that $M(b)$ is an open half plane, so for any $b \in S \cap \partial O$ there is an open neighborhood ε_a such that movement along $\vec{r}$ increases $dist(c, \epsilon_a) = inf_{a \in \epsilon_a} \parallel b - a \parallel$. Let δ be the smallest of the distances from some point in ∂O to any point in one of these neighborhoods ($\delta = inf_{b \in \partial O}(inf_{a \in \epsilon_a} \parallel b - a \parallel)$). Then we claim that we can move the center a distance of $\delta/2$ along $\vec{r}$ such that it no longer contacts the boundary, and every point in C is, in fact, at least $\delta/2$ away from any point in ∂O. Thus, we could have an inscribed circle with center at $c + \delta/2\vec{r}$, with radius $r + \delta/2$, contradicting our assumption that C was maximal. $\square$

Thus we have demonstrated that unit normal wrenches located at the points of $C \cap \partial O$ form a positive spanning set, via Lemmas 1 and 2. This implies that by applying normal wrenches of sufficient positive intensity at these locations we can oppose arbitrary translational perturbations and enforce translational immmobility on the object with respect to the hand. What remains to be demonstrated is that we can select at most four of the points in $C \cap \partial O$ to the same effect. This will complete the proof of Theorem 1.

We distinguish between two cases in describing our selection method for $G(O)$. Note that $\cap_{b \in C \cap \partial O} M(b) = \emptyset$ does not imply that $\cap_{b \in C \cap \partial O} \overline{M(b)} = \emptyset$. We will first consider the case where the intersection of the $\overline{M(b)}$ is empty.

In this case we can select three points b_1, b_2, b_3 such that the tangents through these points form a triangle whose inscribed circle is C. Then unit normals at the b_i are obviously in $N(O)$. For most, but not all boundaries, these three fingers will provide full form closure. The reason these three are not sufficient for form closure is that each of the normals selected may lie in an open neighborhood of $C \cap \partial O$, i.e. circular arcs which are part of the boundary of O. Figure 3.2a shows such an object. This means that, since we are considering frictionless contacts, the object may be free to rotate. Since we are only considering objects which are not circles, however, we can always add a fourth finger, and position the fingers to prohibit rotation and provide full form closure. We distinguish between the cases where one of the original fingers lies at an isolated point of $C \cap \partial O$, and that where $C \cap \partial O$ contains no isolated points.

Assume, first, that one of the points, say b_1, is an isolated point of $C \cap \partial O$. Then it is always possible to replace the unit normal n_1 at b_1 with a pair of normals n_1' and n_1'', which have the properties that c is in the convex cone generated by n_1' and n_1'' and each of the sets $\{n_1, n_2, n_3\}$, $\{n_1', n_2, n_3\}$ and $\{n_1'', n_2, n_3\}$ is positively dependent. The first property follows from the fact that b_1 was chosen to be a point of isolated tangency, whether at a smooth point of ∂O or at a concave discontinuity (again, it is impossible that the minimum inscribed circle intersect ∂O at a point of convex discontinuity). This implies that $b_1 = \min_{b \in \varepsilon_{b_1}} \| b_1 - c \|$, where ε_{b_1} is sufficiently small that it contains no derivative discontinuities (other than possibly at b_1). Such a neighborhood exists since ∂O is piecewise smooth. The second property follows from our selection of the original three points and from the fact that the set of all vectors in $\Re^2$ that are positively dependent with n_2 and n_3 form a convex open set, so that there exists a neighborhood of directions near n_1 in which all vectors are positively dependent with n_2 and n_3.

Consider a perturbing wrench $n = (f_x, f_y, m)^T$, where we are taking m to be the scalar moment with respect to c, and f_x and f_y are the projections of the force vector on the $\Re^2$ coordinate axes. The equations of equilibrium are:

$$\begin{pmatrix} f'_{1x} & f''_{1x} & f_{2x} & f_{3x} \\ f'_{1y} & f''_{1y} & f_{2y} & f_{3y} \\ m'_1 & m''_1 & m_2 & m_3 \end{pmatrix} \vec{\mathbf{x}} = \begin{pmatrix} f_x \\ f_y \\ m \end{pmatrix} \tag{3.3}$$

Here, m'_1 and m''_1 are positive scalar moments associated with the unit normal wrenches n'_1 and n''_1, respectively, and $\vec{\mathbf{x}} = \left(x'_1, x''_1, x_2, x_3\right)^T$ is the vector of wrench intensities. Since the sets $\{n'_1, n_2, n_3\}$ and $\{n''_1, n_2, n_3\}$ are both positively dependent, so must be the set $\{m''_1 n'_1 + m'_1 n''_1, n_2, n_3\}$. Thus there is a strictly positive solution for $\vec{\mathbf{x}}$, in the above equation. This demonstrates that the four unit normals provide form closure.

Now consider the case that none of $\{b_1, b_2, b_3\}$ is an isolated point. Figure 3.2a shows such an object. Since ∂O is not a circle, there are two or more extreme points in $C \cap \partial O$ (since C and ∂O are closed, there are extremal elements, i.e. points b such that no open neighborhoods, ε_b are wholly contained in the intersection). Since our original three point selection, such as in Figure 3.2b, provided translational form closure, we can replace one of our non-isolated contact points, say, b_1 with a new pair, b'_1 and b''_1. These are chosen to be the endpoints of the arc of $C \cap \partial O$ if they are points of derivative discontinuity, or just beyond the endpoints if these points have smooth neighborhoods. In either case, they are selected such that they leave the center c on either side, i.e. they have counterdirectional moment arms. Figure 3.2 shows the final placement of fingers to provide form closure. That this is possible comes from precisely the same argument that made it feasible for the nonisolated case. Notice that the element selected for replacement should not be arbitrarily chosen, but must instead chosen to lie in a circular arc of less than π radians (in fact it is best if chosen from the arc containing one of the b_i which has minimum arc length which must be less than $2\pi/3r$. The normals associated with these contact points are n_1', n_1'', n_2 and n_3. The subsets $\{n_1', n_2, n_3\}$ and $\{n_1'', n_2, n_3\}$ are each positively dependent, or can be esily made so by appropriate moving of b_2 and b_3 in their arcs (which are, by construction, of greater substended angle than the arc containing b_1 and its replacements). This, these unit normal wrenches form a form closure grasp for O, and can be denoted $G(O)$.

We now turn to the case where $\cap_{b \in C \cap \partial O} \overline{M(b)} \neq \emptyset$. Since we know that $\cap_{b \in C \cap \partial O} M(b)$ *is* empty, the intersection of closed half-planes must yield a single line. In light of our earlier discussion on the constraint of the maximal inscribed circle by its points of contact with ∂O, we see that this line indicates directions that the center of C can move. That this can happen for a maximal inscribed circle is not surprising, and examples which lead to such cases are depicted in Figure 3.3. The constraint of C in one direction and the lack of constraint in the

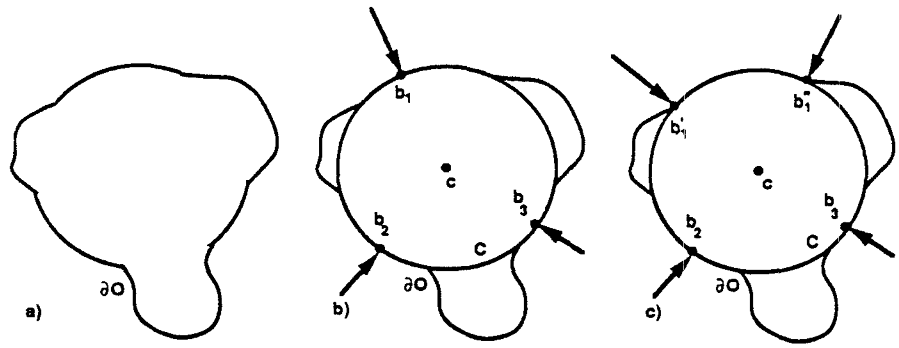


Figure 3.2: Adding a Fourth Finger Provides Rotational Restraint in Frictionless Case

orthogonal direction is a result of antipodal point contacts on C. That is, there are (at least) two point contacts, say b_1 and b_2. It may or may not be the case that there is a third contact point b_3, as is illustrated in figure Figure 3.4. We note that it is possible to find an object such as is depicted in Figure 3.4, with some locations of the maximal inscribed circle contacting two points of ∂O and some contacting three, whereas with some objects, such as the example depicted in Figure 3.5, only two contact points are possible.

We will assume that in the former case, where three contacts exist for a maximal inscribed circle, that we have chosen that circle, and later our algorithm will ensure that we have done so. In this case, we have three inward zero moment wrenches, n_1, n_2, n_3 associated with the contact points. In this case, what is required for form closure is to retain one of the antipodal contact points and

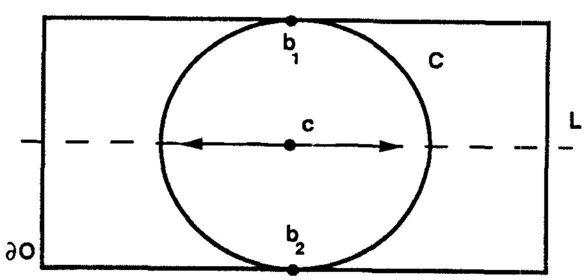


Figure 3.3: Improper Seperation by Antipodal Contacts

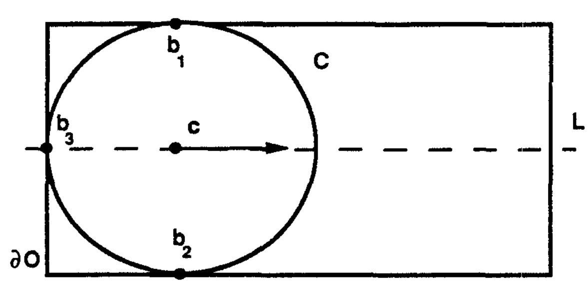


Figure 3.4: Sometimes Three Contacts are Possible

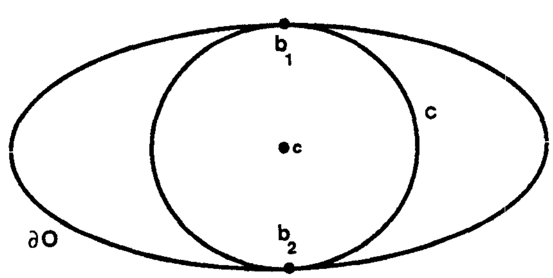


Figure 3.5: Some Boundaries only Permit Two Contacts

its wrench and also retain the n_3 unit normal wrench. To decide which of the antipodal wrenches to retain, we simply search the boundary of O until a contact point is found which is positively dependent with either pair $\{n_1, n_3\}$ or $\{n_2, n_3\}$ and which has nonzero moment with respect to the point c. The alert reader may notice that this is not always possible! If we examine the conditions that lead to our inability to find such a point, b_4 and its associated unit normal wrench, n_4, we arrive with the conclusion that the figure must be a variant of Figure 3.6. That is, all possible unit normal wrenches must either a) not form a positvely dependent set with either pair, $\{n_1, n_3\}$ or $\{n_2, n_3\}$ or b) have a zero moment with respect to the origin. Such is the case when ∂O has two straight, parallel sides, and sides which are circular arcs which have their center's of curvature at c, when C is in contact with ∂O at b_3. This is the case which Markenscoff identifies as his exception to the statement that it is always possible to find a b_4 satisfying the conditions indicated above. Nevertheless, there is a form closure solution even for these objects, as is presented in Figure 3.7.

If we treat these objects as exceptions, we arrive at the conclusion that given an object which has antipodal point contacts at b_1 and b_2, and a third, noncollinear contact b_3, a point b_4 and its associated unit normal wrench n_4 can always be found, if the object does not belong to the class of exceptions we have identified, and for which form closure may easily be found by alternate means. Given that we have found a satisfactory n_4, it remains to be shown how we can complete our form closure grasp. Recall that we will retain b_3 and the one of

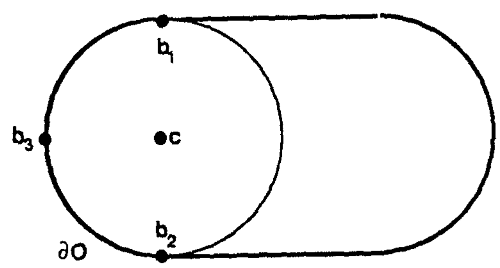


Figure 3.6: No Form Closure Without Friction Possible for Given Contact Locations

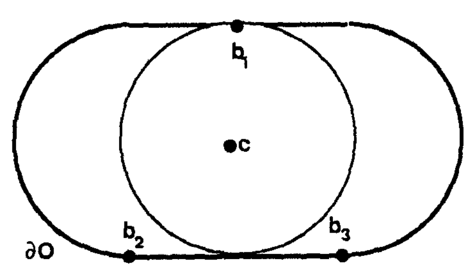


Figure 3.7: Form Closure Exists Even for These Objects

b_1 or b_2 which completes the positive spanning set with b_3 and b_4. In order to complete the grasp, we move the remaining member of the antipodal set either to the left or to the right so that a moment of opposite sign to that of b_4 is created. Assuming, without loss of generality, that we have retained n_1 and moved n_2 to counter the moment of n_4, the equations of equilibrium become:

$$\begin{pmatrix} f_{1x} & f'_{2x} & f_{3x} & f_{4x} \\ f_{1y'} & f'_{2y} & f_{3y} & f_{4y} \\ 0 & m'_2 & 0 & -m_4 \end{pmatrix} \mathbf{x} = \begin{pmatrix} f_x \\ f_y \\ m \end{pmatrix} \tag{3.4}$$

Which always has a positive solution, $\mathbf{x}$, by the same mathematical reasoning as before.

The last case we must deal with is the case where we have two antipodal contacts b_1 and b_2, but it is not possible to find a maximal inscribed circle, C, which has a third contact. An example of such objects is provided in Figure 3.5. In these cases, the strategy to use is to *form closure by couples.* This entails replacing the antipodal pair with two pairs of wrenches which are pure couples, i.e., each pair is such that the force components cancel so that the wrenches summed together form a pure torque. Thus two pairs of wrenches resolve into two opposing moments. Figure 3.8 indicates the difference between nominal and form closure by couples. Since ∂O is piecewise smooth, there will be a neighborhood of each of the contact points in which the normal direction changes continuously. If we move the contacts is opposite directions, e.g. b_1 moves clockwise and b_3 moves counter-clockwise with resepect to ∂O, then the orientation of the unit normal wrenches, n_1 and n_2 will change. They may either change in the same rotational direction or in opposite rotational directions. The rotational direction is determined by the local curvature on each side of each of the original contact points. For example, in Figure 3.5, b_1 is convex left and convex right and b_2 is convex left and convex right. In this case, the clockwise movement of b_1 to b_1' and counter-clockwise movement of b_2 to b_2' give rise to a clockwise of n_1 to n_1' and a counter-clockwise rotation of n_2 to n_2'. Moving b_1 counter-clockwise to b_1'' and b_2 clockwise to b_2'' results in counter-clockwise rotation of n_1 into n_1'' and a clockwise rotation of n_2 into n_2''. Since these rotations occur in a continuous fashion in the neighborhoods of b_1 and b_2, we can split the contacts b_1 and b_2 into the four contacts, b_1', b_1'', b_2' and b_2'', such that the pairs $\{n_1', n_2''\}$ and $\{n_1'', n_2'\}$ each form couples. It is clear that the force components of these four unit normal wrenches form a form closure in $\Re^2$, so any force component of a perturbing wrench $n = (f_x, f_y, m)^T$ can be canceled by appropriate selection of the intensities of n_1', n_1'', n_2' and n_2''. The perturbing torque component, m, plus any additional torque generated by the combination of normal wrenches in the grasp can be then countered by equal increase in the couple which has opposite sign to the net disturbance. Thus, we are able to obtain a combination countering any perturbing wrench, and conclude that we have obtained a form closure by couples. Figure 3.9 depicts the various possibilites and indicates the relative locations of the wrenches in each couple. The method for locating these wrenches precisely will be described with the remainder of the algorithm,

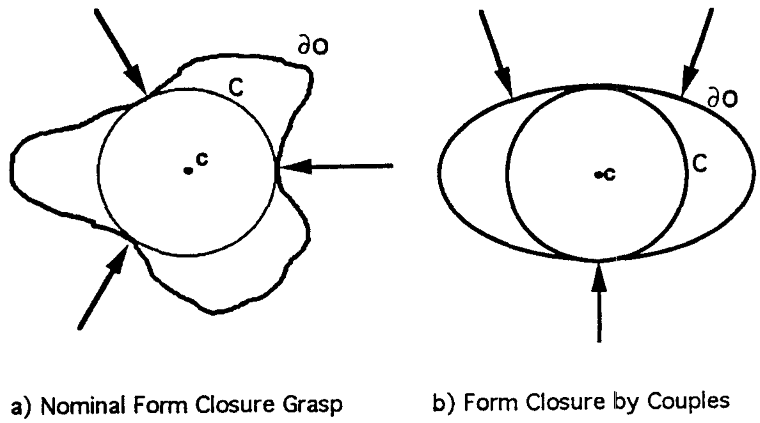


Figure 3.8: Nominal vs Form Closure By Couples

subsequently.

So we have described the method for choosing four unit normals wrenches which can cancel out any perturbing wrench given any general shaped object not one of our exceptions. In all cases, the wrench locations are determined as a result of the locations of the contacts of the maximal inscribed circle and ∂O. Of course, many of the special cases and the use of a fourth unit normal wrench is a direct result of the fact that we have assumed zero friction for all contacts. In the next section, we will see that we can, in all cases, do away with the fourth contact and its associated unit normal wrench. The method for achieving the three (or fewer) contact form closure grasp will heavily rely on the results of the frictionless case.

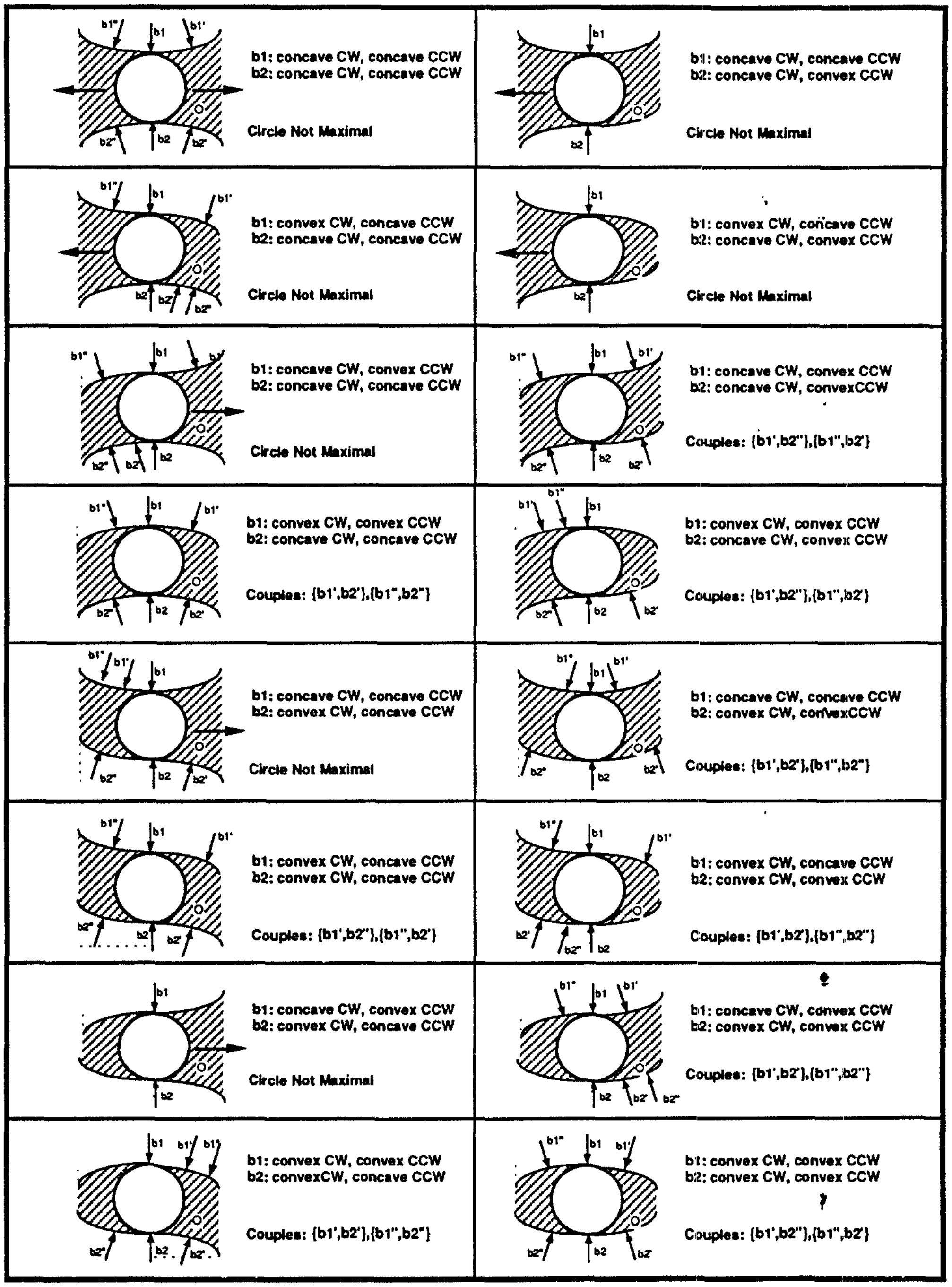


Figure 3.9: Location of Normal Wrenches for Form Closure by Couples

3.2.1 The Effect of Friction on Form Closure

In considering the effect of friction, we will assume a simple Coulomb model, but not assume that we have soft finger contacts, which provide the ability to counter twists about the contact normal. Although the degree to which the MDH fingertips act as soft fingers is dependent on the grasping force being exerted, the soft finger model will usually be appropriate. The existence of resistance to twists about the contact normal can only assist in establishing a firm grasp, however, so grasps assumed stable using the point contact with friction model will surely lead to firm grasps if the fingers are, in reality, soft.

To discuss the grasp by point contacts with friction, we expand on the notion of unit normal wrenches and describe contact forces via their *fingers.* A finger is a pair, $(n_i, \mu n_i^{\perp})$, where n_i is the unit normal wrench at b_i, with force components f_{i_x} and f_{i_y} and moment as above, and $\mu n_i^{\perp}$ is a vector tangent to ∂O at b_i which represents the maximal wrench perpendicular to n_i which may be resisted by the tangental frictional wrench generated by a unit normal wrench n_i at b_i. Here, μ is issumed to be positive and represents the coefficient of friction between the fingertip and the grasped object, which is dependent on the material of the grasped object (as well as on its local geometry, actually, but we shall not add this complication, again erring on the conservative side with this simplification). Initially, we will neglect the effect of finite coefficient of friction altogether. We can show some powerful results about form closure under point contacts with friction which do not depend on the coefficient of friction [2]. This is due to our earlier assumption that arbitrarily high normal grasping forces may be generated. As long as μ is positive, we can generate arbitrarily high forces in the directions tangential to contact normals simply by increasing the intensity of the contact normal wrenches. Of course, we shall have to deal with the fact that we can only generate limited contact forces subsequently, but we will defer this consideration until we have adequately developed the notion of form closure grasp with friction. The result of including the consideration of limited contact force will simply be to limit in magnitude the perturbing wrenches that can be countered. Another consideration that is obliterated by the assumption of equivalency between infinite and finite coefficient of friction is the effect of high contact forces on the object. If, in order to maintain a high tangential force, we must provide high contact force, we develop high internal forces on the object which may damage it. Again, for the sake of this development, we will ignore this complication.

The main result used in this section is:

Theorem 2 *(Markenscoff, Li and Papadimitriou) Any two dimensional object, O, has a form closure under friction with three fingers.*

[2] A proof that form closure by point contacts with infinite coefficient of friction is equivalent to form closure with any finite, positive coeffient is presented in [1], but relies essentially on the notion that any finite tangential force at a contact can be countered by sufficiently high normal forces and finite coefficient of friction.

PROOF: Consider again the maximal inscribed circle, C. Recall that in our consideration of form closure without friction we distinguished between the case of (only) two contact points (when $\cap_{b \in C \cap \partial O} \overline{M(b)} \neq \emptyset$) and that of three contact points such that the contact normals were positively dependent. In the former case, we claim that the two contact normals, n_1 and n_2 are a form closure under friction. Assume, without loss of generality, that the contacts are in the x-direction ($n_1 = (1,0,0)^T$ and $n_2 = (-1,0,0))^T$. Then the friction induced wrenches will be those orthogonal to the contact normals, yielding a grasp matrix of wrenches:

$$\begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 1 \\ 0 & 1 & -1 & 0 & 1 & -1 \end{pmatrix} \tag{3.5}$$

It is easy to verify that these six columns positively span $\Re^3$, and therefore constitute a form closure grasp. In the latter case, where we have three contact normals which are positively dependent, we have already achieved translational restraint of the object via these contacts. Thus, all that needs to be established is that we have rotational restraint. Consider, without loss of generality, a counterclockwise perturbing moment acting on O. Due to friction, there are three wrenches perpendicular to n_1, n_2 and n_3 and oriented in the clockwise direction with respect to ∂O. These vectors will also form a positive spanning set, since each is formed by rotating the members of a positive spanning set, which property is invariant under rotation. However, the positive combination of the tangential set does generate a clockwise moment with respect to the center, c, of the maximal inscribed circle. Thus, any counterclockwise moment with respect to this point can be countered by a positive combination of the tangential wrenches. Since we have assumed infinite coefficient of friction, any positive normal force will cause the tangential set to be sufficient to counter any counterclockwise perturbing moment (the same result holds for clockwise perturbing moments using the counterclockwise oriented tangential wrenches).

□

Since we had previously demonstrated translational restraint, and have now demonstrated rotational restraint, the claim of form closure under friction by three contacts with friction is proven. It is important to note that the contacts selected under considerations of friction are the same as those selected in the frictionless case. We simply do not require an extra contact to insure rotational restraint in each rotational direction.

3.3 A Maximum Inscribed Circle Algorithm for Grasp Selection

Using the notions described in this chapter, we present an algorithm which generates the maximal inscribed circle for a general object boundary. Once such a circle is obtained, we may select grasp points simply by determining the points of intersection of the circle and the object boundary. The portion that we have implemented, and present here generates the maximal inscribed circle for all convex polygons, as well as for those nonconvex polygons which have the property that the center of the maximal inscribed circle lies in the same maximal convex subpolygon as the centroid of the coordinates. This is the challenging portion of the problem, and the solution we present is reliable and robust. We will mention, later, what is required to make this algorithm work for a general polygon. Additionally, we have not indicated how one goes about obtaining a good grasp plane for a three-dimensional object. While the selection of the appropriate grasp plane, or at least its orientation, is often prescribed by the task for which the object is being grasped, the general capability to heuristically select a grasp plane would be useful. We shall also make some comments about how a grasp plane could be selected using the maximal inscribed circle algorithm to rate grasps, and then selecting from the planes examined.

We rely heavily on the work of Melville [18] in the development of this algorithm. Melville presented the solution to what is essentially the dual of the maximal inscribed circle problem, i.e. the minimal spanning circle problem. Although our algorithm by necessity differs substantially from Melville's, the spirit of the approach is the same, and the inheritance from his work is evident.

The basic feature of the maximal inscribed circle (hereafter MIC) is that it contacts the boundary in at least two, and more generally at least three points. In the former case, we can typically move the MIC so that there are at least three contacts (except, for example in an ellipse with different major and minor axes). The contacts in either case satisfy the property that no line which passes through the center of the MIC leaves all contacts in one of the two open half-planes into which the line divides $\Re^2$. In the case of two contacts, they may be contained in a closed half plane, in which case, we say that the MIC is *improperly separated* by the line into a plane containing the contacts and a plane containing no contacts. If there is no line which separates the contacts into one half plane, the circle is said to be *inseparable.* If there is a line passing through the center of the circle that does *properly separate* the contacts into an open half-plane, then the circle is not maximal, as demonstrated in Lemma 2.

This suggests a general method for finding the MIC. If we start with an inscribed circle, we can grow it and continue checking the contacts until we either have an improperly separable set of contacts or we have an inseparable

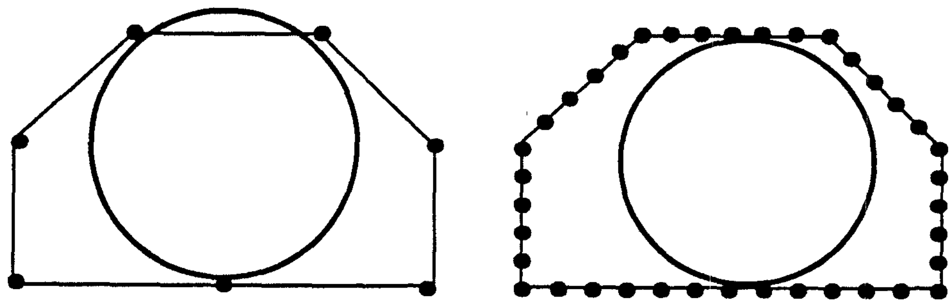


Figure 3.10: Decreasing Point Seperation Decreases MIC Error

set of contacts. This is the basis for our method, although, of necessity it is a bit more involved.

We are given a set of points which approximate the object boundary. It is evident that the error in the radius of the MIC will depend heavily on the separation between points. Figure 3.10 depicts how the separation of boundary points affects the size of the MIC. The first step in our algorithm is to add interim points so that the maximum separation between points is such that the error due to the pointwise approximation of the object boundary is small. Given a collection of points in $\Re^2$, it is easy to determine the minimum and maximum x and y values that are attained. These four values determine rectangle which bounds our polygon. Clearly, the MIC has a diameter which is less than or equal

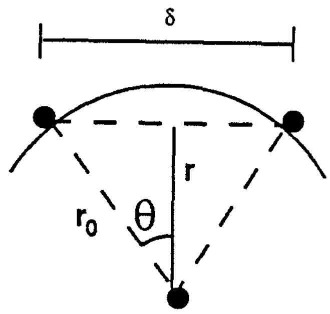


Figure 3.11: Calculation of MIC Error as a Function of Point Seperation

to the smaller of $x_{max} - x_{min}$ and $y_{max} - y_{min}$. Examining Figure 3.11 we see that the angle θ is give both by:

$$\theta = \cos^{-1}(r/r_0) \tag{3.6}$$

and by:

$$\theta = \sin^{-1}\left(\frac{\delta/2}{r_0}\right) \tag{3.7}$$

Solving these for r/r_0, we get:

$$r/r_0 = \cos\left(\sin^{-1}\left(\frac{\delta/2}{r_0}\right)\right) \tag{3.8}$$

Then we may make r/r_0 small by making δ/r_0 small. Since we have bounded r_0 by $r_{0,max} = \min(x_{max} - x_{min}, y_{max} - y_{min})$, we can select $\delta = 0.05r_{0,max}$, yielding a value of $r/r_0 \approx 0.9997$, which is quite acceptable. Thus the first step in our algorithm is to traverse the boundary points, inserting points as necessary to achieve this separation.

Given that we have limited placement error due to boundary approximation, we begin our algorithm in earnest. We must first choose an inscribed circle. As a first guess, we choose a circle whose center is located at the coordinate centroid, i.e.

$$c_x = \sum_{i=0}^{n} p_{i,x} c_y = \sum_{i=0}^{n} p_{i,x} \tag{3.9}$$

where n is the number of points in our boundary and $p_{i,x}$ and $p_{i,y}$ are the boundary point coordinates. We then calculate the radius of our first inscribed circle as:

$$r_c = \min_{i=1,n}(||p_i - c||) \tag{3.10}$$

This yields a circle which is interior to the boundary and inscribed by the boundary points for all convex polygons. We next determine the number of contact points. We know we have at least one (the minimum distance point calculated above), but we may have more. If not, our next step is to get a second contact point. Figuratively, we do this by moving the center of the circle along the direction given by $c - p_i$ away from p_i and enlarging the radius of the circle so that contact is maintained, and stop when the next contact is made. In actuality, we solve this problem analytically by finding a point, p_j such that the location of the intersection of the perpendicular bisector of (p_i, p_j) and the line connecting c and p_i is equidistant from p_i and p_j. We find this point by recasting the coordinates in a coordinate frame located at the center of (p_i, p_j) and with its $x - axis$ directed towards p_j. This makes the calculation much simpler, since we need only solve for the new y-coordinate (the x-coordinate being zero). The y-coordinate of such a point is given by:

$$y'_{c_i} = \frac{x'^2 + y'^2}{2y'} \tag{3.11}$$

where x' and y' are the coordinates of the test point in the new coordinate frame. We search through all such points and find the one which is the smallest distance from p_i. This will be our minimum (at least) two contact inscribed circle.

We perform a check to see that the original circle is not maximal. This involves sorting the contact point(s) by angle with respect to the circle center and then checking the separation between consecutive contact points. If the circle is maximal, there should be no consecutive pair of contacts whose angular separation exceeds π radians. If the circle is not maximal, we begin a repetitive process of growing the circle until a new contact is made and then checking for maximality. If a circle is not maximal, then we may order the contacts such that the outermost pair is separated by less than π radians. At each step, we grow the circle by moving the cener away from the midpoint of the segment joining the outermost two contacts, arbitrarily labelled p_0 and p_1, along the perpendicular bisector of the segment. The circle expansion is accomplished in a similar manner to the way in which the second contact point is obtained. This time, however, we are looking for points, p_i, which have the property that the point at which the line connecting c to p_i intersects the perpendicular bisector of p_0 and p_1 such that the distance from the intersection to p_i is the same as the distance from the intersection to either p_0 or p_1. Again, recasting the coordinates in a frame at the midpoint of (p_1, p_0), with $x-axis$ directed towards p_1, we find that the x-coordinate of such a point is 0, and the y-coodinate is given by:

$$y'_{c_i} = \frac{\left(\left({p'_{0,x}}^2 + {p'_{0,y}}^2\right) - \left({p'_{i,x}}^2 + {p'_{i,y}}^2\right)\right)}{2\left(p'_{0,x} - p'_{i,x}\right)} \tag{3.12}$$

After each expansion, we count the contact points, and determine their separability. Eventually, we arive at a stage when we can no longer grow the circle, and we have either found improperly separable contacts, or found an inseparable set of contacts. We use these for our grasp.

This algorithm has been implemented in the program "graspsel", includes an X-window interface to allow general interactive exploration of MIC grasps. Figure 3.12 shows a screen dump of the tool. It allows the user to interactively draw a polygon or to import one from a file. He may then run the maximal inscribed circle routine and have the MIC overlayed on the polygon. A further click of a button allows the contact points to be plotted. Figure 3.13 shows a number of interesting polygons for which grasps have been found.

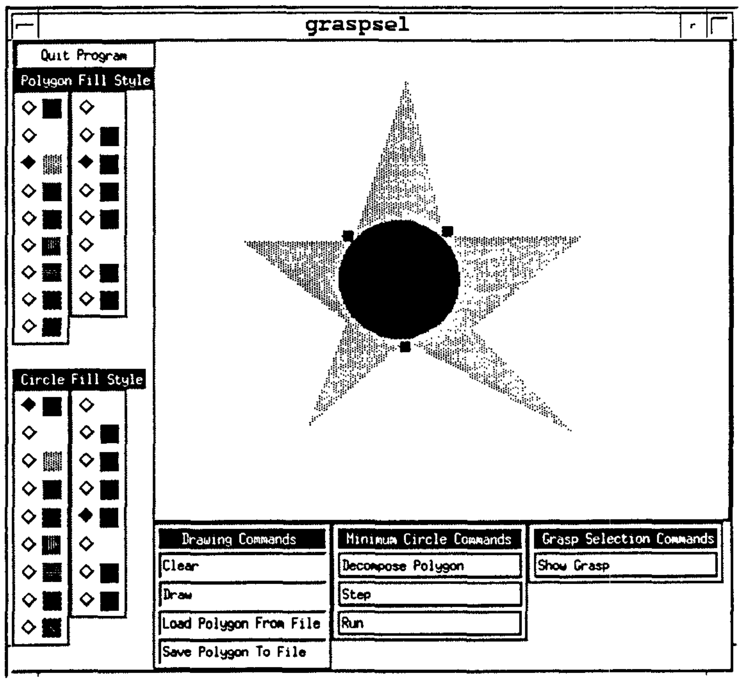


Figure 3.12: A Screen Dump of the Grasp Selection Tool "graspsel"

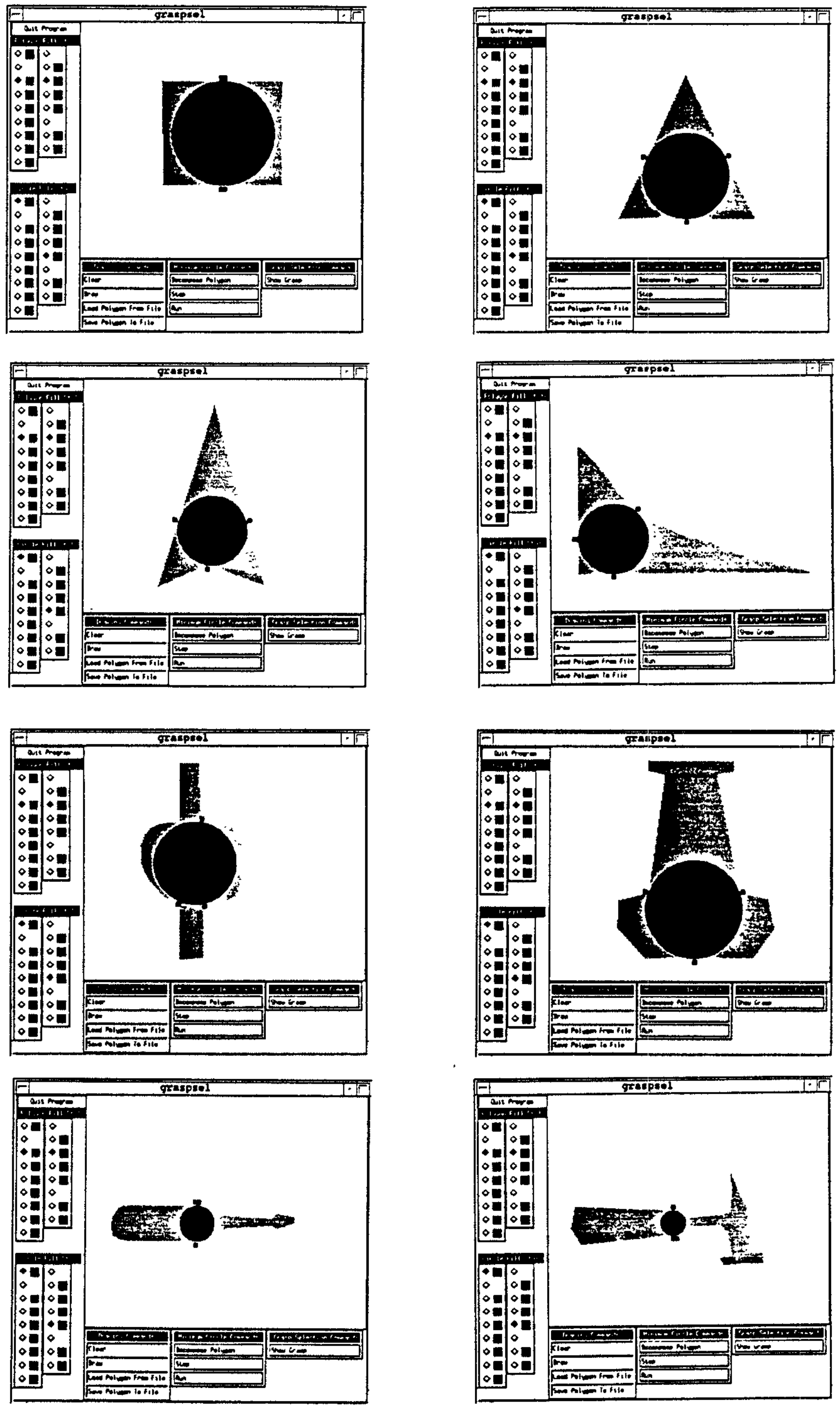


Figure 3.13: Several Examples of Grasps Selected by "graspsel"

3.3.1 Generalized Polygons

As it exists, the graspsel program is only useful for generating grasps for convex polygons, and those nonconvex polygons which have the property that the center of their MIC lies in the same maximal convex subpolygon as the coordinate centroid. This is the case with a large number of typical shapes. Additionally, grasps are generated for other more general polygons, but are not guaranteed to be globally maximal. In order to make the method useful for truly arbitrary polygons, we need to implement a routine which decomposes an arbitrary polygon into a collection of convex subpolygons, such as that proposed in [19]. The method would be to calculate centroids for each of the convex subpolygons and start the basic MIC algorithm with the original inscribed circle generated from the boundary points in that subpolygon. As the circles grow, they are allowed to enter other polygons as necessary, i.e. the boundary points for segments connecting atomic convex polygons are not augmented to constrain the growing circle. So each circle would then be locally maximal. The task, then, would be to scan the list of locally maximal circles to find one which is best suited for the task at hand. Selection criteria might include finding the circle with the globally maximum radius, or perhaps the circle whose center is closest to the boundary centroid (and possibly to the object center of mass to minimize disturbing torques). In any case, the enhancement of the basic MIC algorithm to accomodate arbitrary polygons requires only the implementation of the polygon decomposition routine and the implementation of a routine which scans the circles thus created to find the one which is best for the given task. This is not a difficult enhancement to make, and we hope that future work will take up this task.

3.3.2 Grasp Plane Selection

The MIC grasp selection algorithm assumes that a two dimensional boundary exists. This implies that the plane in which the grasp is to be found has been determined for the object. While the grasp plane is often dictated by the possible approach directions of the hand, or by the task for which the grasp is to be obtained, it would be desirable to be able to locate grasp planes for an arbitrary three-dimensional object (specified by a collection of points from a sampling of the object surface). The MIC algorithm we present could be used in this capacity. For the purpose of maximal stability of the grasp for wrenches of arbitrary direction, the best grasp has the surface normals at the contact points lying within the grasp plane. This indicates that the object is locally prismatic near the grasp points. The MIC algorithm could be successively applied until, say, three successive boundaries yield MICs which have nearly the same radii and which yield contact points in approximately the same (2-D) locations. The central plane, then, would be a good candidate for the grasp plane (and the grasp would be already determined). This would be an interesting area for further research, and we hope that it will be pursued by future researchers in this field.

Chapter 4

Grasp Characterization

We mentioned earlier that the grasps selected using the methods of Chapter 3 are, in some sense, "good". In this chapter, we set upon the task of quantifying what we mean by good. We know already that we can constrain an object to two dimensional (three degree-of-freedom) immobility by selecting a two dimensional form closure grasp, and assuming only point contacts with friction and grasping forces of arbitrarily high magnitude and of arbitrary direction.

The purpose of this chapter will be to analyze the two dimensional stability of our form closure grasp. We will also try to generalize the analysis of grasp to see how it pertains to general three dimensional objects. We had heretofore assumed that the grasping forces could be applied at the point specified by the MIC algorithm in a direction normal to the tangent plane at the contact point. This assumption is convenient for an analysis of the utility of a form closure grasp, but must be re-evaluated in light of the fact that, for the MDH, force application directions are strongly dependent on the locations of the grasp points relative to a fixed coordinate frame attached to the hand. Although we will attempt, through our prescription of the contact angle in the inverse kinematics section (Chapter 5), to ensure that contact forces are close to normal, we must address the effects of non-normal forces on grasp stability. We had also assumed that the forces which the (generic) hand could apply were of arbitrary magnitude. In our effort to determine grasp stability for the MDH, however, we must take into account that maximum applied grasp force is limited by the maximum torque generated by the finger actuator motors, τ_{max}.

The effects of removing the assumptions of contact force normality and arbitrary force magnitude will, in general, be to provide limits to the range of perturbing wrenches which may be withstood by adjusting the grasping forces. The description of the entire range of wrenches which can be withstood by a specific grasp of a specific object is one measure of the stability of the grasp. For our purposes, we will attempt to answer the question of whether the hand can generate sufficient forces to withstand a perturbing wrench of known magnitude and direction. Specifically we shall rely on the following definition:

Definition 1 *A Grasp, $G(O)$, and the configuration variables α_i and θ_i associated with it constitute a* stable (or invariant) grasp with respect to the wrench, *w_p, if the finger contacts are not broken by the application of w_p.*

Of course this is a limited definition of stability, as compared to the definition provided in [6], which equates stability with the state of being at a local minimum of the potential function at the equilibrium position. The fact that our grasps are stable in this sense is proven in that reference, and we shall not repeat the proof here. What we provide by *our* definition, however, is a measure which we can use to evaluate a grasp on-line to determine whether a grasp can withstand task-determined perturbations. We are not concerned about whether our grasp can withstand *some* wrench of arbitrary direction and (arbitrarily small) magnitude, but rather of whether our grasp can withstand *specific* wrenches that the object is likely to encounter in the accomplishment of a specified task. Ideally we would map out the entire space of wrenches that could be withstood and then simply check to see if the expected perturbations fall in this range. However, such a computation would be expensive, and only of value for a specific grasp of a specific object. That being the case, it seems a better idea for an on-line analysis tool to be able to answer the question of whether a grasp is invariant with respect to a specific wrench.

We will quantify these concepts in the context of a two dimensional grasp of the object boundary defined by the intersection of the three dimensional object surface and the grasp plane. Of course, any object to be grasped will inhabit all three spatial dimensions, so we will use our knowledge of two dimensional stability to make inferences about the three dimensional case. We will examine three dimensional stability by permitting ourselves the generous assumption that the object is prismatic in the vicinity of the grasp points, that is, that the surface normals at the grasp points are all contained in the grasp plane. We can then identify the factors which need to be considered for an analysis of the most general three dimensional case.

In the final wash, the task which is to be carried out will be the final arbiter of the "goodness" of a grasp. If we can carry out a specified task without losing the grasp, and allowing for acceptable margins of error in estimates of grasp forces and perturbing forces, then the grasp is good enough. Since the task is known, we can determine the types of perturbing wrenches we are likely to encounter. It is expected that we are lifting the object and so the object's weight, acting through its center of mass is one such perturbation. Additionally, we will be accelerating the object as we move it, thus inducing a dynamic loading which must be compensated. Finally, there are likely to be perturbations which arise from the nature of the task for which we have grasped the object. Prehaps the task is to hold the object against a surface with a known force. Or perhaps we expect to encounter some disturbing torques as we insert one object into another. As long as we can make some meaningful statements about the types of disturbances likely to be encountered, we can use our analysis to see if we are likely to have a grasp that is "good" under those conditions.

This information can be used in grasp synthesis, to provide feedback before a task is undertaken. A simplified schematic of the grasp process is shown in Figure 4.1 illustrates the discussion here. As indicated there, we first try to select a grasp we believe to have good stability properties, based on our notions of form closure grasp. This requires intelligent selection of the grasp plane, and selection of a two dimensional form closure grasp. We can then analyze the stability of the grasp with respect to task-determined perturbation invariance requirements. If the task can be safely accomplished, we may proceed. Otherwise, we seek to modify the grasp, first by performing marginal modifications to the grasp positions in the original grasp plane, and if we are yet unsuccessful, we try a new grasp plane altogether. This process is iterated until a grasp which exceeds our task requirements is found.

In the end, it will be our analysis of grasp stability that allows us to predict whether a task may be safely accomplished. The purpose of this chapter is to show the analytical tools by which this prediction may be obtained.

4.1 Two Dimensional Grasp Stability

Due to the physical nature of the MDH, some simplifications on the general notion of grasp stability are possible. Since the MDH has only three fingers, we are restricted to grasps that consist of at most three contact points, which must, of course, be coplanar. Whatever grasp is selected, the primary contacts with the object will lie in a plane, called the grasp plane. Although the full three dimensional geometry nearby the grasp points will be important to general

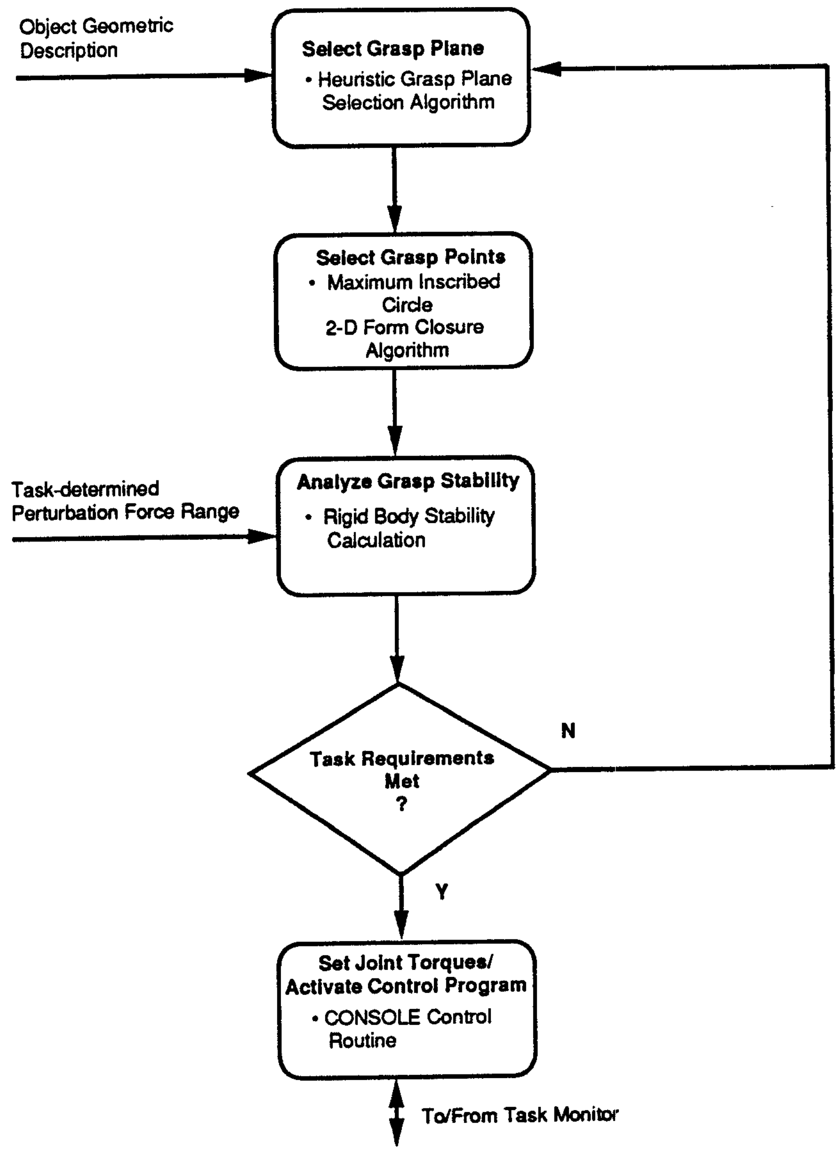


Figure 4.1: Iterative Flow of the Grasp Selection/Analysis Cycle

stability, as will the fact that perturbing forces and moments are inhabitants of three-space, we may obtain significant first order stability information by considering only the grasp of the two dimensional boundary of the object formed by the intersection of the object surface and the grasp plane. This section will present an analysis of two dimensional grasp stability that will be relied upon subsequently to make some statements about the general three dimensional stability question.

4.1.1 Grasping with Directionally Constrained Fingers

4.1.1.1 Frictional Effects of Non-normal Forces

Up to now, we have assumed that we can always place the fingers of our hand in a configuration which allows the grasp wrenches to be applied in the plane of the three contacts, and in a direction normal to the tangent at the contact point. While the physical configuration of the MDH requires the grasp forces to lie in the plane of the contacts, it is not always the case that we can require the direction of force to be normal to the tangent at the contact point. In Chapter 5, we shall examine the issue of how to determine the finger angles for the MDH which produce a given grasp triangle, but for now, we can assume that a particular set of grasp locations corresponds to some set of finger angles, $\{\theta_1, \theta_2, \theta_3\}$. For the i^{th} finger, commanded to a position θ_i, we would ideally have the contact point at $\alpha_i = \pm\pi/2$, as indicated in Figure 4.2. In this case, the direction of application of force and the surface normal will be aligned, assuming we have chosen the contact point using the methods we have described. However, a given grasp triangle is only achievable at a discrete number of configurations of the MDH [20], and so we cannot prescribe both the contact positons (and therefore the finger angles) and the angular position of the contact with respect to the finger axis. In general, then, we will have a contact angle for the i^{th} finger of α_i, as indicated in Figure 4.2. We see that there is an unintended wrench in the direction of the surface tangent at the contact point. There are two issues to consider. First, will the tangential component of the contact force cause the grasp to be broken? Second, how does the addition of these unwanted tangential components at each of the grasp points affect our form closure grasp?

The question of whether the contact is broken due to non-normal force application is easily answered by resorting to the notion of friction cones. This geometric interpretation of Coulomb's law was apparently first constructed by Moseley [21].

Consider a force, f, directed at a point on the object surface at an angle α with respect to the surface normal at the contact point. We may decompose f into its component forces normal and tangential to the surface, f_n and f_t, respectively. Clearly, f_n and f_t follow the relation, $\tan^{-1} \alpha = f_t / f_n$. We know contact will be maintained if the tangential component of the force does not exceed the product of the normal component of the force and the static coefficient

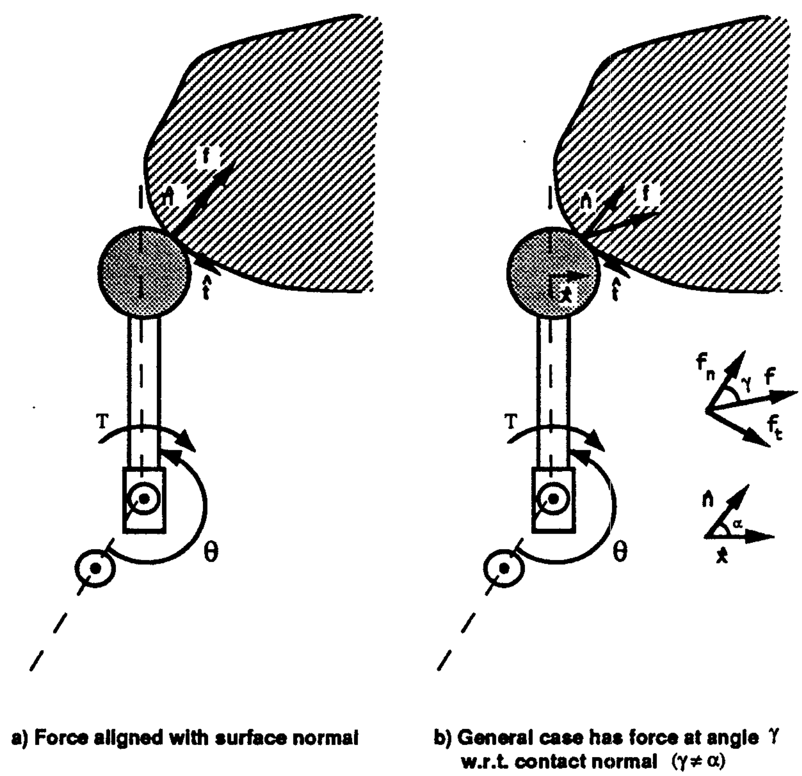


Figure 4.2: Grasp Forces May Not Always be Aligned with the Surface Normal

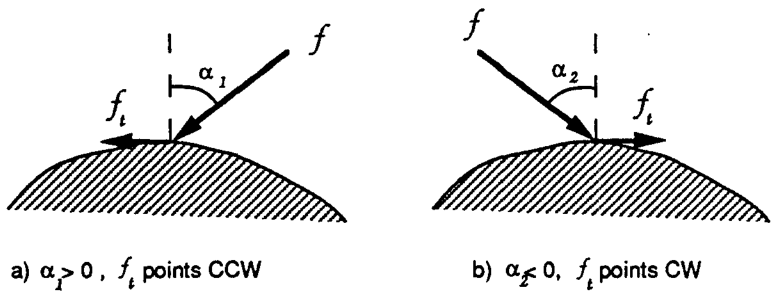


Figure 4.3: The Sign of the Contact Angle, α

of friction, namely $f_t \leq \mu_s f_n$. Substituting, we get the geometrical notion that contact will be maintained so long as $\alpha \leq \tan^{-1}\mu_s$. See Figure 4.4 for an illustration of the notion of a friction cone. Of course, our forces are constrained to lie in the grasp plane, so we only need consider a two dimensional cone. Accordingly, we will use the convention that α is of positive sign when f_t points in the counter-clockwise direction with respect to ∂O and α is of negative sign when f_t points clockwise, as indicated in Figure 4.3. As an indicator of the permissible angles, consider that a coefficient of static friction $\mu_s = 1/2$ gives rise to a friction cone of roughly 26.5 degree half angle.

4.1.2 Effect of Non-normal Forces on Grasp Stability

Answering the question of the effect of non-normal forces on what we had determined to be an equilibrium form closure grasp is not much more difficult. Considering that form closure grasp is equivalent to both force closure and torque

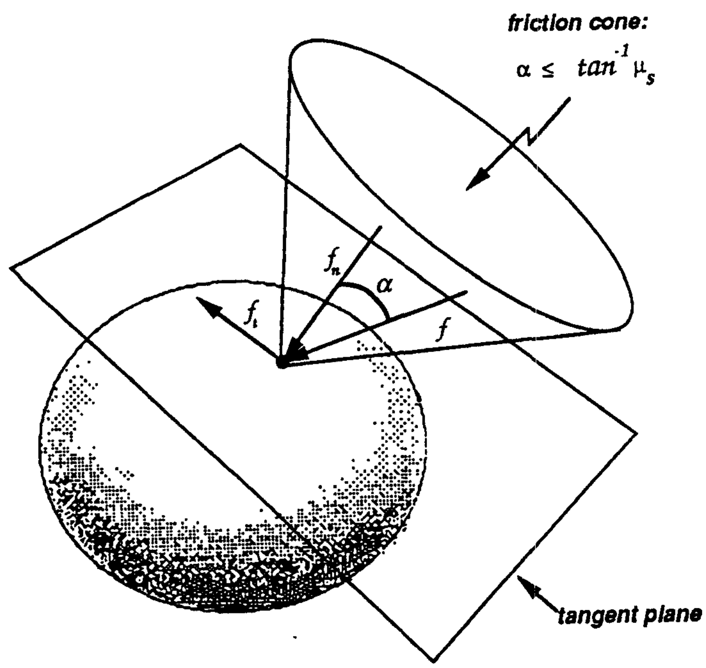


Figure 4.4: The Friction Cone – A Geometrical Interpretation of Coulomb's Law

closure, we must only ask whether the combination of tangential components of the grasp exceed the wrenches that can be withstood by the grasp. We turn to a quantitative answer to this question in the next section, but that discussion will be undertaken with the following qualitative aspects in mind.

First of all, we consider two distinct grasp types. The first comes from our nominal case, where $\cap_{b \in C \cap \partial O} \overline{M(b)} = \emptyset$ (see Section 3.2). In this case, we have three distinct grasp points at the points of contact of the maximal inscribed circle, C, and the boundary of the object, ∂O. In this case, the nominal wrenches for form closure grasp are directed radially towards c, the center of C. Thus they are pure force wrenches with respect to c. In the second case, we have a three finger contact on an object which satisfies $\cap_{b \in C \cap \partial O} \overline{M(b)} \neq \emptyset$, which we have shown indicates only two points exist in the intersection of C and ∂O. We have shown that in this case, form closure by couples may be used to obtain form closure on O. In the case of three contacts with friction, this is accomplished by locating one finger at a point in $C \cap \partial O$ and locating the remaining fingers in such a way that they provide moments of opposite sign, with respect to c. Figure 3.8 illustrates these two basic grasp types.

In the first, nominal case, the addition of tangential components of the grasp forces will not effect the equilibrium as long as the grasp forces lie within the friction cone. In essence, the tangential components of force arising from the non-normal direction of the grasping forces will be cancelled by the tangential component of force arising from the frictional component of the grasping force. This is clear from the geometry of the situation. For the case wherein we rely on form closure by couples, however, we must require that the actual directions that we are able to achieve with our grasp do not cause fingers whose moments are supposed to oppose and cancel each other to, in actuality, have the same sign. As long as this is avoided, we may conclude that the equilibrium form closure grasp will be maintained. It is easy to see in a qualitative way, however, that the magnitude of the disturbing torques which may be countered for a given normal grasping force will gain an asymmetry in the non-normal grasping case. It is as though the addition of tangential components of torque caused by non-normality of grasping wrenches act as an external disturbing wrench. Thus, when in the next section, we describe the loads which are withstandable by grasps wherein the applied force is limited, we must take into account the hidden disturbing wrenches contributed by the non-normality of the grasp wrenches.

4.1.3 Limited Magnitude Grasping Forces and Stability

We now turn to an examination of the practical effects of grasping forces which are limited in magnitude. We shall first examine the effects of limited forces on the two dimensional stabilty of the grasp. Specifically, what translational and rotational disturbances can be countered by three fingers with frictional contacts whose lines of action may not be normal to the surface of the minimal inscribed circle, and whose magnitudes are limited?

Before embarking on this analysis, we will need some machinery. Specifically, we need a mapping which gives contact point position as a function of joint angle, and one which yields force at the i^{th} fingertip contact point as a function of the i^{th} joint torque. The first mapping, one for each finger, is defined by the kinematic homogeneous transformation, $T_0^{C_i}$, where frame $\sum_0$ is the reference frame attached to the center of the palmar plate, and frame $\sum_{C_i}$ is the frame at the contact point of the i^{th} finger. We can get information about the torques exprienced by the joints for given forces on the fingertip by examining the finger Jacobians, J_i, again, with one map per finger. However, the information we need is really the inverse mapping, yielding the torques required to generate specific grasping forces. We briefly review the calculation of these maps for the specific case of the MDH in the next three sections.

4.1.3.1 2-D Kinematics

The general subject of how to derive forward kinematic mappings for open chain mechanical systems is well studied and documented, for example in [22] and [23]. What we will document here is the calculation of these mappings for the fingers of the MDH. We follow the standard technique of assigning Denavit-Hartenburg parameters to the links and joints of each finger and computing $T_0^{C_i}$ as a product of the successive joint transformations $T_0^{B_i}$, $T_{B_i}^{T_i}$ and $T_{T_i}^{C_i}$ where the frames $\sum_{B_i}$,

$\sum_{T_i}$ and $\sum_{C_i}$ are the local frames assigned to the i^{th} finger base, the i^{th} finger tip center and the i^{th} contact point on the surface of the finger tip spheroid, repectively.

We treat the fingers in a homogenous way, by noticing that the fingers are identical, except for the displacement to and orientation of the base frame of the finger. We therefore develop the kinematics of a general finger, and allow the parameters $\vec{b_i}$ and β_i to represent the displacement to and orientation (in the $x-y$ plane) of the i^{th} finger base coordinate frame, respectively. See Figure 4.5 for an illustration of all of the kinematic link and joint parameters.

1. $T_0^{F_i}$

   $T_0^{F_i}$ is the transformation which gives us the vector in the base frame, $\sum_0$, located in the center of the palmar plate, given a vector expressed in $\sum_{B_i}$, a coordinate frame located with it's z-axis aligned with the rotation axes of the i^{th} finger, x-axis aligned with the finger tube, and y-axis aligned in such a way as to form a right-handed coordinate system. Given $\vec{b_i} = (b_{ix}, b_{iy}, 0)^T$ is the location of the origin of $\sum_{B_i}$, β_i is the angle, measured positive counter-clockwise from the x-axis of $\sum_0$, between the x-axis of $\sum_0$ and the x-axis of $\sum_{B_i}$ when in the reference position ($\theta_i = 0$), and θ_i is the rotation,

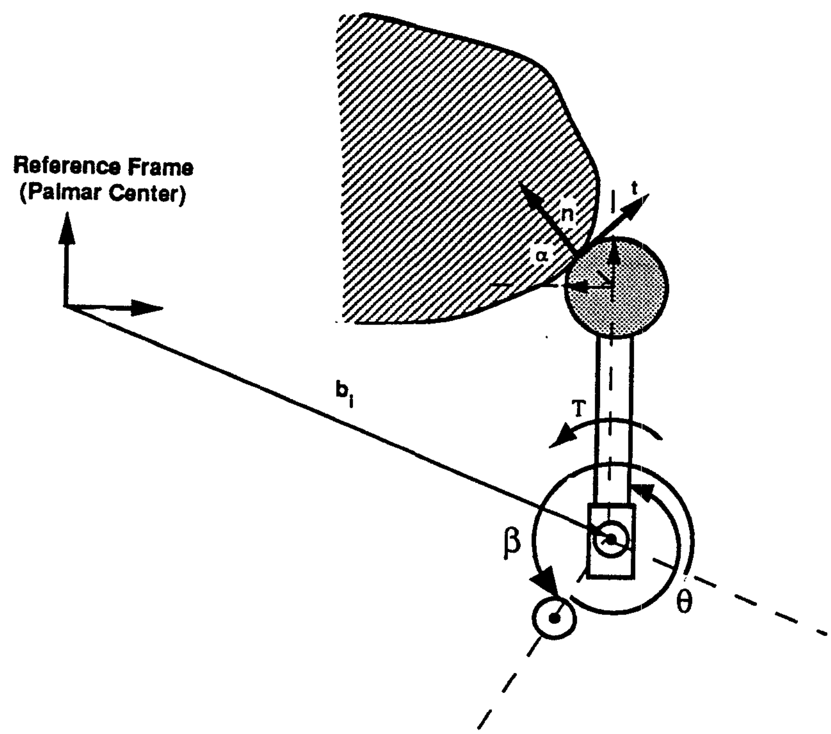


Figure 4.5: Illustration of MDH Link and Joint Parameters

measured positive counter-clockwise from the reference configuration x-axis of β_i, we have $T_0^{B_i}$ is given by:

$$T_0^{B_i} = \begin{bmatrix} \cos(\beta_i + \theta_i) & -\sin(\beta_i + \theta_i) & 0 & b_{ix} \\ \sin(\beta_i + \theta_i) & \cos(\beta_i + \theta_i) & 0 & b_{iy} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

2. $T_{B_i}^{T_i}$

$T_{B_i}^{T_i}$ is the transformation which gives the coordinates in the F_i frame, given a vector expressed in the T_i frame. The displacement vector from the finger base to the fingertip center is given by $\vec{t_i} = (l_f, 0, h_f)^T$, where l_f is the length of the finger tube and h_f is the height (in the z-direction) of the finger tube [1]. The final parameter of this transformation is the angle, α_i which is measured positive, counter-clockwise from the y-axis of the $\sum_{B_i}$ frame (i.e. the total rotation from frame $\sum_{B_i}$ to frame $\sum_{T_i}$ is $90^\circ + \alpha_i$). Thus, $T_{B_i}^{T_i}$ is given by:

$$T_{B_i}^{T_i} = \begin{bmatrix} \cos(90^\circ + \alpha_i) & -\sin(90^\circ + \alpha_i) & 0 & l_f \\ \sin(90^\circ + \alpha_i) & \cos(90^\circ + \alpha_i) & 0 & 0 \\ 0 & 0 & 1 & h_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

[1] Although we are performing a two dimensional analysis, we leave the h_f parameter in our calculation. This simply moves the plane of interest a uniform distance h_f above the reference $\sum_0 x-y$ plane, and parallel to it. This does not complicate our two dimensional analysis, but allows us to use our two dimensional kinematic equations for the general three dimensional analysis, later. The reader may assume $h_f = 0$.

3. $T_{T_i}^{C_i}$

$T_{T_i}^{C_i}$ is the transformation which gives the coordinates in the T_i frame of a vector given in the C_i frame, which is the local frame of the contact point for the i^{th} finger. Since we have oriented the x-axis of the $\sum_{T_i}$ frame to point in the direction of the origin of $\sum_{C_i}$, the transformation between the frame consists solely of a displacement of length r_t along the x-axis, and is given by:

$$T_{T_i}^{C_i} = \begin{bmatrix} 1 & 0 & 0 & r_t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

The forward kinematic transformation for the i^{th} finger is simply given by the composition of the individual transformations for successive links, namely:

$$\begin{aligned} T_0^{C_i} &= T_0^{F_i} T_{B_i}^{T_i} T_{T_i}^{C_i} \\ &= \begin{bmatrix} -\sin(\Theta_i) & -\cos(\Theta_i) & 0 & -r_t \sin(\Theta_i) + l_f \cos(\beta_i + \theta_i) + b_{ix} \\ \cos(\Theta_i) & -\sin(\Theta_i) & 0 & r_t \cos(\Theta_i) + l_f \sin(\beta_i + \theta_i) + b_{iy} \\ 0 & 0 & 1 & h_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.4}$$

where $\Theta_i = \beta_i + \theta_i + \alpha_i$.

This mapping allows us to determine the coordinates, $\vec{r_{i,0}}$ in the reference frame, $\sum_0$ of the contact point of the i^{th} finger given the "joint variables" β_i, θ_i

and α_i, namely:

$$\vec{r_{i,0}} = T_{i,0}^{C_i} \vec{r_{i,C_i}} \tag{4.5}$$

Since the homogenous coordinates of the contact point, C_i in the frame, $\sum_{C_i}$ are simply $r_{i,C_i} = (0,0,0,1)^T$, we have:

$$\vec{r_{i,0}} = \begin{pmatrix} -r_t \sin(\Theta_i) + l_f \cos(\beta_i + \theta_i) + b_{ix} \\ r_t \cos(\Theta_i) + l_f \sin(\beta_i + \theta_i) + b_{iy} \\ h_f \\ 1 \end{pmatrix} \tag{4.6}$$

where: $\Theta_i = \beta_i + \theta_i + \alpha_i$.

4.1.3.2 2-D Jacobian

We derive the finger Jacobian matrices using the forward kinematic relationships developed in the previous section. This method appears frequently in the literature, e.g. in [23]. This method calculates a general Jacobian, J, from the kinematic matrices T_0^i, where i denotes the link number, starting with the base as link 0. This is done by calculating the columns of:

$$J = [J_1 J_2 J_3 \cdots J_n] \tag{4.7}$$

where the i^{th} column J_i is given by

$$J_i = \begin{cases} \begin{bmatrix} z_{i-1} \times (O_n - O_{i-1}) \\ \\ z_{i-1} \end{bmatrix} & \text{if joint i is revolute} \\ \\ \begin{bmatrix} z_{i-1} \\ \\ 0 \end{bmatrix} & if \text{ joint i is prismatic} \end{cases} \tag{4.8}$$

We can obtain z_i from the first three elements of the third column of T_0^i, and O_i from the first three elements of the fourth column of T_0^i. For each finger we must consider the last two columns of $T_0^{B_i}$, $T_0^{T_i}$, and $T_0^{C_i}$. The first and last of these occur above as equations 4.1 and 4.4, respectively. The second transormation is simply:

$$T_0^{C_i} = T_0^{F_i} T_{B_i}^{T_i} = \begin{bmatrix} -\sin(\Theta_i) & -\cos(\Theta_i) & 0 & l_f \cos(\beta_i + \theta_i) + b_{ix} \\ \cos(\Theta_i) & -\sin(\Theta_i) & 0 & l_f \sin(\beta_i + \theta_i) + b_{iy} \\ 0 & 0 & 1 & h_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.9}$$

where: $\Theta_i = \beta_i + \theta_i + \alpha_i$.

In actuality, there is only a single joint, whose joint variable is θ_i, but we have modeled the stationary link from the reference frame as a revolute joint with fixed (for each finger) joint variable β_i, and the stationary link from the fingertip center to the contact point as a revolute joint with fixed (for each finger) joint variable α_i. Thus, the columns for the Jacobian will all use the form for revolute joints. A little calculation yields, for the i^{th} finger:

$$J_i(\beta_i, \theta_i, \alpha_i, \vec{b_i}, l_f, r_t)$$
$$= \begin{bmatrix} -r_t\cos(\Theta_i) - l_f\sin(\Lambda_i) - b_{iy} & -r_t\cos(\Theta_i) - l_f\sin(\Lambda_i) & -r_t\cos(\Theta_i) \\ -r_t\sin(\Theta_i) + l_f\cos(\Lambda_i) + b_{ix} & -r_t\sin(\Theta_i) + l_f\cos(\Lambda_i) & -r_t\sin(\Theta_i) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{4.10}$$

where: $\Theta_i = \beta_i + \theta_i + \alpha_i$ and $\Lambda_i = \beta_i + \theta_i$.

Of course, in evalating J_i for a particular finger, we will have great simplification in that β_i, $\vec{b_i}$, l_f, and r_t constant for the finger in all configurations and α_i determined by the grasp, and assumed constant for small motions of the finger and/or object. We are now able to calculate the velocity of the contact point, given the joint velocity as well as the joint torques for given end-effector (contact) forces using equations:

$$\vec{v_{i,0}} = J_i q_i \tag{4.11}$$

$$\vec{\tau} = J_i^T \begin{bmatrix} f_0^{C_i} \\ n_0^{C_i} \end{bmatrix} \tag{4.12}$$

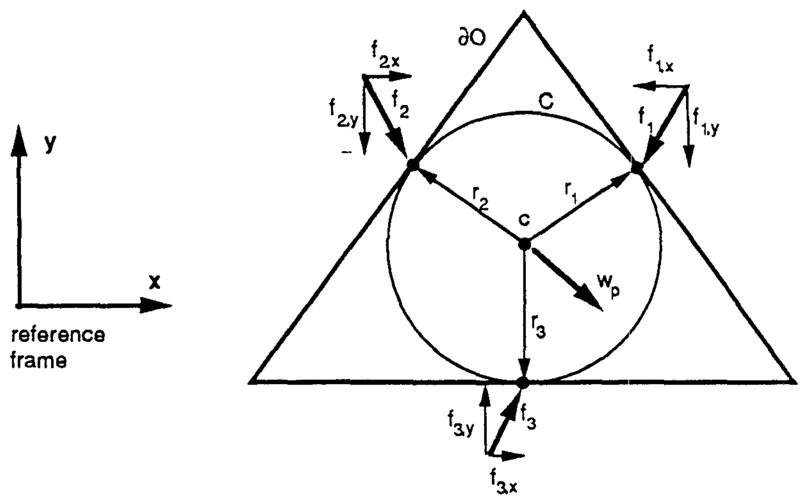


Figure 4.6: Variables for Grasp Force Calculations

4.1.3.3 Grasp Force Calculation

We now turn to the calculation of the grasp forces at the contact points. Figure 4.6 provides illustration of the relevant quantities for the following discussion. Once we have determined the contact forces, we may easily determine the required joint torques from equation 4.12. In order to solve this problem, we must determine six total components of the contact forces (three forces with two components each). So we shall require a total of six linear equations. The first three of these come immediately from the need to balance the perturbing wrench, $w_p = (f_{p,x}, f_{p,y}, n_p)^T$. We desire that the force components of the grasp wrenches balance the perturbing force components, $f_{p,x}$ and $f_{p,y}$ and that the torque components of the grasping wrenches balance the perturbing torque component, n_p. These requirements give rise to the equations:

$$\sum_{i=1}^{3} f_{i,x} = f_{p,x}$$

$$\sum_{i=1}^{3} f_{i,y} = f_{p,y} \tag{4.13}$$

$$\sum_{i=1}^{3} r_i \times f_i = n_p$$

We may rewrite these equations, noting that $r_i \times f_i$ may be written as $(r_i \times) \cdot f_i$, where $(r_i \times)$ is simply given by $(-r_{i,y}, r_{i,x})$ if $r_i = (r_{i,x}, r_{i,y})$. Thus we have:

$$Af = w_p \tag{4.14}$$

where:

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ -r_{1,y} & r_{1,x} & -r_{2,y} & r_{2,x} & -r_{3,y} & r_{3,x} \end{pmatrix} \tag{4.15}$$

$$f = (f_{1,x}, f_{1,y}, f_{2,x}, f_{1,y}, f_{3,x}, f_{3,y})^T \tag{4.16}$$

and

$$w_p = (f_{p,x}, f_{p,y}, n_p)^T \tag{4.17}$$

To determine the normal and tangential wrenches applied by the fingers at the contact points, we will determine the projections of the grasping forces on

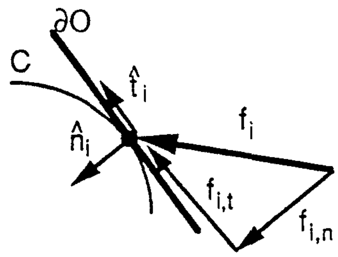


Figure 4.7: Contact Force Decomposition

the unit normal, n_i and the unit tangent, t_i at the contact point, c_i which is the point of application of the grasping force, f_i. Figure 4.7 shows the relevant quantities. If we define $R_0^{C_i}$ to be the rotation matrix which translates vectors in $\sum_{C_i}$ to vectors in $\sum_0$, and noting that the inward contact normal is given by $(1,0,0)^T$ in $\sum_{C_i}$, we have

$$\hat{n}_i = \begin{bmatrix} -\sin\Theta_i & -cos\Theta_i & 0 \\ \cos\Theta_i & -\sin\Theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -\sin\Theta_i \\ \cos\Theta_i \\ 0 \end{pmatrix} \tag{4.18}$$

Similarly, the unit tangent is given by $(0,1,0)^T$ yielding:

$$\hat{t}_i = \begin{bmatrix} -\sin\Theta_i & -cos\Theta_i & 0 \\ \cos\Theta_i & -\sin\Theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -\cos\Theta_i \\ -\sin\Theta_i \\ 0 \end{pmatrix} \tag{4.19}$$

And the normal and tangential components of the force are given by:

$$f_{i,n} = (f_i \cdot \hat{n}_i)\hat{n}_i \tag{4.20}$$

and

$$f_{i,t} = (f_i \cdot \hat{t}_i)\hat{t}_i \tag{4.21}$$

respectively.

Now we need to augment the three equations from 4.14 with three additional equations. We rely on our knowledge of the grasp requirements to deduce these remaining equations. Since we have required the perturbing force, f_p to be balanced by the normal components of the contact forces (i.e. the components of the contact forces aligned with their respective contact normals), we may require that the tangential components of the contact forces impose no net force on the object. This requirement leads to the equation:

$$\sum_{i=1}^{3}(f_i \cdot \hat{t}_i)\hat{t}_i = 0 \tag{4.22}$$

This provides us with two equations (one for each component). For our final equation, we demand that the normal forces are of a given total magnitude. This equation is expressed as:

$$\sum_{i=1}^{3} f_i \cdot \hat{n}_i = M \tag{4.23}$$

where M is a parameter that we can specify. The effect of increasing M will be to increase the total normal grasping force, which will have the effect of increasing the internal forces of the grasp. In effect, increasing M "tightens the squeeze" on the object. It should be noted that increasing the internal forces does not always increase the stability of the grasp [6], especially when the grasp points are at points of convex curvature. However, for our MIC-produced grasp, we will rarely encounter situations in which we select grasp points at which stability is decreased by increasing M. More often, we will want to limit M due to the fragility of the object to be grasped, or by the desire to limit the joint torques we apply. Still, it is possible that we cannot achieve a certain grasp with a given M simply because we have demanded forced that are higher than can be produced at maximum joint torque, or because we have demanded to little normal force to adequately balance the perturbing wrench, w_p. In this case, it will be neccessary to iterate the grasp force calculation, adjusting the value M, appropriately. For convenience, we can estimate the value of M to start the process with the knowledge that the normal components must balance the perturbing force. As an initial guess, we can assume:

$$M = 3\|f_p\| + \mu^{-1}\frac{n_p}{\|r_C\|} \tag{4.24}$$

This heuristic has the effect of making the normal grasping forces on the order of the perturbing force plus enough total normal force to cancel the perturbing torque. It also gives a bound on the maximum perturbing moment, n_p, which may be cancelled in the sense that since:

$$n_{max} = \sum_{i=1}^{3} r_i \times f_{i,t} = \mu r \sum_{i=1}^{3} fi, n \leq \mu r M \tag{4.25}$$

This final equation gives us a full complement of six equations. Trial numerical investigations reveal that the matrix of coefficients derived from these equations is not trivially singular, so we may expect that the generic instance of these equations is solvable. If matrix is singular for a particular instance of the parameters, it is easy to move away from the singular value of the paramenters by a slight adjustment of the grasp. However, since the coefficients are dependent on the locations of the contact points, the conditions governing the selection of these points will determine the liklihood of singularity of the coefficient matrix. The fact that these points are non-collinear, and that the unit normals at the contact points all intersect at the center of the maximal inscribed circle makes the chance of encountering a singular coefficent matrix unlikely.

Assuming we have a nonsingular set of coefficients, we may solve for the contact forces using the relation:

$$f = \mathcal{A}^{-1}(f_{p,x}, f_{p,y}, n_p, 0, 0, M)^T \tag{4.26}$$

where $\mathcal{A}$ is the matrix A augmented by the three equations 4.22 and 4.23. We then determine the required joint torques using equation 4.12. We can then determine whether the maximum joint torque is exceeded at any joint. If so, we can first try to solve 4.26 for lower values of M. If this also fails, then we must resort to trying a new grasp.

This method allows us to answer task specific questions about stability. Given a specific task for which the object is being grasped, we need to know

1. static loading (weight)
2. dynamic loading (accelerations)
3. task specific disturbances or maximum allowable disturbances

Given this information about the wrenches which may be externally applied to the object, we can use the above method to determine whether the grasp will remain secure. We mention that the above method is probably conservative, since we are independently setting the rough magnitudes of the internal forces. If we find a solution, it is quite possible that a solution could also be found using a weaker value of M. It is also conservative in the sense that we are requiring the tangential components to exert no net force on the object, with the perturbing force entirely countered by the normal components of the grasping force. It is possible that a solution which requires less total applied force could be found absent this assumption. Still, the assumption allows us to find a solution to the grasp force problem and to deduce the necessary torques, so it is of unquestionable utility.

We wish to point out that our selection of form closure grasp, accomplished via the MIC algorithm has allowed us to decouple the problem somewhat. We can use the normal components of the grasping force to solve the complete restraint problem, assuming we have enough generated torque, and then examine the induced moment problem seperately. This would be more difficult for a general grasp as we would need to first find the center of force to decouple it in this manner. Given that we have used a form closure grasp, this information is available at no cost.

4.2 Three dimensional Grasp Stability

A complete analysis of three dimensional stability requires us to essentially repeat the above analysis carrying along the additional terms. To a large extent the extra degrees-of-freedom allowed the perturbing wrench will require that we know a great deal about the local geometry of the object near the contact points in the object $\hat{z}$ direction. If we make the assumption that the object is locally prismatic, that is, that the sides are parallel to the object z-axis near the contact points, we can easily show that there are sufficient degrees-of-restraint available due to the fact that we are grasping using frictional fingers which give us

1. an additional frictional component in the object $\hat{z}$ direction

2. an additional frictional torque component about the contact normal

The $\hat{z}$ components can be used to prevent slipping in the $\hat{z}$ direction, provided that we can generate enough normal force (this may cause us to increase the wrench intensities λ_i). The torque components can easily be shown to provide immunity to rotation about any axis in the grasp plane. This is because the axes around which these torques act are the finger normals, which positively span the grasp plane. Since any perturbing torque in the grasp plane will rotate about some axis in the plane, there will be some positive combination of intensities that will cancel any perturbing torque. Additionally, the $\hat{z}$ components provide resistance to torques. So it is easy to see that resistance to the extra degrees of possible perturbation is feasible for locally prismatic objects. This makes a good criterion for grasp plane selection as well. Given a specific orientation for the grasp plane (which is often provided by the direction of approach or by task requirements), one can simply attempt MIC form closure grasps at intervals along the grasp plane perpendicular and select the plane as that near which the two dimensional projections of the grasp points do not vary much.

For objects for which no locally prismatic grasp can be found, the issue of three dimensional stability is more difficult. Following the reasoning we employed in the two dimensional case, it seems that the normal grasping forces lying in the friction cone at the contact point is a minimal requirement. Beyond that, one must consider the effects of $T_{T_i}^{C_i}$ having an additional rotational component, although the analysis will follow much the same path as we followed for the two dimensional case. We leave this issue now, with hope that it will be pursued by future researchers.

Chapter 5

Inverse Kinematics and Path Planning

This chapter describes some of the necessary calculations which must be perfomed for the MDH to achieve a grasp, whose contacts are determined by the methods outlined previously. We start out with a description of how the selection of contact points on the object is mapped into a selection of joint angles for the MDH fingers. Since many equivalent choices of joint angles which achieve the same relative finger tip placement are possible, we examine criteria for selecting among them. Once we have determined what the finger angles should be, we need to calculated a sequence of moves to get the fingers from their current configuration into the desired one. The first problem we face is that the fingers can collide, if the movement is not carefully coordinated. We provide an algorithm for correctly attaining a desired configuration despite the existence of this *self-collision obstacle*. Now, in the case where there is in object in the hand's workspace, the problem of attaining the desired grasp configuratiion is more difficult. In fact, it will often not be possible to achieve a grasp directly from an arbitrary initial configuration. We provide an analysis of the obstacle avoidance problem for the MDH, and then modify the finger movement algorithm to achieve the grasp directly, if possible, or otherwise to signal the host computer that the hand must be moved in such a way that the object no longer impedes finger movement.

5.1 Inverse Kinematics

The inverse kinematics problem for the MDH is that of determining the finger angular positions $\theta_1, \theta_2, \theta_3$, which cause the contact points of the MDH fingers to attain the spatial coordinates required for the grasp. The discussion herein is an expansion on the basic ideas of inverse kinematics for the MDH presented in [20]. Since the algorithm presented in [20] finds the joint angles corresponding to the placement of the fingertip centers, we must enlarge the object boundary by r_t so that the placement points correspond to the fingertip centers. We briefly present the numerical method of Loncaric for finding the solutions for a given grasp triangle. Multiple solutions may exist due to symmetries in the mechanics of the hand (since fingers may be arbitrarily relabeled), as well as to the inverse kinematic map being multivalued.

Given the mechanical configuration of the MDH, we may desire to achieve a grasp, specified by three spatial points, by rotating the crank-like fingers to specific angular configurations, denoted by $\theta_1, \theta_2, \theta_3$. If the hand is moveable as a unit, as it would be if it were attached to a 6 degree-of-freedom arm, then the absolute spatial locations of the grasp are unimportant, since the spatial relationships of the grasp are captured simply by considering the grasp triangle, consisting of the three grasp points. All necessary information is captured by supplying the lengths of the three sides of the grasp triangle, $l_{i,j}$, which is the length of the side connecting the contact point on finger i to that on finger j. The solution to the inverse kinematic map is then obtained as a function of these values. Once a solution which realizes a given grasp triangle is found, and the hand driven to attain that configuration, the hand must be maneuvered so that the vertices of the grasp triangle coincide with the required spatial locations of the grasp contact points on the object's surface.

We shall sketch the basic method outlined in [20]. This method will produce initial joint angle estimates θ_i. The method used is to fix one joint angle – without loss of generality, we choose θ_1 – and then to solve two four-bar linkage problems to find θ_2 and θ_3 such that the desired lengths $l_{1,2}$ and $l_{3,1}$ are achieved. We then solve the reduced inverse kinematics problem, consisting of determining γ_1 such that the required length $l_{2,3}$ is obtained. Figure 5.1 shows the relationships of the variables needed for the basic inverse kinematics problem.

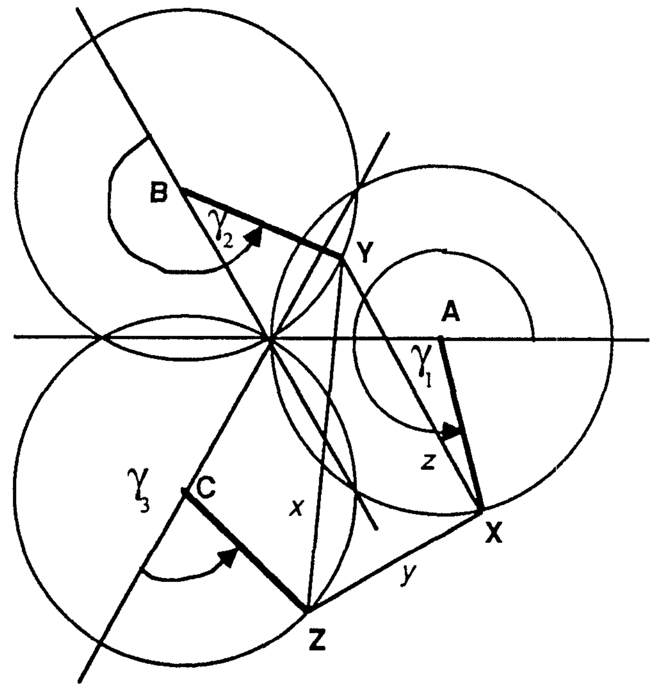
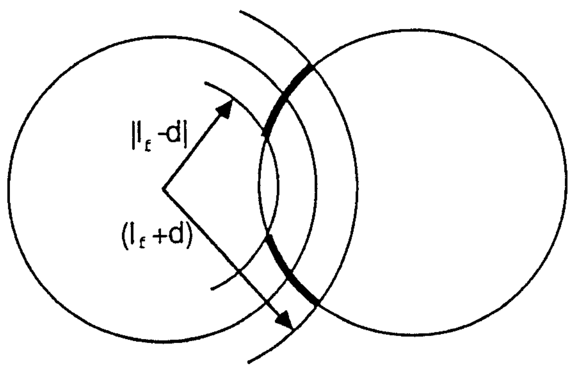


Figure 5.1: Variable Relationships for Basic Inverse Kinematics

Since not all arbitrarily chosen values of θ_1 will yield solutions to the two underlying four-bar linkage problems, we must restrict our initial choice of θ_1. Refer to Figure 5.2 to clarify the following discussion. Assuming that the horizontal portion of the finger crank is of unit length, we see that θ_1 is confined to lie in one of two disjoint intervals for triangles with the relevant side of length less than $2-\sqrt{3}$. These intervals merge when the side length is exactly $2-\sqrt{3}$. As the side length continues to grow, the permissible range for θ_1 grows until it includes the entire circle at side length $\sqrt{3}$. Beyond this length, the interval seperates again into two disjoint intervals, with permissible regions on the opposite side of the circle from those indicated for short side lengths. No solutions for θ_1 exist when side length exceeds $2+\sqrt{3}$. Lengths of exactly $2-\sqrt{3}$ and $\sqrt{3}$



Four bar linkage problem can be solved for those crank positions which are reachable. The "fully extended" circle of radius $(l_f + d)$ and the "bent over" circle of radius $|l_f - d|$ limit the reachable points. Here, $0 < d < (2 - \mathrm{sqrt}(3))l_f$

Figure 5.2: Restricted Domains for Initial Choice of θ_1

produce singular configurations which must be considered seperately.

Due to the symmetry of the mechanism, when there are solutions to the four-bar linakge problems, each problem permits two solutions. By examining the possible pairings of these solutions, we have four possible lengths of the side $l_{3,2}$, and thus there are four branches of the solution of the reduced inverse kinematics problem that we must track as we search through the range of permissible values of θ_1. For an equilateral triangle of side length $\sqrt{3}$, however, we expect a continuum of solutions, so a discrete solution of the basic inverse kinematics problem should not be expected.

The steps required to achieve numerical solution to the basic inverse kinematics problem are as follows:

1. Given the three grasp points, c_1, c_2, c_3, compute the triangle side lengths, $l_{1,2}, l_{2,3}, l_{3,1}$, relabeling as necessary to make $l_{2,3}$ closest of the three to $\sqrt{3}$ (with lengths normalized to crank length). This heuristic helps maximize the domain restriction in the next step.

2. Identify the intervals of values for θ_1 for which the four-bar linkage problems have solution(s).

3. For each of the intevals found, look for a solution of the reduced inverse kinematics problem by bracketing. Four branches must be tracked, as mentioned previously. Once a solution has been bracketed, refine it using the Brendt method. Otherwise, if an extremum has been bracketed, find the extremum and bracket there to check for potential double roots.

The basic procedure presented in [20] has been widely tested for all legal triangles with side lengths of up to $2 + \sqrt{3}$ in steps of $1/128$ of full range. This computation of over 10^5 inverse kinematic solutions has demonstrated robustness and accuracy of the basic method. The computation of the correction factor has no adverse effect on the robustness of the numerical solution, but does slightly lower the limit on the size of the maximal grasp triangle that can be obtained. The correction does, however, allow the method to be used with the MDH spherical fingertips, and provides the desirable characteristic that the contact angles thus generated are at or near 0 or π, which has advantageous implications from the point of grasp stability, as indicated in Chapter 4. The method will provide 0, 2, 4, 6 or 8 solutions for all legal triangles, only producing 0 solutions for extremely large triangles, where physical limitations of the mechanism come into play. Furthermore, since the contact points may be arbitrarily relabeled, there is an additional multiplicity of solutions of either $3! = 6$ if the orientation of the palmar plane is not required to be on a specific side of the grasp plane, or 3 otherwise. Thus, in the most general case, we may have up to 48 possible solutions to the inverse kinematic problem!

Having multiple solutions to the inverse kinematics may make the problem more difficult computationally, but provides a desirable degree of flexibility. Some solutions may not be feasible due to clearance problems between the mechanism and the object, while others may be undesirable due to the lack of mechanical advantage they afford. With the addition of a clearance checking routine, we are able to throw out the first class of solutions, and with the ability to analyze the stability of grasps, given joint angles as parameters, we can select from amongst the possible grasps by ranking according to the mechanical advantage provided. This capability makes our solution to the inverse kinematic problem for the MDH robust, accurate and flexible, and therefore of great utility in selecting the hand configuration needed to achieve a specified grasp.

5.2 Path Planning

We next deal with the issue of how to drive the fingers of the MDH to a postion from which we can grasp the objcet. We designate a position with the finger contacts placed just exterior to the object surface as a pre-grasp configuration. The question, then, is how do we drive the hand from an arbitrary initial configuration to an arbitrary goal pre-grasp position. The problem, of course, is that the fingers can interfere with each other in the center of their range, in the vicinity of π radians. This is evident from the observation that with all three fingers at π radians, the fingertip *center* coincide, causing complete overlap of the 0.5 inch radius fingertips. We describe the self collision problem from a mathematical perspective, and present the problem as the need to avoid an obstacle in the hand's configuration space. Using this as our model, we then present an algorithm which produces collision free paths connecting any two configurations.

5.2.1 Self-Collision Avoidance

The self collison problem may be considered from a mathematical perspective as the need to avoid a surface at which the fingertips touch. Since we begin on the exterior of this surface, we confine the domain of our search for paths to the region exterior to this surface. Since the configuration space of the MDH gripper stage consists of three angles $(\theta_1, \theta_2, \theta_3)$, we may model the configuration space a three dimensional cube, with sides composed of the intervals $[0, 2\pi]$. In actuality, the entire range of the configuration space is not reachable since the index pin occupies a portion of the angular range of the fingers. We find that the range eliminated by the index pin is $\gamma = 2\tan^{-1}(w_p/l_c)$, where $w_p = 0.5(d_p + d_t)$ is the width of the angular region inaccessible due to the index pin, d_p is the diameter of the index pin, d_t is the diameter of the finger tube, and l_c is the distance along the finger tube from the rotation axis of the finger to the center of the index pin. Numerically, the pin angular occlusion is approximately 0.797 radians, thus yielding a configuration cube of side length $2\pi - 0.797 \approx 5.486$ radians.

The position of a fingertip center in the finger base reference frame $\sum_{B_i}$, is given by the equation:

$$\vec{r}_{i,B_i} = \begin{pmatrix} l_f \cos(\beta_i + \theta_i) \\ l_f \sin(\beta_i + \theta_i) \\ h_f \end{pmatrix} \tag{5.1}$$

Assume the base reference frame, $\sum_0$ to be defined such that its origin is at $\sum_{B_1}$ with $\sum_{B_2}$ located along the positive x-axis. The the location of the origin of $\sum_{B_2}$ is given by $\vec{r}_{B_i} = (\sqrt{3}l_f, 0, 0)^T$. Then the first fingertip position is given by:

$$\vec{r}_{1,0} = \begin{pmatrix} l_f \cos(\beta_i + \theta_i) \\ l_f \sin(\beta_i + \theta_i) \\ h_f \end{pmatrix} \tag{5.2}$$

and the position of the second fingertip is given by:

$$\vec{r}_{2,0} = \begin{pmatrix} \sqrt{3} l_f + l_f \cos(\beta_i + \theta_i) \\ l_f \sin(\beta_i + \theta_i) \\ h_f \end{pmatrix} \tag{5.3}$$

so the difference in position is given by:

$$\begin{aligned} d_{1,2}(\theta_1, \theta_2) &= \|\vec{r}_{2,0} - \vec{r}_{1,0}\| \\ &= \sqrt{(\sqrt{3} l_f + l_f \cos(\Theta_2) - l_f \cos(\Theta_1))^2 + (l_f \sin(\Theta_2) - l_f \sin(\Theta_1))^2} \end{aligned} \tag{5.4}$$

where $\Theta_i = \beta_i + \theta_i$.

Due to the symmetry between this pair and the $2-3$ and $3-1$ pairs, we will have similiar expressions for $d_{2,3}(\theta_2, \theta_3)$ and $d_{3,1}(\theta_3, \theta_1)$. The collision between two fingertips, i and j, will occur when:

$$d_{i,j}(\theta_i, \theta_j) = 2r_t \tag{5.5}$$

In order to allow for inaccuracies in finger positioning due, for example to overshoot in the finger control, we pad this surface by adding a tolerance τ_0, so that the surface of our new obstacle, O' is given by the equations:

$$d_{i,j}(\theta_i, \theta_j) = 2r_t + \tau_0 \tag{5.6}$$

The three equations will yield generalized cylinders, oriented parallel to the axis of the joint variable not involved. We plot these surfaces in the configuration cube (with $\tau_0 = 0$) for visualization in Figure 5.3.

5.2.2 An Algorithm for Self-Collision Avoidance

From a geometric perspective, then, the self collision avoidance problem can be recast to that of finding a path in this configuration space which avoids the obstacle we have described. Using this as the basis for our analysis, we have developed an algorithm which achieves the objective of finding a collision-free path joining any two configurations. The path thereby generated is not guaranteed to be optimal in all cases, but we shall see, the paths are quite good, and their generation is direct, compared to the typically slow convergence of algorithms using a relaxation method for generation of the optimal path.

5.2.2.1 Straight Line Path

Throughout this section we will present some figures demonstrating the various stops of our algorithm. These figures, unless otherwise indicated, correspond to the case of starting at configuration point $(0.0, 0.0, 0.0)$ and ending at configuration point $(5.49, 5.49, 5.49)$. The straight line connecting these points is the major diagonal of the reachable configuration cube, and passes through the point (π, π, π), which is a point of pairwise collision for all three obstacle axis,

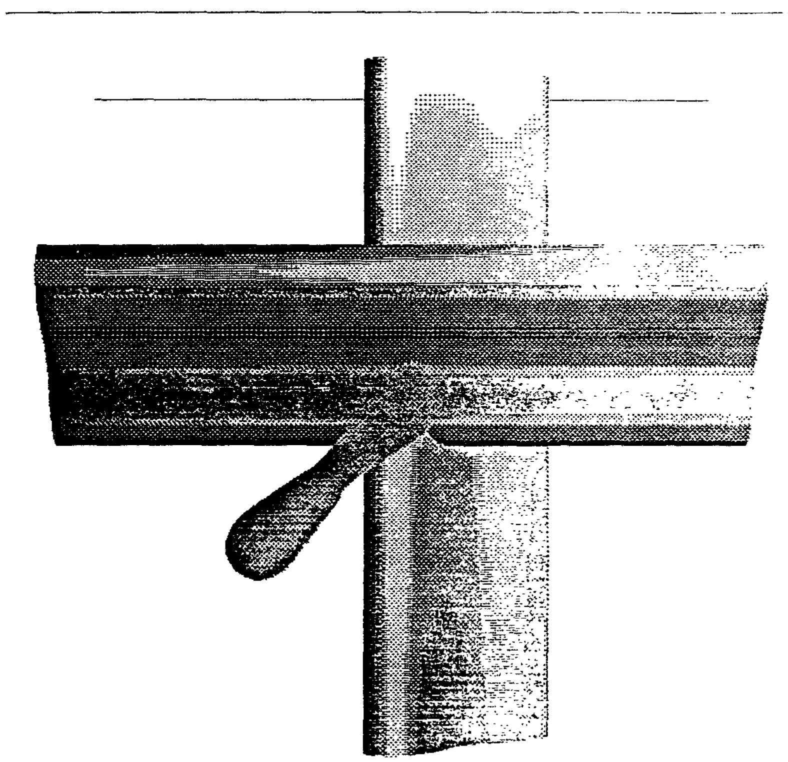


Figure 5.3: The MDH Configuration Space with Self-Collision Obstacle

so this is, in some sense, a canonical example of the use of the path planning algorithm. Figure 5.4 shows the start and goal points, represented by small spheres, with the start position in the upper left and goal position in the lower right (the display program allows the configuration to be rotated to an aribrary orientation).

Given two configurations, $\mathcal{S}$ and $\mathcal{G}$, the start and goal configurations, respectively, we wish to generate as short a path as possible between them. Absent the self-collision obstacle, the optimal path would obviously consist of a straight line in configuration sapce joining these two configurations (we shall use the term "points" for configurations henceforth). In fact, there are a large number of start goal pairs for which this path does not intersect the obstacle, and so for which this path may easily be selected. Figure 5.5 shows the optimal path connecting our example start and goal pair.

5.2.2.2 Feasible Roadmap Path

In the cases for which the optimal path does intersect the obstacle, we must find the path by a different method. What we shall do is to construct a feasible path by moving around the faces and edges of the cube as much as possible. This area of the configuration space is more likely to be contention free than the interior of the cube. The algorithm for constructing this path is referred to as a roadmap algorithm. Our initial goal will be to construct a collision-free

$\theta_1 + \theta_2$	$\theta_2 + \theta_3$	$\theta_3 + \theta_1$	x	y	z
$\leq 2\pi$	$\leq 2\pi$	$\leq 2\pi$	-1	-1	-1
$> 2\pi$	$\leq 2\pi$	$\leq 2\pi$	0	0	-1
$\leq 2\pi$	$> 2\pi$	$\leq 2\pi$	-1	0	0
$> 2\pi$	$> 2\pi$	$\leq 2\pi$	0	1	0
$\leq 2\pi$	$\leq 2\pi$	$> 2\pi$	0	-1	0
$> 2\pi$	$\leq 2\pi$	$> 2\pi$	1	0	0
$\leq 2\pi$	$> 2\pi$	$> 2\pi$	0	0	1
$> 2\pi$	$> 2\pi$	$> 2\pi$	1	1	1

Table 5.1: Contention-Free Directions from Interior Configurations

path from a point to an edge of the cube. From that point, if is a simple matter to get to any point on any other edge by traversing the cube edges, which are collision-free.

The method for obtaining a path to a cube face is determined by observing that the configuration space can be divided into octants such that in each octant[1], there is direction which is guaranteed to move away from the obstacle. The directions are determined from vectors whose x, y, and z components are listed in Table 5.1.

Following this direction will cause path to arrive at either an edge of a face of the cube. If we are at an edge, we are done with this portion of the algorithm. If we are at a face, we must get to an edge. This is accomplished by a similiar mechanism, this time using the octant information above, along with the knowledge of which face we are on (determined by calculating the shortest

[1]This method of finding contention-free paths from the cube interior to a cube edge results from the work of J. Loncaric, relayed in private correspondence.

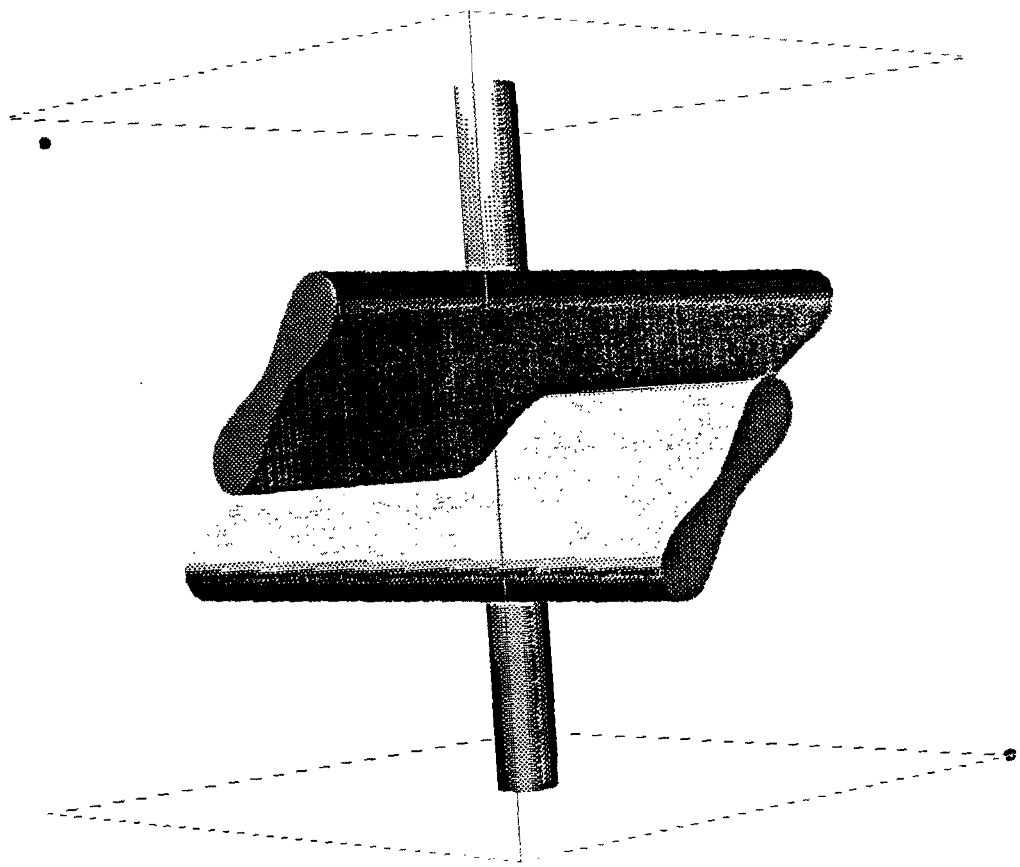


Figure 5.4: Start and Goal Configurations are Specified

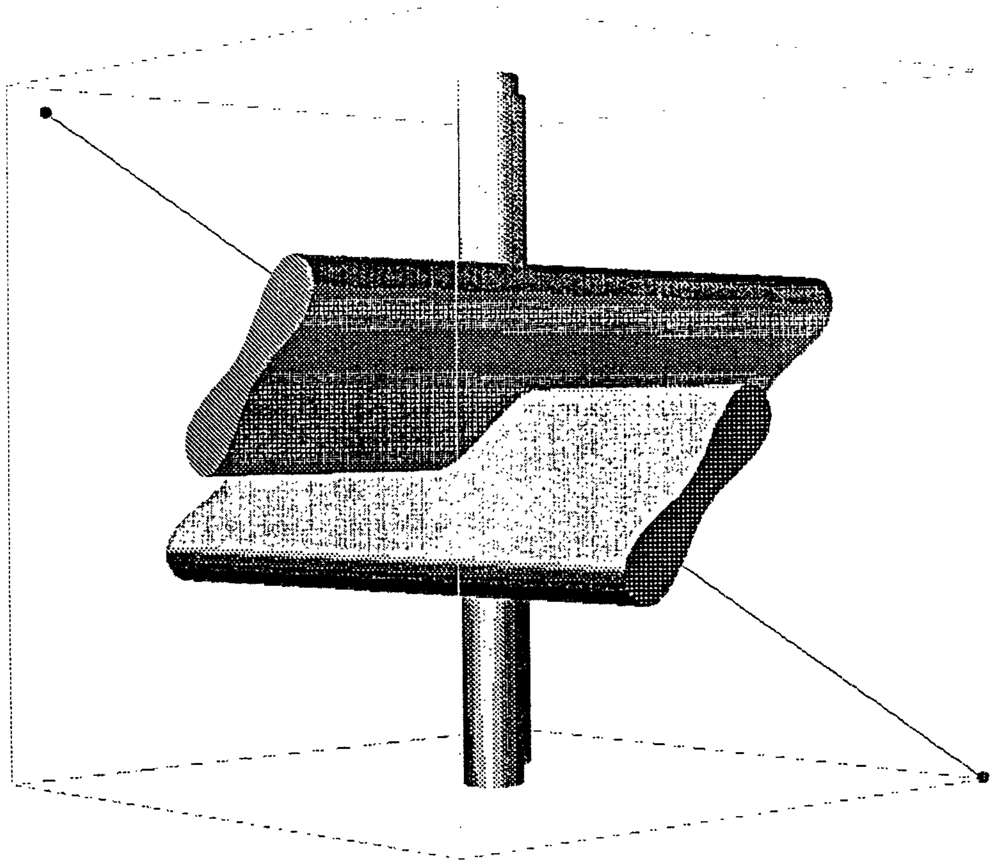


Figure 5.5: The Straight Line Path Connecting $\mathcal{S}$ and $\mathcal{G}$

$\theta_1+\theta_2$	$\theta_2+\theta_3$	$\theta_3+\theta_1$	Face	x	y	z
$\leq 2\pi$	$\leq 2\pi$	$\leq 2\pi$	$\theta_1 = 0$ *or* 2π	0	-1	-1
			$\theta_2 = 0$ *or* 2π	-1	0	-1
			$\theta_3 = 0$ *or* 2π	-1	-1	0
$> 2\pi$	$\leq 2\pi$	$\leq 2\pi$	All	1	1	0
$\leq 2\pi$	$> 2\pi$	$\leq 2\pi$	All	0	1	1
$> 2\pi$	$> 2\pi$	$\leq 2\pi$	All	-1	0	-1
$\leq 2\pi$	$\leq 2\pi$	$> 2\pi$	All	1	0	1
$> 2\pi$	$\leq 2\pi$	$> 2\pi$	All	0	-1	-1
$\leq 2\pi$	$> 2\pi$	$> 2\pi$	All	-1	-1	0
$> 2\pi$	$> 2\pi$	$> 2\pi$	$\theta_1 = 0$ *or* 2π	0	1	1
			$\theta_2 = 0$ *or* 2π	1	0	1
			$\theta_3 = 0$ *or* 2π	1	1	0

Table 5.2: Contention-Free Directions from Cube Face to Cube Edge

distance to intersection of the face plane). These directions are given by x, y, and z components listed in Table 5.2.

Following this direction will yield a collision-free path to the cube edge. The combination of these two portions will yield a guaranteed collision-free path from anywhere in the cube interior, outside the obstacle, to a cube edge. From here it is a simple thing to find the shortest path along the cube edge which connects one edge point to another. We begin by labelling the cube vertices with a 3-bit binary number, as indicated in Figure 5.6, such that if two vertices are joined by a common edge, their label differ in at most one bit position. It is evident then, that the number of edges one must traverse to get from one vertex to another is the number of bits in which the vertex labels differ. To get from one edge point to another, we simply observe that we may begin by moving to either of the vertices which are connected by the edge on which we are located.

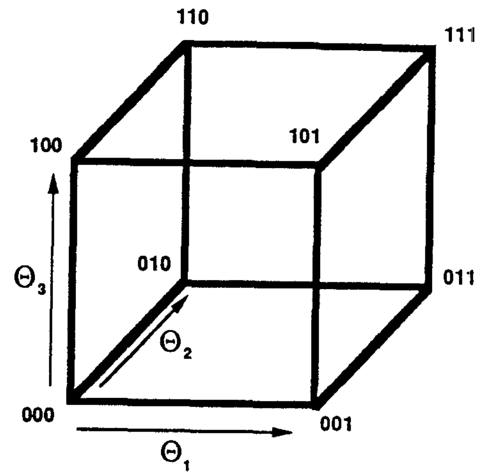


Figure 5.6: Labeling of Configuration Cube Vertices

The edge point corresponding to the goal position is similiarly situated between two vertices. If we perform a bitwise comparison between the vertex labels for the four possible pairs of start vertex and goal vertex, we can determine which path requires the least number of full edge traversals. In the case of a tie, we can break the tie by determining the length from the start edge point to each of the start vertices and from the goal pint to each of the goal vertices and use the pair which has the least number of full edge traversals and the shortest edge to vertex distance sum.

Thus, given a start configuration $\mathcal{S}$ and a goal configuration $\mathcal{G}$, we have derived a feasible path connecting them. It is feasible in the sense that there are guaranteed to be no collisions with obstacle, and thus no collisions of fingertips. The path follows the sequence:

$$\mathcal{S} \rightarrow \mathcal{SF} \rightarrow \mathcal{SE} \rightarrow \mathcal{V} \rightarrow \mathcal{V} \rightarrow \mathcal{V} \rightarrow \mathcal{GE} \rightarrow \mathcal{GF} \rightarrow \mathcal{G}$$

where $\mathcal{SF}$ is the start face point, $\mathcal{SE}$ is the start edge point, $\mathcal{GF}$ is the goal face point, $\mathcal{GE}$ is the goal edge point, and $\mathcal{V}$ indicates an interim vertex on the path. Of course, any of these may be omitted if not required for the given roadmap path. Figure 5.7 shows the feasible roadmap path generated for our example case.

5.2.2.3 Projected Path

The path we have constructed will work. Unfortunately, it is about the worst feasible path we could sensibly choose, in terms of path length, and thus in terms of the speed with which we may switch between configurations. What we would like is to have a path which is as close as possible to the optimal straight line path, but which has the feasiblity inherent in the roadmap path. Our algorithm finds such a compromise path essentially by projecting the roadmap path on the optimal path. For this purpose, we pad the obstacle by an additional tolerance, τ_1, yielding the obstacle O''. That is, the distance functions for this obstacle are given by:

$$d_{i,j}(\theta_i, \theta_j) = ||\vec{r}_{i,0} - \vec{r}_{j,0}|| + \tau_0 + \tau_1 \tag{5.7}$$

If the projection of the roadmap onto the feasible path is not occluded by O'', we chose the corresponding optimal path points for our projection path. Where the projection is occluded by the obstacle, we choose the points of intersection of the segment connecting the roadmap path point and optimal path point with

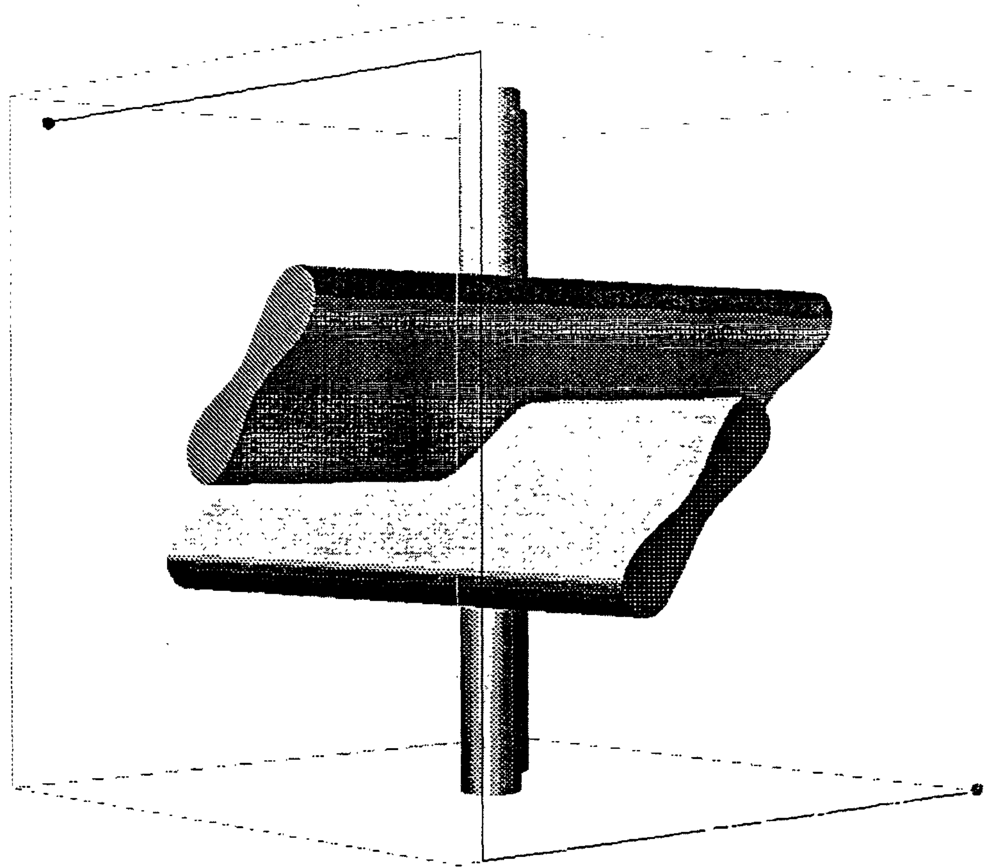


Figure 5.7: Feasible Roadmap Path Connecting $\mathcal{S}$ to $\mathcal{G}$

the obstacle surface. The piecewise linear path thus constructed is at least close to being feasible, since the vertices along the path are feasible by construction, lying on the non-occluded optimal path or on the surface of O''. Figure 5.8 shows the projection segments, indicating how the segments provide points for our projection path, while Figure 5.9 shows the path thereby constructed.

5.2.2.4 Augmenting the Projected Path

But can we say that the entire path is feasible? The answer is, sadly, no. As the projected path traverses the segment joining any two of its points, it may intersect the obstacle. We can minimize the liklihood of this by increasing the number of points in the projection, and by padding the obstacle with an additional tolerance, but we cannot entirely avoid it. We therefore include in our algorithm a routine which scans along the projected path, checking for intersections with the obstacle and fixing them when they occur. The mechanism for this path augmentation is straightforward. For each segment of the projected path, we numerically calculate the minimum distance along the segment and the point at which it occurs. Since the segment endpoints are feasible by construction, we are assured that if the minimum distance along the segment is in violation of the distance constraint, then it occurs in the segment interior. We fix violating segments by moving the minimum point away from the obstacle. To do this, we invoke a protion of our earlier roadmap algorithm to find a face

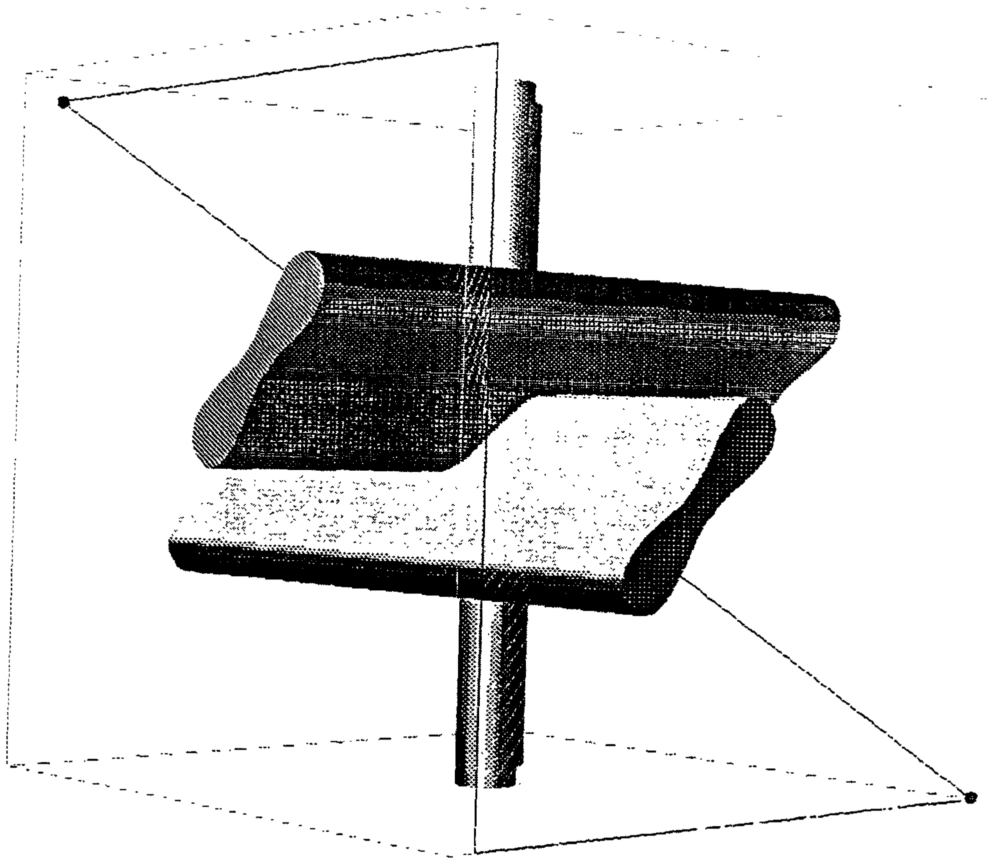


Figure 5.8: Projection From Feasible Path to Straight Line Path

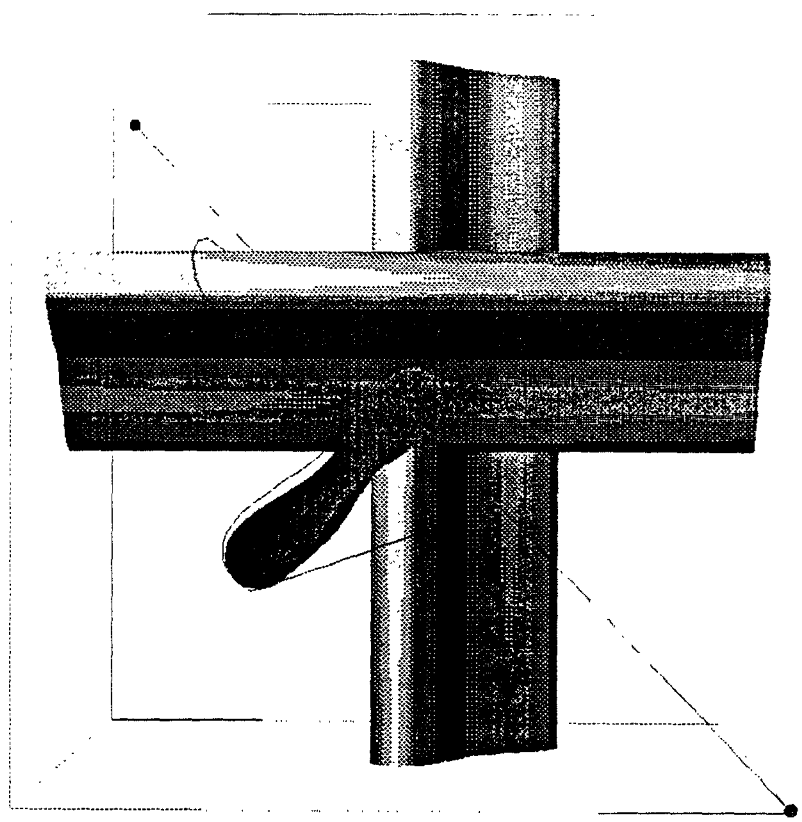


Figure 5.9: Projection Path Follows Obstacle Contour

point whose direction guarantees a collision free move away from the object's surface. We then find the intersection of the segment joining the minimum point and the face point with the surface of O''. This new point is feasible, as it is on the surface of the (padded) obstacle, and can be used as an interim vertex, dividing the original segment in two. This routine is then recursively applied to the resulting two halves, until the entire span between the original segment endpoints is contention free. By using O'' as the surface for interim point placement, we ensure that the recursion need not proceed too deep, since points are places well above the O' surface. In practice, recursions deeper than 5 levels are rare. Figure 5.10 shows that some small augmentations were required even for our example path.

5.2.2.5 Shortening the Augmented Path

After the path is augmented by the above procedure, we have a guaranteed contention-free path that lies as close as possible to the optimal path. This is not the best path we can create, however. In some places, the path may be concave away from the obstacle. In these cases, it is always possible to simply omit the points which lie in the concavity. Since this moves the path away from the obstacle, it is assured to maintain feasibility, while shortening the path. Algorithmically, this procedure is accomplished by scanning the augmented path

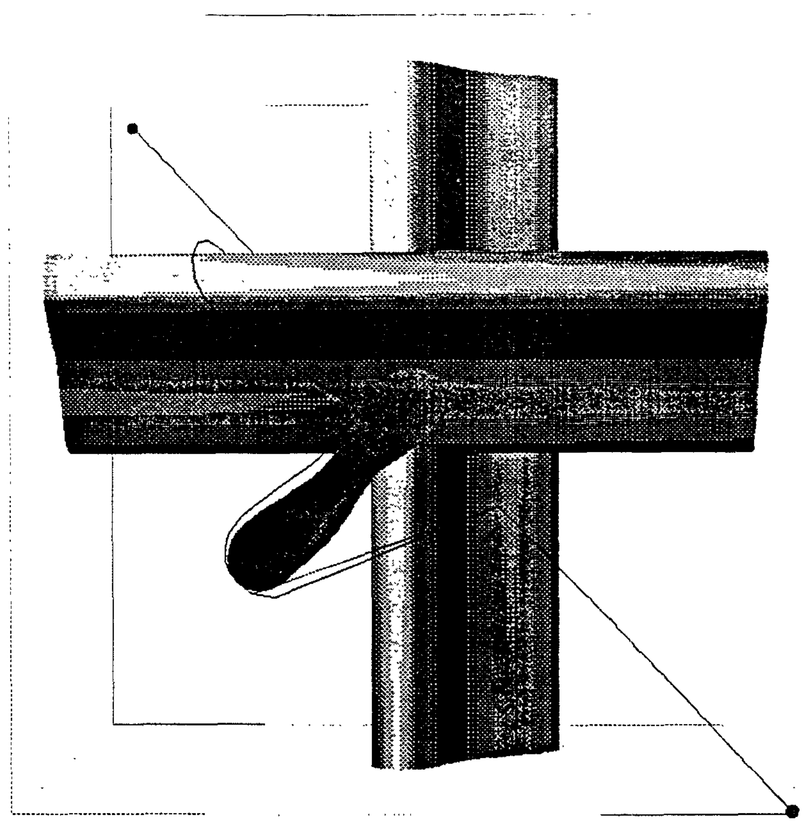


Figure 5.10: Augmentation of Projected Path Ensures Feasiblilty

to check for "line of sight" visibility of points further in the path. This is accomplished by checking the segment joining each pair of points for intersections with the obstacle O'. If no intersections are found, the point is "visible", and we may proceed directly from the first point to the last visible point in the path. We perform this operation for a point on the list, and then use the last visible point from the previous sequence of visiblity calculations as the beginning point for the next sequence. This continues until the entire path has been shortened as much as possible. The resulting path has the properties that it is:

1. *Feasible* The endpoint projection and subsequent augmentation ensure that there are no intersections with the obstacle surface.

2. *Short* The projection against the optimal path and O'' ensure that the path is as close to the optimal path as is practical, and then the subsequent shortening process gets rid of unnecessary "kinks" in the path produced by the projection. Figure 5.11 shows the final shortened path in comparison with the augmented path from Figure 5.10. The improvement in the path is obvious.

5.2.2.6 Euclidean Suboptimality of the Generated Path

The question naturally arises as to whether we could do better. The answer is that, in some cases, we could. Most portions of the path we produce are very good, but there is an artifact of the projection process which causes

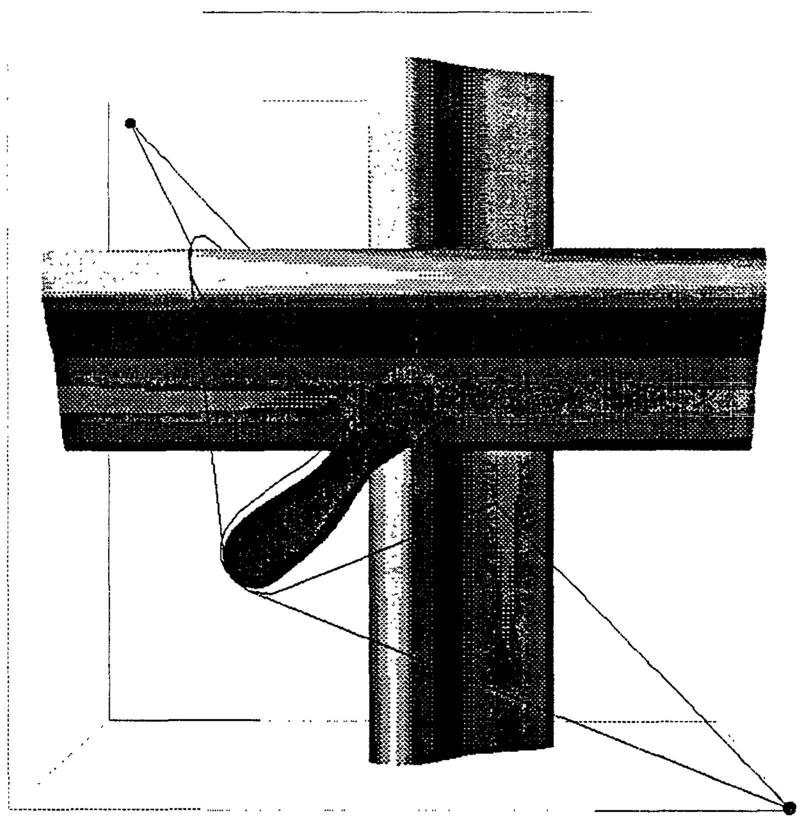


Figure 5.11: Shortening the Augmented Path

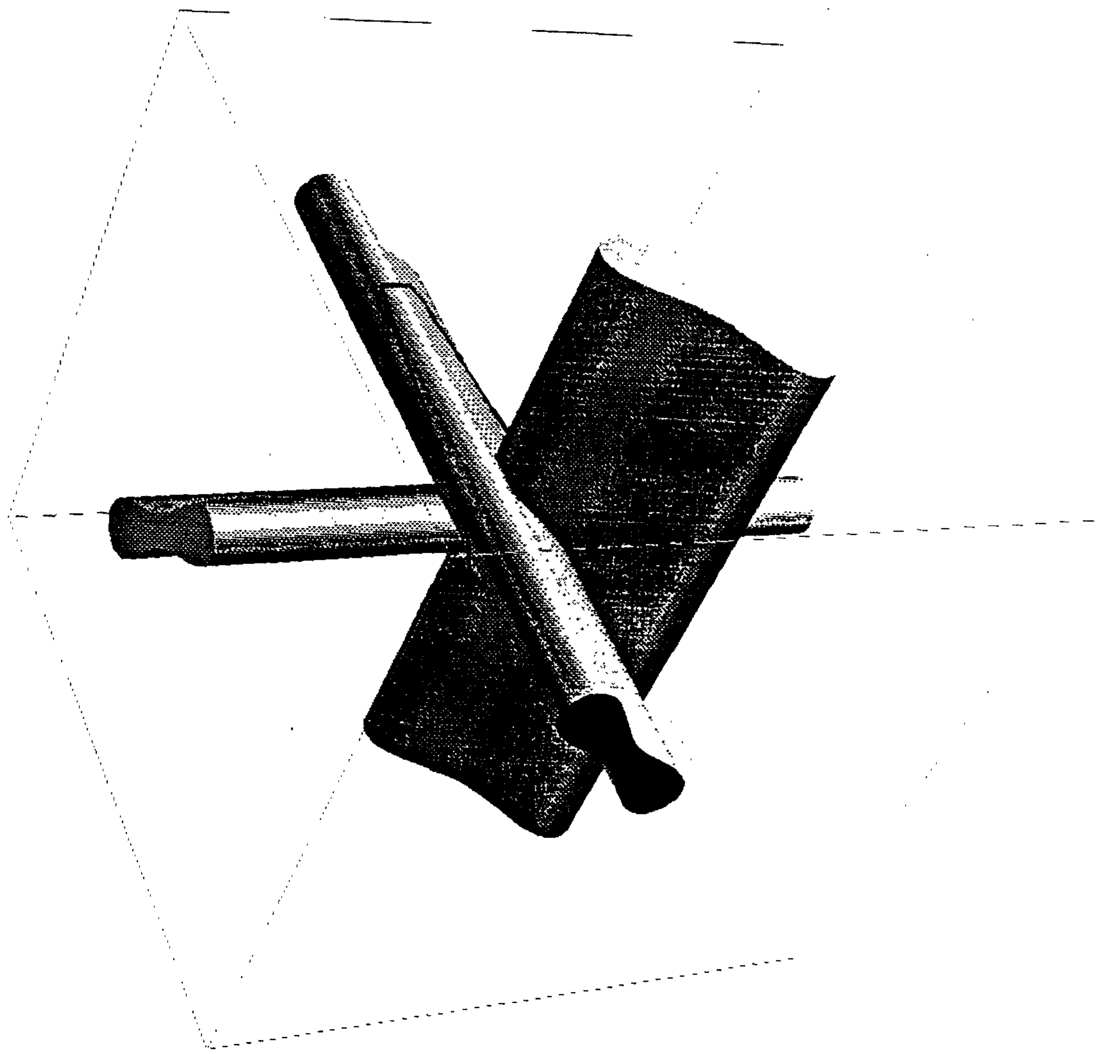


Figure 5.12: Suboptimality of the Generated Paths

the path to be suboptimal, in the sense that its Euclidean length is not the minimum over all feaslible paths. The problem is that the projection originates along cube edges. If the optimal path has a large component along the obstacle axis parallel to an edge of the feasible path, it is likely that a large segment of the projected path will run along the "ridge" of the obstacle. It is clear that a path which angles up to a ridge, runs along it, then runs down away from the ridge at an angle could be replaced by one of constant angle with respect to the ridge, which crosses it at only one point. Figure 5.12 shows an extreme example of this phenomenon, from an example different than that we have been thus far using. This example is the path which connects the two triple points of the obstacle, namely where the surfaces of all three pairwise obstacles have value zero. This represents a clear case of suboptimality, as measured by ratio of the generated path Euclidean length to that of the optimal path.

The problem could be fixed by instituting a rubberbanding algorithm, which would attempt to collapse the endpoints of each segment while checking for feasibility with each contraction step. This is an interesting problem, which we hope will be added in the future to ensure that optimal paths are always generated. It needs to be mentioned, however, that the dynamics of the MDH make the effects of the suboptimality of the generated paths extremely small. Since the fingers can accelerate to the maximal speed almost instantaneously, the limiting factor of any segment traversal is the length of the longest component of the segment, not its Euclidean length. Since this is the case, we find that extra traversal along the obstacle surface always corresponds to travel in a direction which is necessary. As long as the component of the path in this direction is large compared to the orthogonal components, the preformance of the MDH is hardly affected by the suboptimality of the path. This condition, of course, is exactly that which causes our "ridge running" problem in the first place! Thus our path generation algorithm provides extremely good performance when compared with the optimal Euclidean path. This suggests that the metric by which optimality should be measured may not be the standard Euclidean metric, but rather an l_∞ metric. However since we do not possess an easy way of producing l_∞-optimal paths, we shall forego making any claims about the l_∞-optimality of our generated paths. Since our path generation algorithm produces paths more quickly than a quadratically converging relaxation method (for example), we believe that our method has much to recommend it.

5.2.3 Obstacle Avoidance

It is worthwhile to note that our algorithm works equally well for avoidance of obstacles in the workspace of the hand. That is, if we have an object in the hand's workspace, and we wish to switch configurations, it is possible to define the object boundary in configuration coordinates and place it as an additional obstacle in the configuration space. Our algorithm works independently of the number or shape of the obstacles, and so would yield collision-free paths where they exist. The problem is that if an object lies in the path of a fingertip, it essentially divides the configuration space into two regions, one accessible and one inaccessible. If the object lies in the path of all three fingers (such as would an object we had just released from a grasp), it essentially provides only one reachable octant. Finding paths in such a region is neither particularly difficult, nor particularly useful, and so we did not pursue it. For the purpose of using the MDH for grasping, it will typically be necessary to move the hand away from the object until its configuration space is free of obstacles (other than the self-collision obstacle). The fingers may then be driven into the pregrasp configuration for the next hand, and the hand returned to the vicinity of the object. However, if this capability is desired in the future, the existing code can be used with only trivial modifications.

Chapter 6

Conclusion

We have attempted to show a complete methodology for applying the MDH to the task of grasping an object. We have presented a method for determining what the grasp locations should be, given a geometrical description of the object, and shown that the grasps we selected are likely to provide a firm grasp. We implemented this algorithm for some interesting cases and showed that it does provide good grasps. Given the grasps selected, we have shown by a stability analysis, how to answer the question of whether a grasp can withstand task specific perturbing wrenches. We have then shown how the joint angles of the pregrasp configuration can be attained via the inverse kinematics of the hand, and how the fingers can be moved to this configuration in a collision-free manner. Thus all of the information needed to grasp an object and to determine if the grasp is firm has been provided.

Needless to say, however, there is much interesting work that remains to be done. We have indicated that the addition of polygon convex decomposition routines would allow our MIC grasp selection algorithm to be extended to truly arbitrary boundary shapes, and how the MIC routine could be used to automate the grasp plane selection. We have also stopped short of an exhaustive analysis of three-dimensional stability, giving conditions which are sufficient for three-dimensional stability, but not answered whether they are necessary. In path planning, we have indicated that further additions could be made to ensure Euclidean optimality of the generated path. So there remains a good deal of fruitful work to be done. We hope to have provided a complete baseline system for grasping, so that future researchers may focus more intensively on specific areas of the grasp problem. One could make an entire dissertation on an in-depth analysis of stability for example. We hope to provide such a future researcher with the complement of the grasp knowledge base. That is, an experimenter could investigate stability issues, without the requirement to determine how a grasp is selected, or how the grasp is actually achieved. As all work in a technical field stands on the shoulders of the work of earlier researchers, so, too, do we hope to provide future researchers in this field with somewhat higher shoulders to stand on.

Bibliography

[1] X. Markenscoff, N. Luqun, and C. Papadimitriou. The geometry of grasping. *The International Journal of Robotics Research*, 9(1):61–74, February 1990.

[2] D. J. Montana. The kinematics of contact and grasp. *The International Journal of Robotics Research*, 7(3):17–32, 1988.

[3] Z. Li, P. Hsu, and S. Sastry. Grasping and coordinated manipulation by a multifingered robot hand. *The International Journal of Robotics Research*, 8(4):33–504, August 1989.

[4] K. B. Shimoga and A. A. Goldenberg. Grasp admittance center: A concept and its implications. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 293–298, 1991.

[5] J. Kerr and B. Roth. Analysis of multifingerd hands. *The International Journal of Robotics Research*, 4(4):3–17, Winter 1986.

[6] B. S. Baker, S. Fortune, and E. Grosse. Stable prehension with a multifingered hand. In *Proceedings of the 1985 IEEE International*

Conference on Robotics and Automation, pages 570–575, 1985.

[7] B. Mishra, J. T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grasps. *Algorithmica*, 1987.

[8] B. Mishra and N. Silver. Some discussion of static gripping and its stability. *IEEE Transactions on Systems, Man and Cybernetics*, 19(4):783–796, August 1989.

[9] V. Nguyen. Constructing stable grasps. *The International Journal of Robotics Research*, 8(1):26–37, February 1989.

[10] R. A. Grupen and R. S. Weiss. Force domain models for multifingered grasp control. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 418–423, 1991.

[11] D. J. Montana. The condition for contact grasp stability. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 412–417, 1991.

[12] G. Guo and W. A. Gruver. Fingertip force planning for multifingered robot hands. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 665–6725, 1991.

[13] Y. Chen, I. D. Walker, and J. B. Cheatham. A new approach to force distribution and planning for multifingered grasps of solid objects. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 890–895, 1991.

[14] M. R. Cutkosky and P. K. Wright. Friction, stability and the design of robotic fingers. *The International Journal of Robotics Research*, 5(4):20–37, Winter 1986.

[15] Loncaric J., deComarmond F., Bartusek J., Pati Y. C., Tsakiris D., and R. Yang. Modular dextrous hand. Technical Report SRC TR 89-31, Systems Research Center, University of Maryland, 1989.

[16] F. Reuleaux. *Theoretishe Kinematik.* Dover, 1875.

[17] B. Roth. Screws, motors, and wrenches that cannot be bought in a hardware store. In M. Brady and R. Paul, editors, *First International Symposium on Robotics Research*, pages 679–693. MIT Press, 1984.

[18] R. C. Melville. An implementation study of two algorithms for the minimum spanning circle problem. In *Computational Geometry*, pages 267–295. Elsevier Science Publishers B.V. (North-Holland), 1985.

[19] D. H. Greene. The decomposition of polygons into convex parts. *Advances in Computing Research*, 1:235–259, 1983.

[20] J. Loncaric. Kinematics of grasping with the modular dextrous hand. Technical Report SRC TR 89-32, Systems Research Center, University of Maryland, 1989.

[21] H. Moseley. *Illustrations of Mechanics.* London, 1839.

[22] T. Yoshikawa. *Foundations of Robotics.* The MIT Press, 1990.

[23] M. W. Spong and M. Vidyasagar. *Robot Dynamics and Control.* John Wiley and Sons, 1989.