

# Chamferless Assembly of Rectangular Parts in Two and Three Dimensions

by

Michael E. Caine

B.S.M.E., Tufts University  
(1982)

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

Master Of Science  
In Mechanical Engineering

at the

Massachusetts Institute Of Technology  
June 1985

©Massachusetts Institute of Technology, 1985

Signature of Author \_\_\_\_\_

Department of Mechanical Engineering  
May 10, 1985

Certified by \_\_\_\_\_

Tomás Lozano-Pérez  
Thesis Supervisor

Certified by \_\_\_\_\_

Warren P. Seering  
Thesis Supervisor

Accepted by \_\_\_\_\_

Ain A. Sonin  
Chairman, Departmental Graduate Committee

MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY

JUL 22 1985

LIBRARIES

Archives

# Chamferless Assembly of Rectangular Parts in Two and Three Dimensions

by

Michael E. Caine

Submitted to the Department of Mechanical Engineering on May 10, 1985 in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering.

## Abstract

Automation of the assembly process requires parts to be assembled in the presence of uncertainty. This uncertainty results from an imperfect knowledge of the system being assembled as well as the performance limitations of the devices performing the assembly.

This thesis develops models that describe the assembly of three dimensional rectangular parts in the presence of friction and presents techniques for analyzing and evaluating these models. In particular, a means of determining the constraints on the applied forces and moments that may act on an assembly during compliant motion is presented. In addition, a means of visually representing these constraints is provided.

A series of reliable initial conditions are examined and a set of heuristics developed to aid in the development and evaluation of robust assembly strategies. Finally, compliant assembly strategies are developed and implemented that succeed in inserting a rectangular peg into a rectangular hole, in both two and three dimensions, in the presence of significant positional uncertainty.

Thesis Supervisors: Tomás Lozano-Pérez  
Associate Professor of Electrical Engineering  
and Computer Science

Warren P. Seering  
Associate Professor of Mechanical Engineering

## Acknowledgments

I would like to express special thanks to my thesis supervisors, Tomás Lozano-Perez and Warren Seering, for their advice, support, and constant encouragement. All of the ideas presented in this thesis originated in discussions with Tomás and Warren, whose enthusiasm for and insight into the subject were a constant source of inspiration. This work would have been impossible without their guidance.

I would like to thank many friends at the A. I. Lab whose help and humor made my stay here such an enriching and enjoyable experience. In particular, I would like to thank the members of Prof. Seering's research group: Mike Benjamin, Gary Drlick, Steve Eppinger, Dan Frost, Alfonso Garcia, Steve Gordon, Fred Martin, Peter Meckl, Ken Pasch, Robert Podoloff, Neil Singer, Sara Tabler, Erik Vaaler; and Allen Ward for their advice and friendship. Thanks to Jeff Abramowitz, Dave Brock, Steve Buckley, John Canny, Steve Chiu, Bruce Donald, Mike Erdmann, Joe Jones, Dr. Ken Salisbury, Ron Wiken, and all of the members of the A. I. community. In addition, I would like to thank Prof. Patrick Winston, Karen Prendergast, and the staff of the Artificial Intelligence Laboratory for providing such a unique and challenging environment.

Thanks to the General Electric Foundation and the System Development Foundation for financial support.

Thanks also to the Macsyma Consortium for the capability to manipulate large and complicated equations.

Special thanks to my family, Steve, Brian, Pam, Liam, and Allison, for their love and support.

Finally, thanks to my parents, whose love and support over many years were without bound.

To the memory of my parents  
Philip and Helen Caine

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>                                  | <b>8</b>  |
| 1.1      | Problem Statement . . . . .                          | 8         |
| 1.2      | Brief Outline . . . . .                              | 9         |
| 1.3      | Background . . . . .                                 | 11        |
| 1.3.1    | Uncertainty . . . . .                                | 11        |
| 1.3.2    | Modes of Failure During Assembly . . . . .           | 13        |
| 1.3.3    | Using Compliance to Aid Assembly . . . . .           | 14        |
| 1.3.4    | Motion Planning . . . . .                            | 16        |
| 1.4      | Rectangular Parts . . . . .                          | 18        |
| 1.5      | Outline of The Thesis . . . . .                      | 21        |
| <br>     |  |           |
| <b>2</b> | <b>A Planar Example</b>                              | <b>23</b> |
| 2.1      | Brief Overview . . . . .                             | 23        |
| 2.2      | Some Definitions . . . . .                           | 24        |
| 2.2.1    | The Friction Cone . . . . .                          | 25        |
| 2.2.2    | Jamming and Wedging . . . . .                        | 27        |
| 2.3      | A Proposed Assembly Strategy . . . . .               | 31        |
| 2.3.1    | Selection of Initial Conditions . . . . .            | 31        |
| 2.3.2    | Applied Forces and Moments . . . . .                 | 34        |
| 2.4      | A Model of the 2D Chamferless Peg and Hole . . . . . | 35        |
| 2.4.1    | Assumptions of Model . . . . .                       | 35        |
| 2.4.2    | Possible Configurations During Assembly . . . . .    | 36        |
| 2.4.3    | Geometry of the Chamferless Case . . . . .           | 39        |
| 2.4.4    | Quasi-static Force Analysis . . . . .                | 42        |
| 2.4.5    | The Applied Force-Moment Constraint Space . . . . .  | 50        |
| 2.4.6    | The Resulting Insertion Strategy . . . . .           | 67        |
| 2.5      | Summary . . . . .                                    | 68        |

|          |   |            |
|----------|---|------------|
| <b>3</b> | <b>Assembly in Three Dimensions</b>                               | <b>69</b>  |
| 3.1      | Brief Overview . . . . .  | 70         |
| 3.2      | Outlining a Strategy . . . . .                                    | 71         |
| 3.2.1    | Identifying Useful Properties . . . . .                           | 72         |
| 3.2.2    | Determining the Set of Possible Contact Cases . . . . .           | 79         |
| 3.2.3    | The Independent State Parameters . . . . .                        | 83         |
| 3.2.4    | Determining Velocity Trajectories Between States . . . . .        | 86         |
| 3.2.5    | Selecting the Proper Forces and Moments . . . . .                 | 38         |
| 3.3      | Modeling and Analysis . . . . .                                   | 89         |
| 3.3.1    | Geometric Representation of Parts . . . . .                       | 89         |
| 3.3.2    | Selecting a Subset of Configurations . . . . .                    | 92         |
| 3.3.3    | Kinematic Constraints . . . . .                                   | 95         |
| 3.3.4    | Representation of Sliding Constraints . . . . .                   | 102        |
| 3.3.5    | Representing the Assembly State . . . . .                         | 106        |
| 3.3.6    | Quasi-static Equilibrium Equations . . . . .                      | 109        |
| 3.4      | The Applied Force-Moment Solution Space . . . . .                 | 111        |
| 3.4.1    | The 5 Dimensional Force-Moment Space . . . . .                    | 111        |
| 3.4.2    | Making the Solution Space More Tractable . . . . .                | 112        |
| 3.5      | Selecting a Strategy . . . . .                                    | 123        |
| 3.5.1    | Determining State Parameters . . . . .                            | 125        |
| 3.5.2    | Choosing 'Cross-sections' of Higher Dimensional Regions . . . . . | 129        |
| 3.5.3    | Finding Solution Regions . . . . .                                | 130        |
| 3.5.4    | Multi-Step Strategies . . . . .                                   | 135        |
| 3.6      | Evaluation . . . . .  | 140        |
| 3.6.1    | Determining the Effects of Uncertainty . . . . .                  | 140        |
| 3.6.2    | Propagation of Errors Between Discrete States . . . . .           | 142        |
| 3.7      | Summary . . . . .   | 145        |
| <b>4</b> | <b>Implementation</b>   | <b>147</b> |
| 4.1      | 2D Example: A Passive Implementation . . . . .                    | 148        |
| 4.2      | 3D Example: Implementing a Strategy on a Robot . . . . .          | 152        |
| 4.2.1    | The MIT Compliant Motion System . . . . .                         | 152        |
| 4.2.2    | Experimental Setup . . . . .                                      | 154        |
| 4.3      | Results . . . . .   | 158        |
| <b>5</b> | <b>Conclusion</b>   | <b>163</b> |
| 5.1      | Summary . . . . .   | 163        |
| 5.2      | Generality of the Planning Process . . . . .                      | 164        |
| 5.2.1    | Higher Dimensional Constraint Regions . . . . .                   | 165        |
| 5.2.2    | Relationship to More Formal Techniques . . . . .                  | 167        |

|   |     |
|---|-----|
| 5.3 Suggestions for Future Work . . . . . | 168 |
| A Generating the Constraint Surfaces      | 172 |
| B Expressions for Case One                | 233 |
| C Constraint Surfaces                     | 257 |
| References                                | 270 |

# Chapter 1

## Introduction

### 1.1 Problem Statement

Many of today's industrial and consumer products consist of a large number of separate parts that must be assembled as inexpensively as possible. Recently the field of robotics has attracted much attention in the effort to automate the assembly process. In order to successfully perform assembly operations, robots, as well as other assembly machines, must be able to work in the presence of uncertainty. This uncertainty arises from the performance limitations of the robot, its sensors, the parts being assembled, the wear of jigs and fixtures, and the mathematical models of the parts used to generate the assembly strategies.

Developing strategies that guarantee successful assembly in the presence of uncertainty is an important problem in automated assembly. In order to generate these strategies it is necessary to understand and model those factors that significantly affect parts during assembly. This thesis will develop models for use in planning assembly and techniques to use these models to develop reliable assembly strategies in three dimensions.



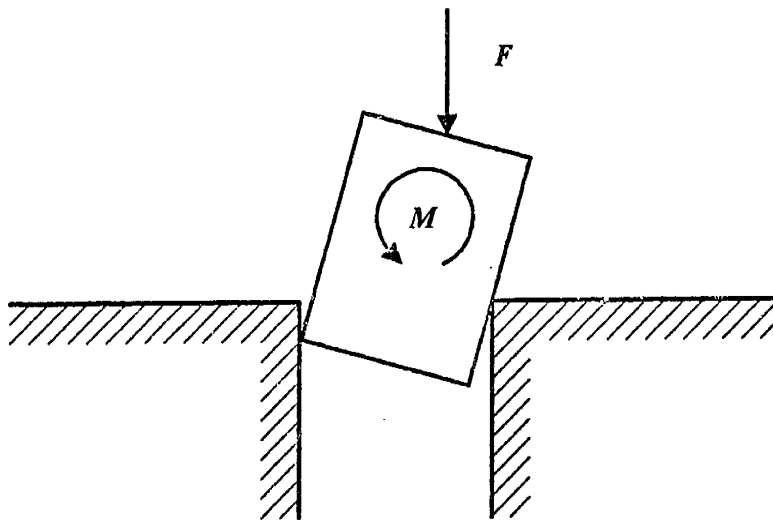


Figure 1.1: Peg and Hole

## 1.2 Brief Outline

Often robots are called upon to perform assembly in circumstances where their positional uncertainty is greater than the dimensional tolerances of the parts to be assembled. This means that a robot operating under position control alone would be unreliable. By controlling the forces applied to parts during assembly and allowing their positions to be determined by the geometric constraints, it is possible to reduce the overall positional precision required.

The basic approach of this thesis is to model the reactions between parts in contact. By determining those reactions that will cause an assembly to prematurely terminate or otherwise fail, the constraints on the applied forces and moments that these reactions represent may be determined. One possible assembly strategy, then, would be to exert a set of forces and moments that satisfy all of these constraints.

Figure 1.1 shows a classic example used when discussing assembly planning, the planar peg and hole. If we apply a force and a moment to the peg while it is in contact with the hole then the peg will either stick in its initial position or move. If the peg is stuck then the hole is exerting a force and moment on

the peg equal and opposite to those being applied. If the peg moves, on the other hand, the applied force and moment are exceeding the ability of the hole to oppose them, resulting in a net force and moment acting on the peg. The determination of what forces and moments will cause sticking or motion, and what the resulting motion would be, will be dealt with in the following chapters.

In order to determine the precise nature and location of the reaction forces, geometric models of the parts will be developed. Since actual assemblies consist of parts that are three-dimensional, the models developed must also be three-dimensional. Often, the geometry of actual parts can be quite complex. In addition, the modeling and analysis of these parts in three dimensions will require a significant amount of computation. In order to make the problem more tractable, we will limit the geometries considered to rectangular parts. We shall see later that many geometrically complex parts may be represented as collections of simple rectangles.

From the geometric models, we shall derive equations that describe the constraints on the applied forces and moments. In order to visualize and better understand these constraints, we shall represent them graphically in a space whose dimensions are the components of the applied force and moment. This force-moment space will serve as a convenient domain in which to develop our strategies.

To develop our strategies, we shall adopt a design approach to assembly planning. Namely, the models and equations used to describe the assembly process will not in themselves be sufficient to develop a strategy. In addition, a number of assumptions and decisions will have to be made at various stages throughout the development process. Often, as more information becomes available, some of the earlier assumptions will have to be revised. In order to aid the designer in making these decisions, the means to evaluate and compare intermediate results will also be developed. A final assembly strategy, then, will often be the result of a number of iterations.

Finally, the means of implementing some of the resulting strategies will be examined. Two examples of implementation, one passive, the other using a robot, will be presented and their performance compared.

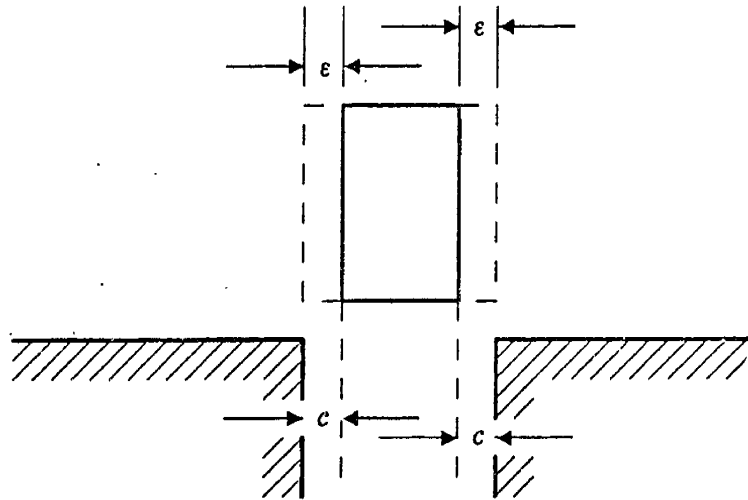


Figure 1.2: A Geometric Representation of Positional Uncertainty

## 1.3 Background

### 1.3.1 Uncertainty

Figure 1.2 illustrates the effect that positional uncertainty can have on performing an assembly. Here a peg has been positioned over a hole which is larger than the peg by the clearance  $2c$ . We will assume that the peg is perfectly vertical and has been centered with the hole to within some uncertainty  $2\epsilon$ . If we represent this uncertainty by increasing the size of the peg  $2\epsilon$ , as shown, we can see that it will be impossible to guarantee the peg will enter the hole under position control unless the uncertainty  $\epsilon$  is less than the clearance  $c$ .

Uncertainty enters into other aspects of assembly planning as well. Aside from the uncertainty associated with specifying the position of a part, there is also uncertainty associated with the positions of other parts in the assembly. Specifically, as parts are placed in an assembly, they become part of that assembly. Other parts, in later assembly operations, may be required to interact with these previous parts. Since each of these parts has associated with it some uncertainty, the overall uncertainty of the system will tend to increase with each

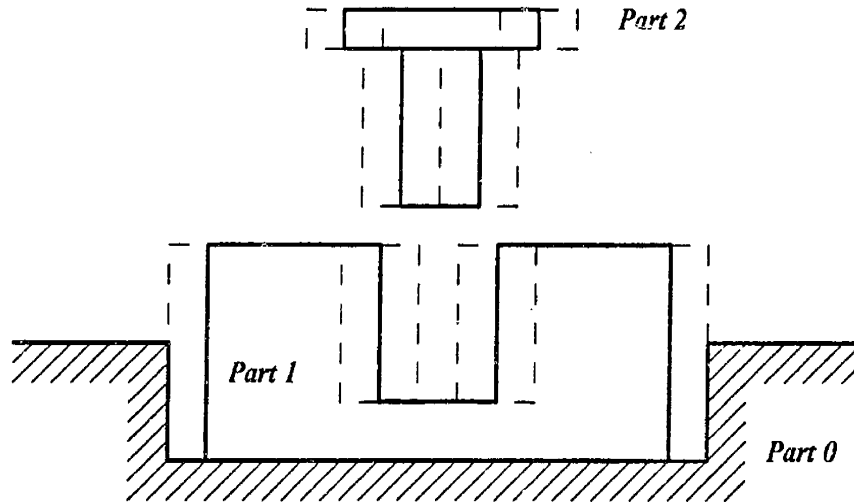


Figure 1.3: Compounded Uncertainty

assembly operation performed. This increase in uncertainty is illustrated in Figure 1.3. Often, for functional reasons, the number of parts that may be ‘stacked’ in this way will be limited in the design of the assembly.<sup>1</sup>

In addition to position, other sensed values such as force are also subject to uncertainty. However, although adding force information introduces additional uncertainty, a strategy that uses one source of information to augment another will in general more reliable [Simunovic 79].

Finally, the parts being assembled will themselves have some uncertainty associated with their dimensions. While all parts have tolerances specified when they are manufactured, occasional variations in those tolerances will affect the reliability of the strategies used to assemble them. For example, no assembly strategy, no matter how reliable, will succeed in assembling parts which do not fit together because of incompatible tolerances.

Because uncertainty invariably arises in every aspect of assembly, strategies must be developed to be as insensitive to it as possible. This requires that uncertainty be considered in every phase of the assembly planning process [Erd-

---

<sup>1</sup>See [Brooks 82].

mann 84]. Often the models used to develop these strategies will introduce uncertainties arising from the assumptions and simplifications made in their development. These model based uncertainties differ from the system based uncertainties in that they are not inherent to the system being studied, giving us some control over their magnitude. We may therefore reduce the effects of model based uncertainty by carefully choosing our models and making fewer simplifying assumptions.

### 1.3.2 Modes of Failure During Assembly

To say that an assembly has been successfully completed indicates that it has reached some desired end state within prescribed tolerances. Often, the uncertainty that makes position control strategies unreliable also makes the determination of successful vs unsuccessful motion termination by means of position sensing equally unreliable [Erdmann 84]. For a strategy to guarantee successful completion, then, the various conditions that would prematurely terminate the assembly process must be identified and avoided. The task of the designer of an assembly strategy, then, is to model every type of failure an assembly is likely to encounter.

One mode of failure has already been illustrated in Figure 1.2. If the bottom of the peg contacts one of the top edges of the hole the assembly motion will be terminated. Similarly, if we apply a force and moment to the peg such that the reaction forces imposed by the hole negate them, then motion will also be terminated. This form of motion termination is known as jamming and will be treated in detail in the next chapter [Simunovic 79, Whitney 82].

Another form of failure during assembly involves changing or damaging the parts being assembled in such a way that they will no longer meet the functional or cosmetic specifications set for them. This type of failure could result from applied forces exceeding the ability of the parts to withstand them. In addition, the deformation of parts during assembly can lead to a condition known as wedging which we will also examine in the next chapter [Simunovic 79, Whitney 82].

Let us again imagine the task of inserting a peg in a hole, this time starting

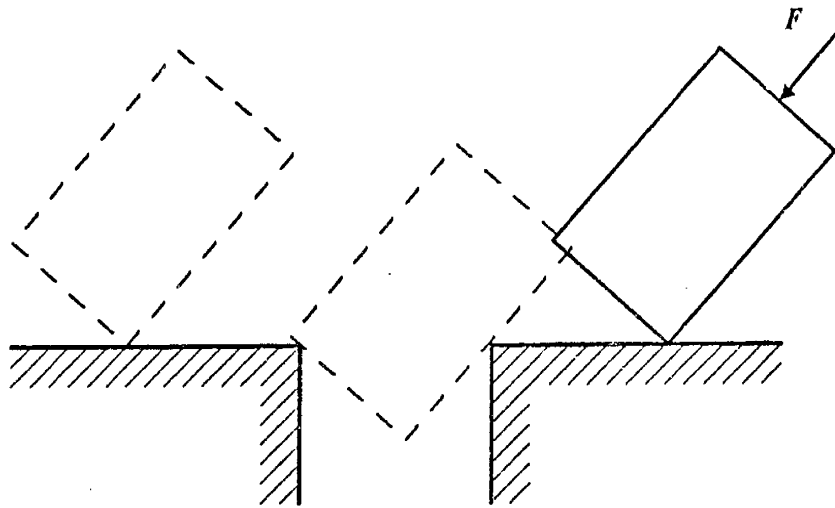


Figure 1.4: Peg Overshooting the Hole

out with the peg to one side of the hole as shown in Figure 1.4. If we apply a force to the peg pointing towards the left such that the peg will not jam, the peg will slide towards the hole (assuming no rotation). When the bottom of the peg crosses the edge of the hole, it may be possible for the peg to continue to slide through two point contact and out across the other side of the hole. In this case the failure results not from the termination of motion, but rather from the improper *transition* of motion from the outside of the hole to the inside. This type of failure may also result from points of contact being broken during assembly or forces and moments being applied that cause motion to occur in the wrong direction.

### 1.3.3 Using Compliance to Aid Assembly

In Figure 1.2 we saw how uncertainty could prevent an assembly strategy using only position control from working. The edges of the hole were taken to be obstacles which had to be overcome for assembly to take place. By controlling the forces and moments applied to the peg, however, the edges of the hole become geometric guides on which the peg may slide towards insertion [Mason 83].

Under force control the positional uncertainty of the system, while still present, does not prevent the peg from entering the hole. The precise location of the surfaces in contact, then, need not be known for a force control strategy to succeed.

Force control is one form of general motion control known as compliance. The basic principle of compliance is to allow the position of a system being assembled to be controlled by the geometric constraints inherent in that system. Various implementations of compliance relate the parameters determining the state of an assembly to the applied forces and moments by means of different control laws. In the case of force control, the control law is simply

$$\begin{aligned}\vec{F}_{commanded} - \vec{F}_{exerted} &= 0 \\ \vec{M}_{commanded} - \vec{M}_{exerted} &= 0\end{aligned}$$

where force and moment sensors could be used the case of feedback force control, or the forces could be commanded open loop by means of motor currents.

Another form of compliance implementation is known as stiffness control. Here the governing control law takes the form of a generalized spring

$$\vec{F} = [K] (\vec{X}_{commanded} - \vec{X}_{actual})$$

where  $\vec{F}$  is a vector of force and moment components,  $[K]$  is a generalized stiffness matrix, and  $\vec{X}$  are position vectors consisting of linear and rotational components. In order to specify a set of forces and moments given the stiffness matrix, a commanded position offset is issued. The differences between these commanded positions and the actual positions constrained by the environment determine the resulting force and moment. It is also possible to make the elements of the matrix  $[K]$  the command variables. Thus, by changing the stiffness of the assembly system as desired, contact forces may be arbitrarily specified [Salisbury 80].

In addition to generalized stiffness, another form of compliance implementation is known as the generalized damper [Whitney 77]. Here, the commanded state variables are velocities. The control law here is of the form

$$\vec{F} = [B] (\vec{V}_{commanded} - \vec{V}_{actual})$$

where  $[B]$  is a generalized damping matrix, and  $\vec{V}$  are velocity vectors consisting of linear and rotational components. The differences between the commanded velocities and the resulting velocities tangent to the surfaces of the parts in contact determine the resulting force and moment.

Both generalized stiffness and generalized damping control can be further generalized as examples of impedance control [Whitney 85]. In both cases, the input is related to force indirectly by means of impedance terms. In some cases, such as following an uneven surface, it may be desirable to control forces in some directions while independently controlling position or velocity in others. In this case a technique known as hybrid control may be used [Raibert and Craig 81]. For a historical background and summary of compliance and its various forms of implementation see [Mason 81] and [Whitney 85].

Whatever its implementation, by introducing force information into the assembly process in addition to position, compliance augments the capabilities of assembly systems in the presence of uncertainty [Simunovic 79]. This information need not be incorporated solely into an active control strategy. In fact, many of the control laws may be implemented using entirely passive devices constructed with physical springs and dampers [Whitney 82].

### 1.3.4 Motion Planning

Many automatic assembly operations performed today have been carefully planned and implemented for one specific set of parts. Often, when a new set of parts are to be assembled, an entirely new assembly strategy must be developed and implemented. The process of developing these strategies often involves the specification of tasks at a very low level, such as moving to a pallet of parts, opening a gripper, moving down over the first part, closing a gripper, lifting the part out of the pallet, etc.. The goal of motion planning is to provide a series of higher level task specifications and have the lower level tasks be specified automatically.

The tasks that must be specified in motion planning may be broken down into two broad categories. The first, involving the specification of collision free



motions between obstacles in a workspace is known as gross motion planning [Lozano-Pérez 76, Donald 84]. For example, the task of moving a manipulator from a fixture to a pallet of parts and back to the fixture while avoiding collisions would be an example of planning gross motion trajectories. The second task involves planning motions that are constrained by contact with the environment, such as parts being assembled, and is known as fine motion planning [Lozano-Pérez, Mason, and Taylor 83]. Compliance, then, is used in developing the assembly strategies for fine motion tasks.

It has been shown that small variations in the geometry of parts can significantly affect the strategies used to assemble them [Lozano-Pérez, Mason, and Taylor 83]. Therefore, in order to perform fine motion planning, models of the parts to be assembled must be developed and used to represent the geometric constraints on the task. In addition, other factors besides geometry that are significant in the assembly process must be identified and modeled. A number of models have been developed to describe the assembly of parts in the presence of friction. Planar models of two-dimensional pegs and holes with chamfers, as well as multiple pegs and multiple holes, have been developed and used to determine the conditions under which these assemblies could fail [Simunovic 79, Whitney 82, Ohwovoriole, Hill, and Roth 80]. In addition, Ohwovoriole and Roth [81] have developed models that describe general three-dimensional parts in contact. In order to make the resulting equations manageable, however, a cylindrical peg and hole was used as the primary example, which eventually reduces to a planar problem. For more complicated parts where planar solutions are not possible, the models available at present tend to be rather unwieldy for use in strategy analysis.

One useful technique for representing the geometric constraints imposed on parts in contact is the *configuration space* [Lozano-Pérez 81]. The configuration space of a part in contact with an assembly contains surfaces which represent the constraints imposed on the degrees of freedom of the part by the assembly. These surfaces represent a convenient and general way to describe the way in which the geometry of an assembly affects the resulting motion, both in terms of the translation and rotation, of rigid parts. In addition, the constraint surfaces

in configuration space exhibit the same properties as the corresponding real space surfaces. Therefore, factors such as friction may be represented in the configuration space domain for use in assembly planning [Erdmann 84].

Lozano-Pérez, Mason, and Taylor [83] proposed a formal approach for the automatic synthesis of fine motion strategies from high level task specifications. Specifically, goal states of an assembly are identified from which pre-goal regions are constructed. These pre-goal regions, known as pre-images, represent those regions from which the desired goal can be reached by a single commanded motion in the presence of uncertainty. These pre-goal regions can then be recursively backchained until the initial state is included in the set. A strategy then results whereby a series of commanded motions from the initial state to the desired goal state are made that are guaranteed to succeed in the presence of uncertainty. In his masters thesis, Erdmann [84] developed means of explicitly constructing these pre-images (under certain assumptions) known as backprojecting. By modeling friction in configuration space and representing the commanded motions in terms of velocities acting under damper dynamics, he succeeded in automatically constructing strategies that simulated the insertion of a two dimensional chamferless peg into a hole in the presence of significant uncertainty.

## 1.4 Rectangular Parts

Figure 1.5 shows a pair of relatively simple parts that are typical of many of the parts assembled into products today. It is not difficult to imagine that the models necessary to completely and faithfully represent each detail of such parts could be rather complicated. Many of these details, however, have more to do with the function of the parts (be it mechanical or cosmetic) rather than their assembly. Often in assembly, we are concerned only with achieving a certain state of one part relative to one or more other parts. This state will involve the interaction between certain surface features of these parts. A model that describes only these features, as well as those that interact during the assembly of the parts, will be sufficiently complete for the purposes of planning assembly.

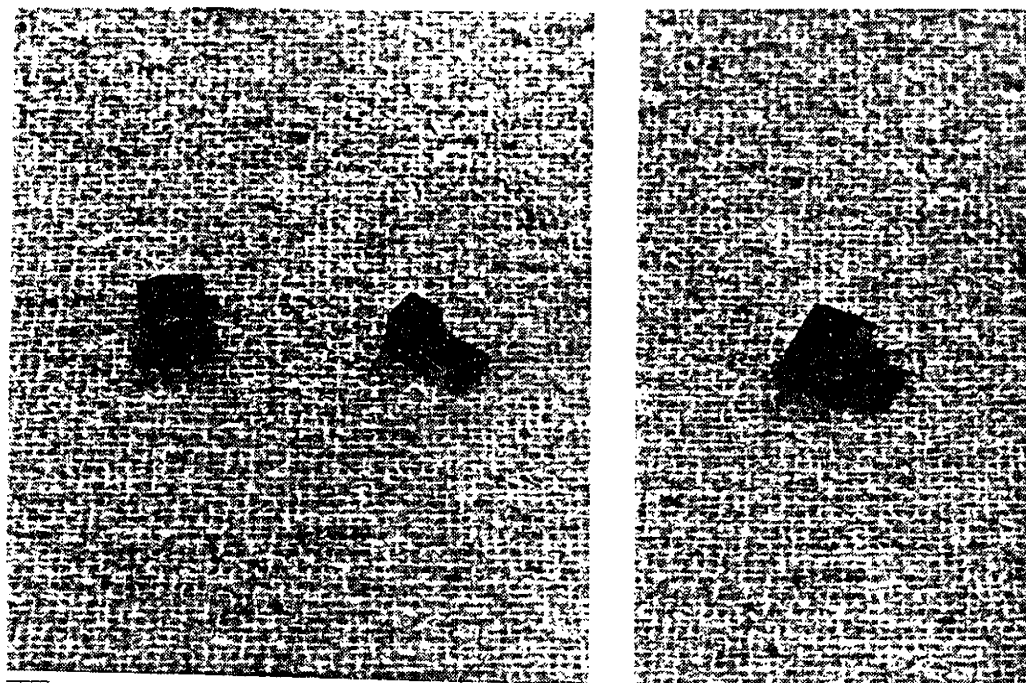


Figure 1.5: A Typical Assembly

In considering only the assembly of parts when modeling them, we can consider an alternative model shown in Figure 1.6. Here the edges and surfaces that do not interact during assembly have been eliminated. Only those geometric features of the parts that are likely to interact with each other (shown as darker lines) remain. We see that for these particular parts, the interacting edges and surfaces can be easily represented as rectangular solids. In fact, many parts may be similarly modeled using collections of rectangles as well as other simple geometric shapes. Figure 1.7 shows a collection of parts taken from some common assemblies. It is easy to see how these parts could be represented by combining a number of simple geometric shapes. For this reason, we shall be considering simple rectangular parts in this thesis when developing models for use in assembly strategy development. Our assumption will be that a large number of parts may be represented by these models using an approach similar to that used for the part of Figure 1.5.<sup>2</sup>

As the standard example of a generic rectangular part, we shall model and

---

<sup>2</sup>We make no assumption here of how parts are to be grasped or fixtured. The only features

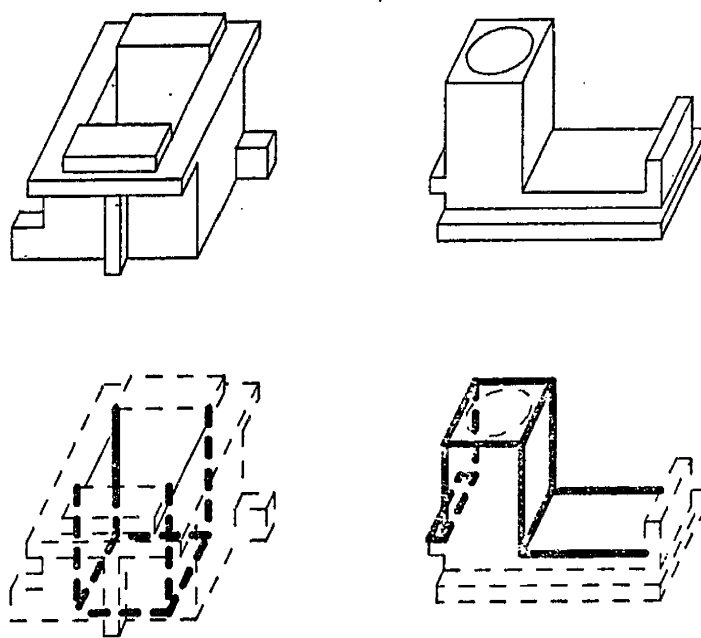


Figure 1.6: A Model for Use in Assembly Planning

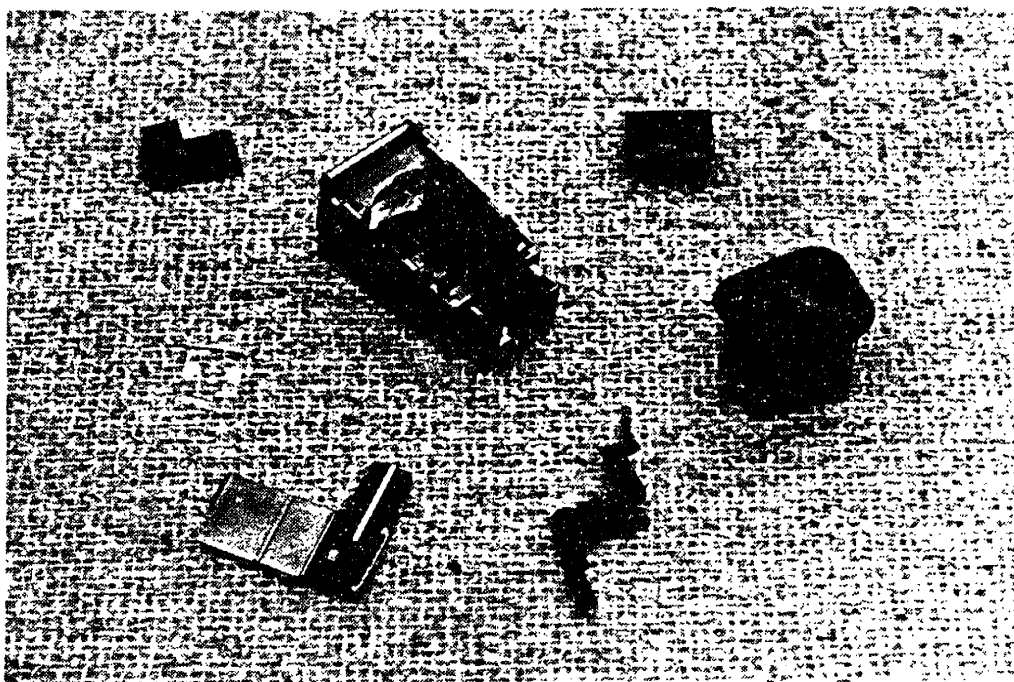


Figure 1.7: Some Common Parts

develop compliant assembly strategies to insert a rectangular peg into a rectangular hole. Since our aim will be to develop reliable strategies that do not require physical modifications of the parts to be assembled, we shall plan the insertion without the aid of chamfers.

## 1.5 Outline of The Thesis

In Chapter 2 we examine the planar insertion of a chamferless peg into a chamferless hole. We begin the process of modeling assembly in the simplified planar domain and develop techniques to represent constraints on the applied forces and moments that govern the assembly. In particular, the values of these forces and moments that will cause jamming and breaking of contact are determined, and the effects that varying parameters have on the resulting assembly strategies is examined.

---

concerning us are those that affect the interaction of parts in contact.

Chapter 3 extends the modeling of assembly into three dimensions. Heuristics are developed which help plan assembly strategies by identifying useful properties of the assemblies. Techniques are developed to deal with the added complexity of three-dimensional models and the specification of the added parameters necessary. The means to represent the constraints on the applied forces and moments that will guarantee the assembly to succeed are also developed as are the means to visualize these constraints. These models and development techniques are then used to develop compliant strategies for inserting a rectangular peg into a rectangular hole in the presence of significant uncertainty and tight part tolerances. Methods of evaluating the resulting strategies are presented that allow areas of possible improvement to be identified and examined.

Chapter 4 presents the implementation of the assembly strategies developed in Chapters 2 and 3. Specifically, a passive device is presented which performs tight tolerance chamferless insertion in two dimensions with a high degree of reliability. For the three dimensional rectangular peg and hole, the implementation of the insertion strategy by means of a robot operating under force control is presented.

Finally, in Chapter 5, the modeling and strategy development techniques outlined in this thesis are reviewed and their generality compared to more formal approaches towards assembly planning.

# Chapter 2

## A Planar Example

In order to determine the nature of the analyses necessary for planning assembly strategies in three dimensions, we will first examine a simpler subset of the general problem, namely, assembly in two dimensions. Our goal here will be to develop modeling and analysis techniques within this simplified domain that can later be extended into three dimensions. We shall identify a number of factors that affect the success of an assembly and develop ways to incorporate these factors into our models. In addition, we shall begin to examine some of the assumptions that will be necessary to make such problems more tractable, and the effects that these assumptions will have on our resulting conclusions.

We shall use as our primary example the two dimensional chamferless peg and hole, also known as the tab and slot. This example is a direct subclass of the three dimensional rectangular parts described in the last chapter. With it we shall perform the necessary mathematical analyses and plan assembly strategies that will succeed in inserting the peg into the hole in the presence of significant positional uncertainty and tight part tolerances.

### 2.1 Brief Overview

As we mentioned in the first chapter, one of the most significant challenges facing designers of assembly systems is the development of devices and strategies that can perform assembly reliably in the presence of uncertainty. To meet these

challenges it will be necessary to gain a better understanding of the behavior of parts as they are being assembled and the factors that affect that behavior. We will begin by identifying and modeling within the two dimensional domain some of these factors that we consider to be most important in terms of guaranteeing successful assembly.

We shall review the modes of failure mentioned in the first chapter in terms of jamming and wedging in the presence of Coulomb friction using the concept of the friction cone [Simunovic 79]. As our control variables we shall assume a generalized applied force and moment applied to the peg. Using the Coulomb friction model of jamming we shall represent the modes of failure of the assembly in terms of limits on the applied forces and moments. We shall thereby be able to identify those forces and moments that will guarantee a successful assembly. In addition we shall identify initial conditions that are robust and easy to obtain given a significant amount of positioning uncertainty in our system.

The constraints on the applied forces and moments will be represented graphically in a force-moment solution space. Regions in the solution space bounded by these constraint curves will represent those forces and moments that will not jam or otherwise terminate the assembly process. This solution space will serve as a valuable tool with which to visualize the effects of various factors on the assembly in terms of their effects on these solution regions. By intersecting solution regions within the force-moment space that represent different configurations of the peg and hole system, values of the applied forces and moments that can reliably slide during transitions between these configurations will be identified.

Finally, the sensitivity of the strategies developed from the force-moment solution space will be examined in terms of variations in the parameters of the parts being assembled.

## 2.2 Some Definitions

To begin our analysis of two dimensional assembly we shall introduce a few useful concepts. Many of the definitions mentioned here are the result of earlier



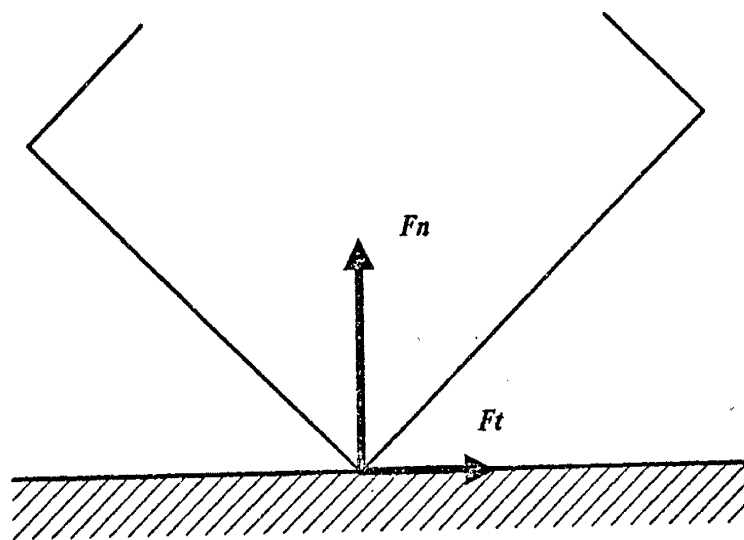


Figure 2.1: Single Point Contact Between Two Objects

work done in the modeling of assembly processes.<sup>1</sup>

### 2.2.1 The Friction Cone

Consider an object in single point contact with a surface as shown in Figure 2.1. The surface exerts a reaction force on the object which can be broken into components that are normal and tangent to the surface, labeled  $f_n$  and  $f_t$  respectively. If we assume that the contact with the surface is governed by dry Coulomb friction and has a static coefficient of friction  $\mu$ , then the maximum tangential component of the reaction force that can be exerted is given by

$$f_{t, \text{maximum}} = \mu f_n \quad (2.1)$$

This maximum value of the tangential component of the reaction force can also be represented graphically as shown in Figure 2.2. This graphical representation of frictional reaction forces is known as the *friction cone*, where the half-angle

<sup>1</sup>See [Simunovic 79], [Whitney 82], and [Ohwovoriole, Hill and Roth 80].

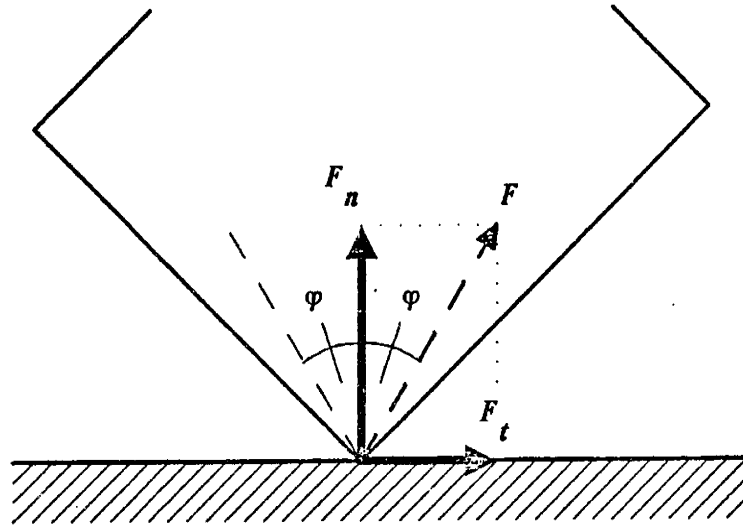


Figure 2.2: The Two Dimensional Friction Cone

$\phi$  of the cone is given by the relation

$$\phi = \arctan(\mu) \quad (2.2)$$

and the perpendicular bisector of the cone is perpendicular to the surface in contact. Another way to interpret the friction cone is to consider an applied force  $F$  as shown in Figure 2.3. If the force points into the friction cone, then no matter what magnitude this force has, the tangential component of the reaction force due to friction will cancel the tangential component of the applied force. If the force lies outside the friction cone however, the maximum frictional force will negate only a portion of the applied force, leaving a net applied force  $F_{net}$  parallel to the surface as shown. Therefore if an object in one point contact with a surface exerts a force that lies within the friction cone, it will be stuck and unable to move. If the force the object applies lies outside the friction cone, the remaining net applied force will cause the object to move and accelerate along the surface. The interior of the friction cone, then, represents the range of possible reaction forces one object or surface can exert on another in one point contact in the presence of friction.

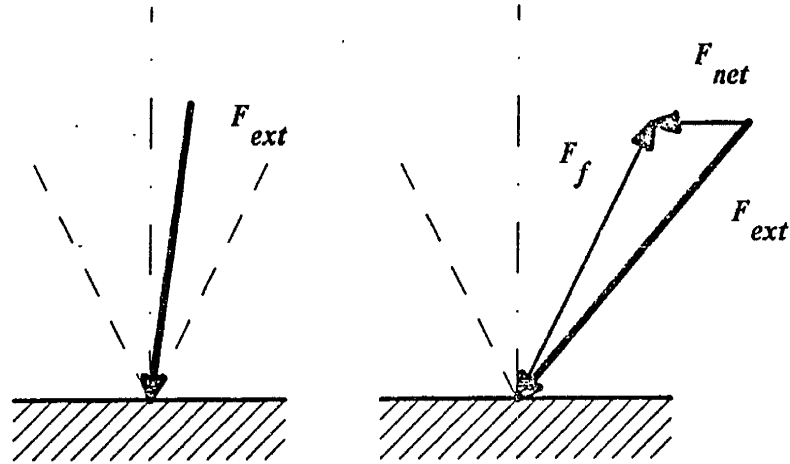


Figure 2.3: Sticking and Sliding

We shall use the friction cone both here and in later chapters to provide a visual way of interpreting the force interactions possible between parts during assembly. For cases where there is more than one point of contact, the friction cone shall be used to visualize the ranges of possible reaction forces at each point.

### 2.2.2 Jamming and Wedging

Using the concept of the friction cone we may now formalize our definitions of the modes of failure during assembly mentioned in the first chapter. Figure 2.4 shows the peg in two point contact with the sides of the hole as well as the associated friction cones at both contact points. If we apply a force and or moment to the peg, a certain set of reaction forces applied by the hole on the peg will arise. These forces must, by definition, lie within the shaded areas defined by the friction cones.

The question of whether or not the peg will slide under these conditions becomes one of whether the reaction forces are in the interior of the friction

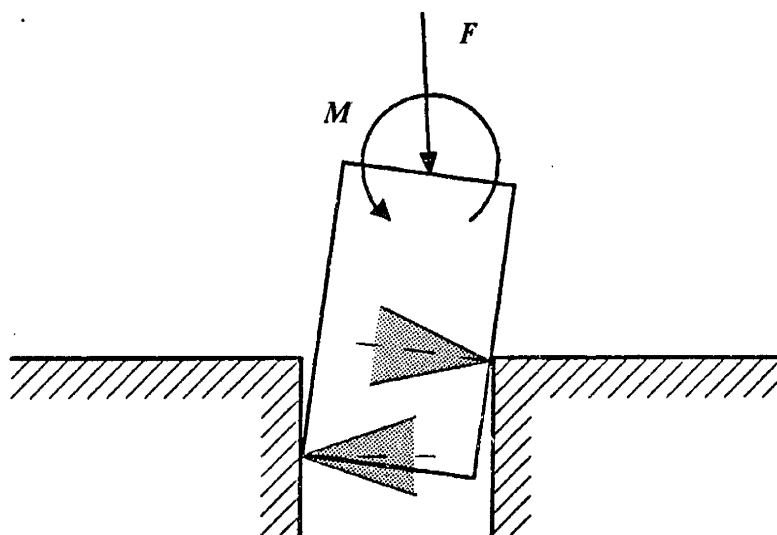


Figure 2.4: Jamming

cones or on their boundaries. If the reaction forces are inside the friction cones then we say the peg is jammed in the hole. No motion between the peg and hole is possible unless the applied force and or moment is changed. If, however, both reaction forces lie on the edges of their respective friction cones, the peg *may* be able to slide. We say *may* because we have not determined if there are any net applied forces remaining at the points of contact. If there are net applied forces remaining, parallel to the surfaces of contact, then the points of contact will slide and accelerate in the directions of those forces. If the applied force and or moment is applied in such a way that the reaction forces lie on the edges of the friction cones and there are no net forces remaining, then we say that the peg is in a state of *impending motion*. Impending motion means that, like the jammed state, the peg is not moving and hence in static equilibrium. However, unlike the jammed state, the peg is on the verge of moving. If the applied forces and or moments were changed just slightly towards disequilibrium, then net forces would appear and the peg would start to accelerate.<sup>2</sup>

Figure 2.5 shows the peg again in two point contact with the sides of the hole. This figure is exactly the same as the previous with the exception that the peg is not as deep in the hole as before. We also notice that the friction cones are in slightly different positions and orientations relative to one another. In particular, the origin of each friction cone lies within the boundaries of the other. An interesting consequence of this condition is known as *wedging*. Let us assume for the moment that the peg, and possibly the hole as well, are not rigid, but instead have some stiffness  $K$ . If we allow the peg to become jammed in the hole, then some component of the reaction forces on the peg may point towards each other. Since the peg has some finite stiffness, these components of the reaction forces will store energy in the peg by deforming it. This stored energy and the associated forces generating it are separate from those reaction forces that are due solely to the force and moment applied to the peg externally. In other words, if we allow the peg to become wedged in the hole, i.e. jammed with some part of the reaction forces pointing towards each other, then even

---

<sup>2</sup>See [Simunovic 79] and [Whitney 82].

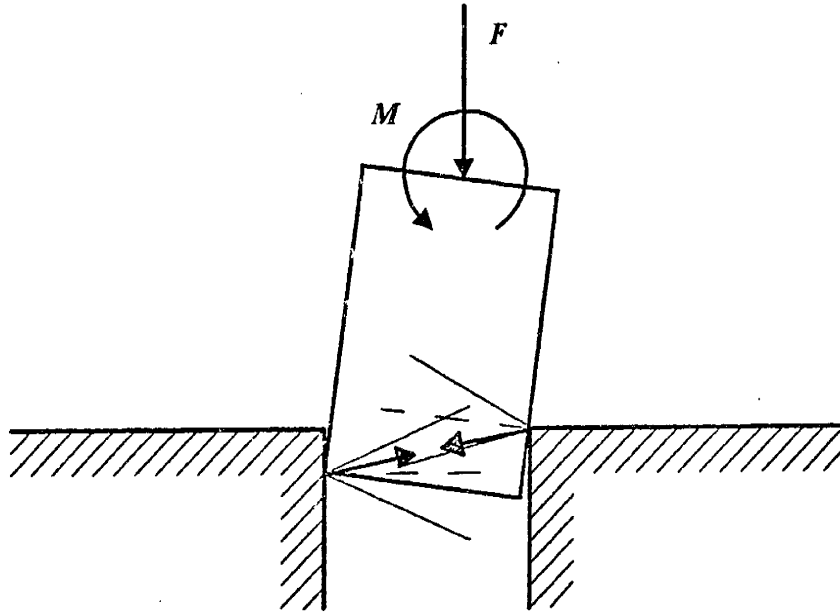


Figure 2.5: Wedging

if we change the applied force and moment, the resultant reaction forces may remain within the friction cones. Specifically, if the components of the reaction forces generated by the deformation of the peg are large in comparison to those generated by the externally applied force and moment, it is possible to end up in a situation where it is impossible to continue the assembly.<sup>3</sup>

The determination of when it is possible to wedge and when it is not depends on the position of the peg and hole in two point contact. Our primary goal in developing strategies for successful assembly will be based on avoiding the jammed state altogether. We shall assume that if we are able to develop strategies that are guaranteed not to result in jamming, then wedging will not be a problem. We shall therefore not specifically deal with the issue of wedging but shall, from time to time, refer to the concept to limit some of our later assumptions.

## 2.3 A Proposed Assembly Strategy

Having established more clearly the conditions under which our assembly could fail, we shall outline a strategy designed to avoid the modes of failure just discussed. In particular, we shall determine a strategy that is guaranteed not to jam the peg during its insertion into the hole. A strategy that is relatively insensitive to variations of such parameters as part tolerances, positioning errors, and slight variations in part geometry will be the most desirable.

### 2.3.1 Selection of Initial Conditions

Every strategy will begin with a set of initial conditions from which to proceed. If these initial conditions are difficult to achieve reliably and the resulting strategy is too sensitive to these conditions, then the overall strategy will not be very robust. For example, we can imagine a strategy for inserting the peg into the hole that requires the axis of the peg to be perfectly aligned with that of the

---

<sup>3</sup>See [Simunovic 79] and [Whitney 82].

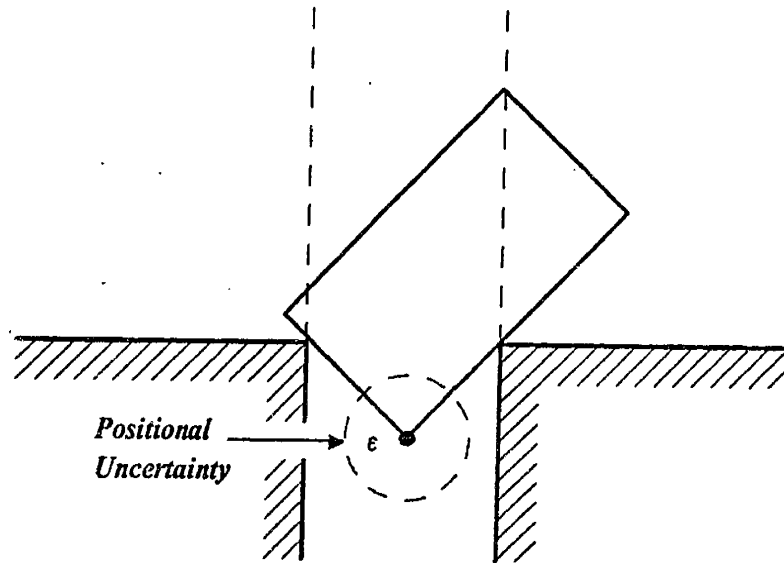


Figure 2.6: A Reliable Initial Condition

hole and the peg positioned within the range of tolerances of the hole.<sup>4</sup> The resulting strategy would then simply be to move the peg down into the hole under position control. If the initial conditions could be achieved reliably and the resulting commanded motion reliably executed, there would be no doubt as to the success of the assembly. This is in fact how many machines, including robots, are presently utilized to execute assembly tasks. For cases where such initial conditions can not be guaranteed, however, the associated strategy will almost surely fail.

In Figure 2.6 we see another proposed initial condition from which to insert the peg into the hole. Here the peg has been tilted relative to the hole such that one corner of the peg is now in the hole. If we require as our initial condition only that one corner of the peg be in the hole, then we have reduced the positioning accuracy required to obtain such a condition. Specifically, if we are able to

---

<sup>4</sup>See Figure 1.2.



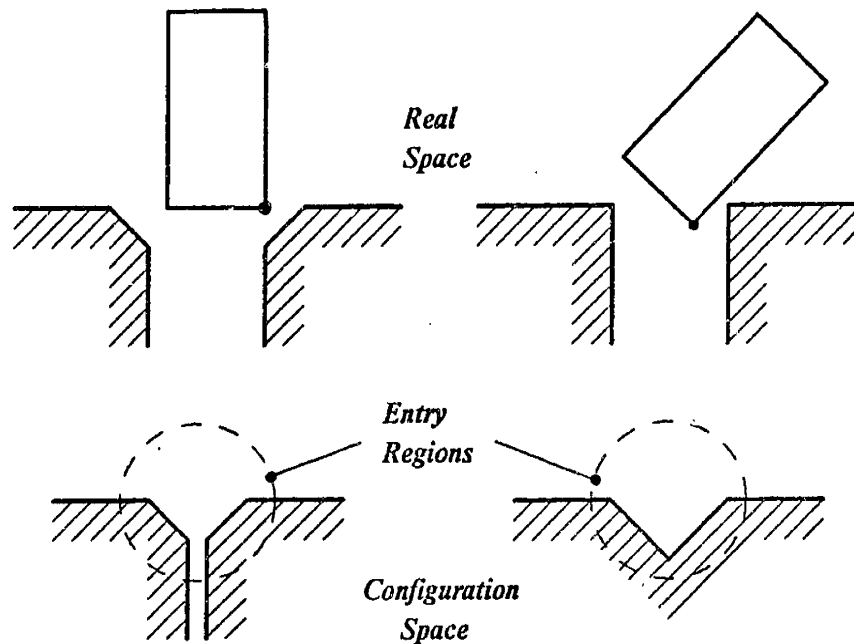


Figure 2.7: Kinematic Equivalent of a Chamfer

position a point within the dimensions of the hole then we can reliably obtain such an initial condition. Because this is such a robust initial condition, we shall choose it as the starting condition for our assembly strategy. In fact, as we shall see later, we will be able to relax even this condition since our resulting strategy will not be very sensitive to it.

Another way to think about tilting the peg into the hole as an initial condition is to consider the function of a chamfer. Figure 2.7 shows two pegs, both with their rotational axes fixed, in contact with a hole. In the first case, the hole is chamfered and the peg is vertical with respect to the hole. In the second case, the hole is unchamfered and the peg is tilted with respect to the hole. The bottom part of the figure shows a slice of the associated *configuration spaces* of the two systems. Namely, if we choose a point on the bottom of each peg and slide the pegs while in contact with the respective holes, keeping the rotational axes fixed, the points will trace out the curves shown. These curves can be

used to represent the kinematic constraints between the peg and the hole in each case. We can see from these curves that as far as kinematic constraints are concerned, the entry conditions for the chamfered hole and the unchamfered hole with tilted peg are equivalent. By choosing to tilt the peg in establishing our initial conditions, then, we are in effect establishing the kinematic equivalent of a chamfer.

### 2.3.2 Applied Forces and Moments

As we outlined earlier, the forces applied to objects in contact play a major role in determining whether the objects will slide or stick. If we choose to control our assembly by means of applied forces, we can directly determine whether these forces will result in jamming. If we choose a position-controlled strategy, for example, the resulting applied forces would be determined by the inherent stiffness of the positioning device and in general would be difficult to control. For the purposes of developing a strategy, then, we shall choose an externally applied force and moment to control the insertion of the peg into the hole. At this point we shall make no assumptions as to the source of these forces and moments.

In using the applied force and moment to guide our assembly, we shall constrain the peg and hole to slide along each other's surfaces. Our constraints on the applied forces and moments, then, will take the form of avoiding jamming and avoiding the breaking of contacts. This second constraint arises from the fact that the jamming constraints will in general depend on the position of the peg relative to the hole. If we allow contacts to be broken, then we will be unable to precisely determine in what position the peg shall be when contacts are resumed. By following surfaces and restricting the breaking of contacts we shall be taking advantage of the geometry of our system to guide the assembly.<sup>5</sup>

---

<sup>5</sup>See Section 1.3.3.

## 2.4 A Model of the 2D Chamferless Peg and Hole

We now begin the process of developing the mathematical models with which we shall generate our applied force-moment jamming and breaking contact constraints. As we develop these models we shall be making a number of assumptions. The limitations these assumptions place on our models and resulting strategies will therefore also have to be examined. Our aim will be to reduce these model-based uncertainties wherever possible while maintaining the tractability of the problem. We will at the same time attempt to avoid assumptions and approximations that would preclude the extension of our models to three dimensions.

### 2.4.1 Assumptions of Model

As we recall from Section 2.2.2, the three possible conditions resulting from a certain applied force and moment on the peg in two point contact with the hole, excluding wedging or breaking contact, are jamming, sliding, and impending motion. In the case of impending motion, the peg was considered to be on the verge of moving while still maintaining static equilibrium. This impending motion state represents the boundary between jamming and sliding. In terms of representing the jamming constraints on applied forces and moments, then, we shall examine this boundary state in detail. In particular, our model of the peg and hole system will be developed under the assumption of *quasi-static equilibrium*. By quasi-static we mean that the force and moment terms introduced by motion, i.e. damping and inertia, will be assumed negligible. Therefore the only terms that will appear in our governing equations will be static force and moment terms. By making our quasi-static assumptions we are not restricting the peg from moving, but we are restricting the resulting velocities and accelerations to be small enough so as not to affect the force and moment equilibrium of our system.

As we stated earlier in Section 2.2.1, we will be assuming that dry Coulomb

friction is acting between the parts. In particular, we shall be using the *static* coefficient of friction  $\mu_s$ . In general the static coefficient of friction is larger than the dynamic coefficient of friction  $\mu_d$  that governs friction between moving objects. By assuming only one coefficient of friction  $\mu$  equal to the static coefficient of friction, we are making a *conservative* assumption. We are therefore implicitly assuming that solutions which work for systems with higher coefficients of friction will also work for systems with a lower  $\mu$ .

In modeling the geometry of our system we shall be assuming rigid parts. In addition, since we are assuming a quasi-static analysis, we shall model the parts without mass or rotational inertia. Since the parts are assumed to have negligible mass, the effects of gravity will also be considered negligible.

In deriving and simplifying our system equations, we shall avoid the temptation to make small angle approximations. This will be especially important in cases where the operating range of the system is rather large, such as the case where the peg is tilted relative to the hole.

As we progress through the strategy development procedure we will continue to make assumptions as necessary. As we shall see, we will be required to revise and extend some of the assumptions we have already made to arrive at an acceptable solution. This need to revise assumptions is an inherent part of the design process, examples of which we shall see over and over again throughout the following chapters.

### 2.4.2 Possible Configurations During Assembly

In order to derive the appropriate quasi-static equilibrium equations, we will first have to identify the contact configurations possible between the peg and the hole, since each of these configurations will have different boundary conditions.

Figure 2.8 illustrates seven of the possible configurations between the peg and the hole. The other seven are the mirror images of these seven and result from the peg being tilted in the opposite direction, giving a total of 14 possible configurations. All of the contacts are assumed to be point on plane. We notice that configurations 1 and 7 look identical except for being on opposite sides of

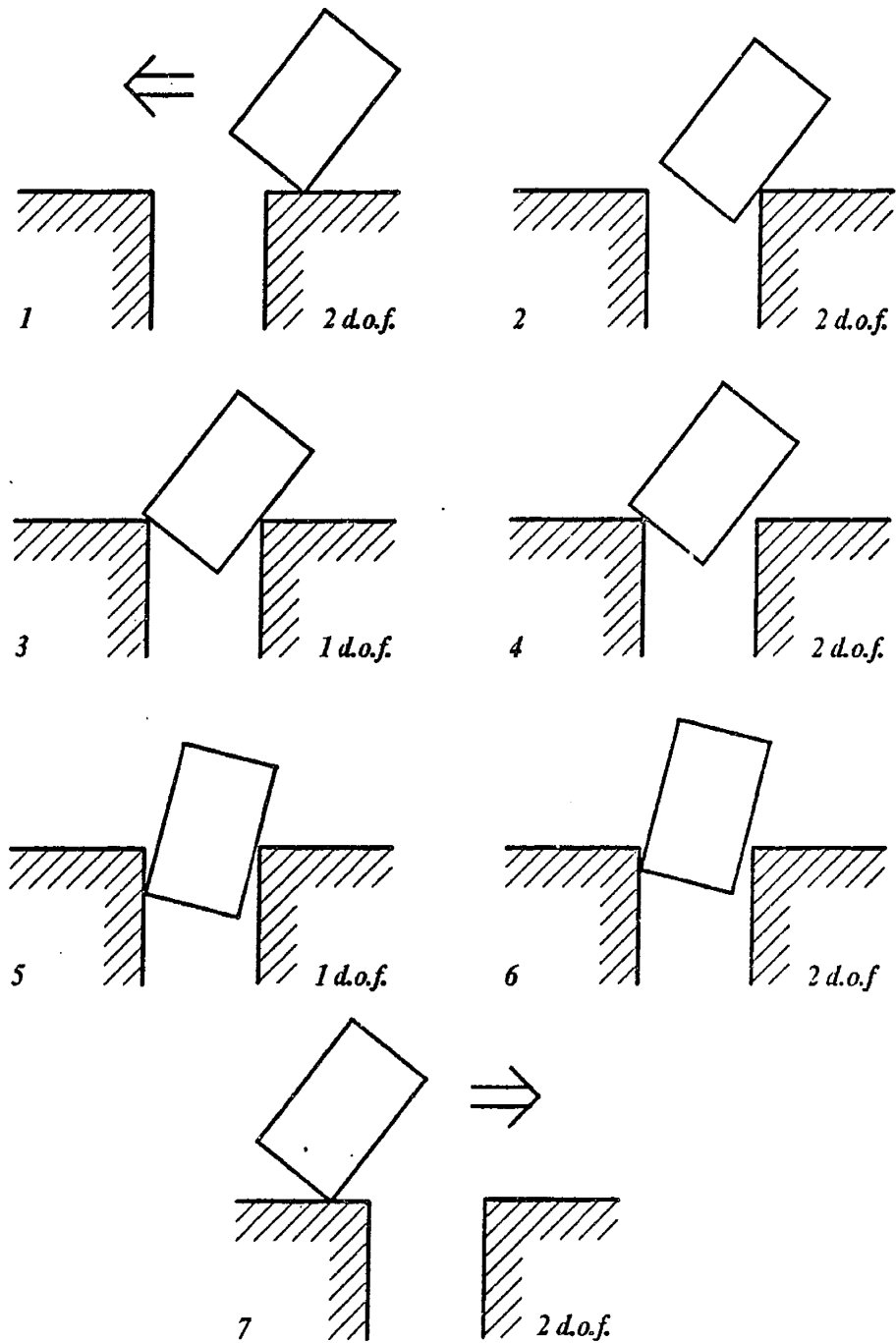


Figure 2.8: Contact Configurations

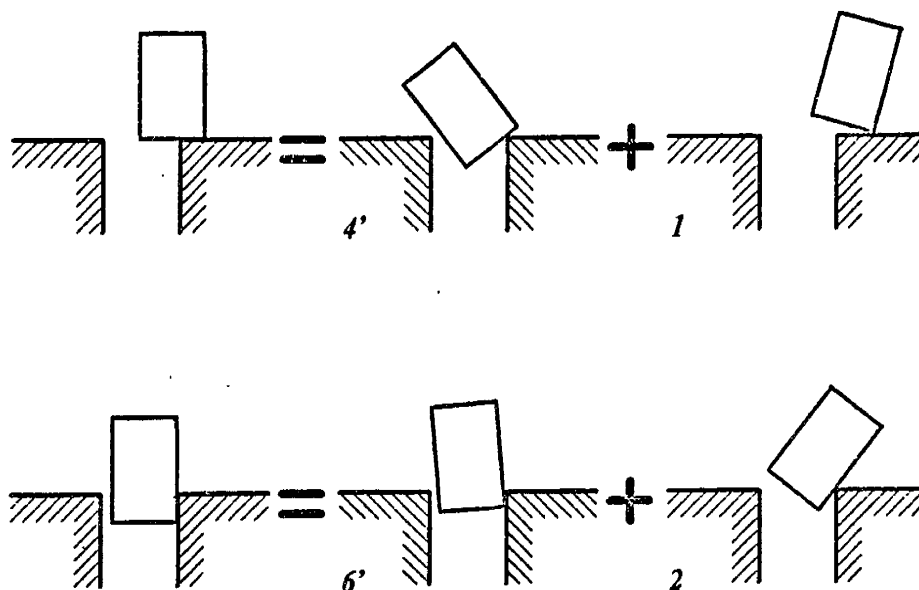


Figure 2.9: Plane-Plane Contact

the hole. In fact the only difference between them is in the fact that the peg in configuration 1 must slide to the left to reach the hole and configuration 7 must slide to the right. As we shall see, the direction of sliding will affect the resulting equations, so we shall distinguish these as two separate configurations. Two kinds of configurations omitted from this list are shown in Figure 2.9 where the contact is between two planes. Another special case is shown in Figure 2.10. Here the bottom corner of the peg has just crossed the top corner of the hole. These special configurations represent transitions between the original 14 configurations listed. Each of them can be represented as some combination of the other 14, as shown. We will number the configurations as shown, using a 'prime' to indicate a corresponding configuration with opposite tilt.

The degrees of freedom of each of these configurations are listed next to the appropriate illustration. For the peg unconstrained in the plane of the assembly, there are three independent parameters necessary to specify its position  $(x, y, \theta)$

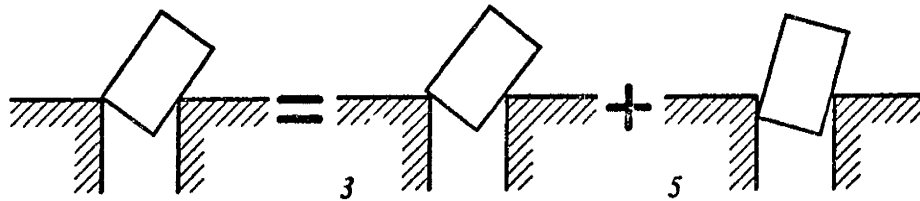


Figure 2.10: Corner Crossing

and therefore three degrees of freedom. For the peg in one point contact with the hole there are two degrees of freedom, and for the peg in two point contact with the hole there is one degree of freedom, and so on.

In general a given assembly strategy will not involve all 14 of the configurations listed and so we shall choose a subset sufficient to describe the insertion process. This subset will be selected later as part of the strategy development process.

### 2.4.3 Geometry of the Chamferless Case

When the peg is in contact with the hole and the number of degrees of freedom is less than three, the set of dependent position parameters can be expressed in terms of those that are independent. We shall determine these dependency relationships by deriving the kinematic constraints between the peg and the hole for various contact configurations.

Figure 2.11 shows the peg and the hole with a coordinate frame affixed to each. The width of the hole is given by the parameter  $D$ , the width of the peg by  $d$ , and the displacement of the peg's coordinate frame from the tip along the axis of the peg is given by  $L$ . One way to specify the position of the peg relative to the hole is to just determine the coordinate transformations between the two frames.

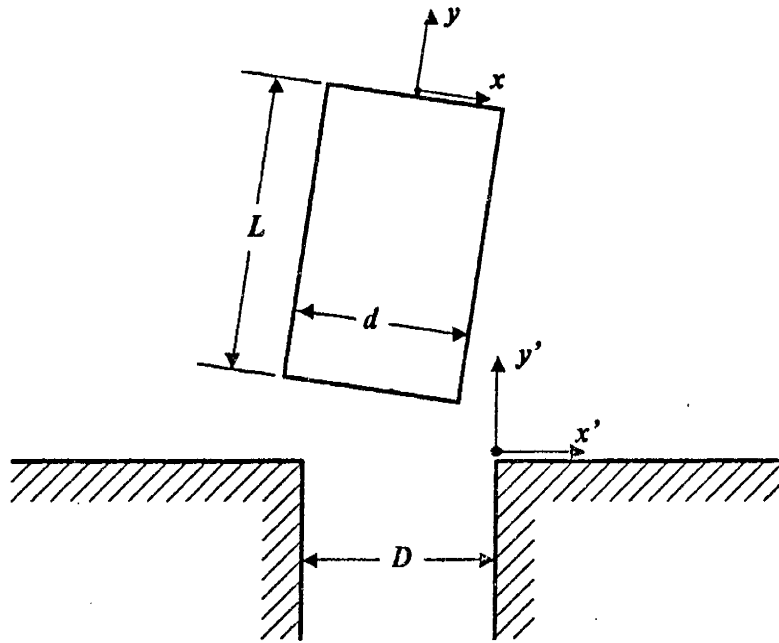


Figure 2.11: Peg and Hole Coordinates

However, such a representation would be somewhat more cumbersome than we need at this point.<sup>6</sup> A simpler way to represent the kinematic constraints of our system is shown in Figure 2.12.

For the case where the peg is in two point contact outside of the hole (configuration 3), we can define the linear contact parameters  $l$  and  $s$  as shown. From the figure and the dimensions given we can then write

$$\begin{aligned} (d - s) \cos \theta + l \sin \theta &= D \\ (d - s) \sin \theta &= l \cos \theta \end{aligned} \quad (2.3)$$

Solving for  $s$  and  $l$  we obtain

$$\begin{aligned} s &= d - D \cos \theta \\ l &= D \sin \theta \end{aligned} \quad (2.4)$$

where the sense of  $\theta$  is as shown. For the case where the peg is in two point contact inside of the hole (configuration 5), as shown in the lower part of the

<sup>6</sup>In Chapter 3, where the geometric constraints will be considerably more complex, we will adopt such an approach.



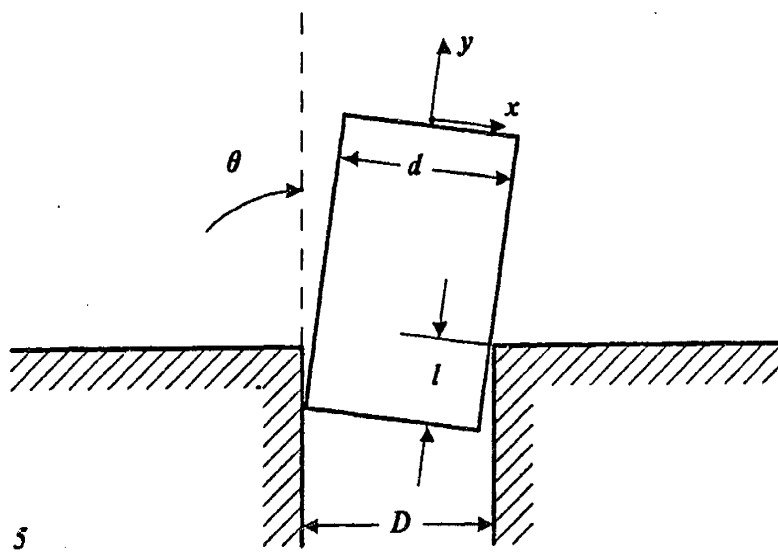
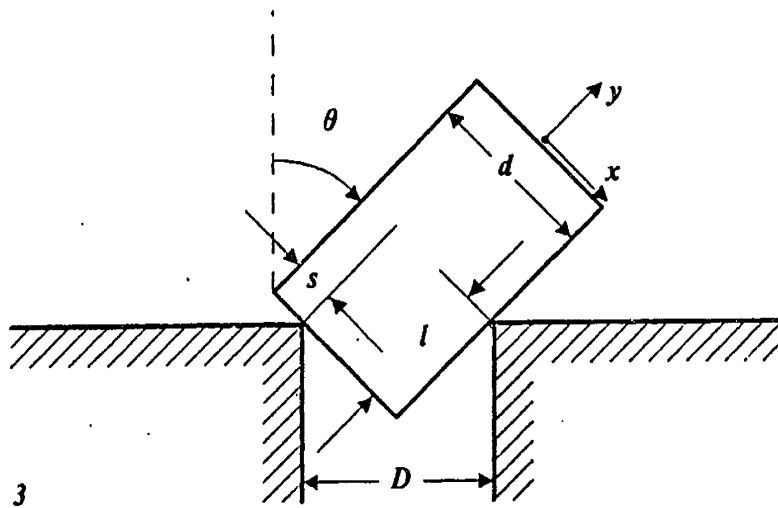


Figure 2.12: Kinematic Constraints in Two Point Contact

figure, we can write

$$l \sin \theta + d \cos \theta = D \quad (2.5)$$

which gives us the expression for  $l$  as

$$l = \frac{D - d \cos \theta}{\sin \theta} \quad (2.6)$$

Using these expressions we can determine the positions of the contact points on the peg solely in terms of  $l$ ,  $s$ , or the angle of tilt  $\theta$ . For the one point contact cases, the tilt and contact position variables will be independent and will therefore have to be specified separately.

#### 2.4.4 Quasi-static Force Analysis

Now that we have developed the geometric models of the peg and hole system we will derive the governing quasi-static force and moment balance equations. As we stated earlier, we will assume that the peg is in a state of impending motion and will therefore constrain the reaction forces to lie on the boundaries of their respective friction cones.

We will begin by considering the case of the peg in two point contact outside of the hole (configuration 3). Figure 2.13 shows the applied force and moment and the resulting reaction forces that the hole applies to the peg. Note that the frictional forces are acting to oppose the motion of the peg as it aligns itself with the hole, i.e. the peg is in impending motion towards the direction of insertion. The applied force and moment are applied to the peg with the force vector passing through the origin of the coordinate frame as shown and we shall derive all of our equations in the peg coordinate frame. The applied force  $F$  will be represented in polar form as shown in the figure. Therefore the  $x$  and  $y$  components of the force become

$$\begin{aligned} F_x &= F \sin \alpha \\ F_y &= -F \cos \alpha \end{aligned} \quad (2.7)$$

where the sense of  $\alpha$  is as shown. We can write the force and moment balance

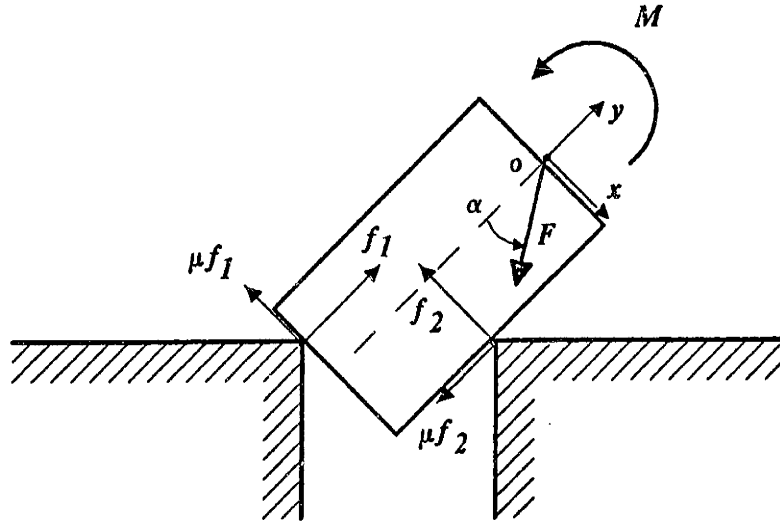


Figure 2.13: Force Equilibrium: Configuration 3

equations as follows

$$\begin{aligned}\sum F_x &= 0 \\ \sum F_y &= 0 \\ \sum M_o &= 0\end{aligned}\tag{2.8}$$

where the point  $o$  is located at the origin of the peg's coordinate frame. For this particular configuration we can write

$$\begin{aligned}\sum F_x &= F \sin \alpha - \mu f_1 - f_2 = 0 \\ \sum F_y &= -F \cos \alpha - f_1 - \mu f_2 = 0 \\ \sum M_o &= M - \mu f_1 L - f_1 \left( \frac{d}{2} - s \right) - \mu f_2 \frac{d}{2} - f_2 (L - l) = 0\end{aligned}\tag{2.9}$$

When we plug the kinematic constraint relations for  $l$  and  $s$  in terms of  $\theta$  and eliminate  $f_1$  and  $f_2$ , we obtain the following expression

$$M = F \sin \alpha \left[ L + D \frac{\mu \cos \theta - \sin \theta}{\mu^2 + 1} \right] + F \cos \alpha \left[ -\frac{d}{2} + D \frac{\cos \theta + \mu \sin \theta}{\mu^2 + 1} \right]\tag{2.10}$$

This equation determines the values of the applied force  $F$ , at an angle  $\alpha$  to the axis of the peg, and the applied moment  $M$  that will keep the peg in a state of impending motion in a given position of the peg determined by  $\theta$ . We can also solve the two force balance equations for the reaction forces  $f_1$  and  $f_2$  in terms of the applied force angle  $\alpha$

$$\begin{aligned} f_1 &= F \frac{\mu \sin \alpha + \cos \alpha}{\mu^2 + 1} \\ f_2 &= F \frac{\sin \alpha - \mu \cos \alpha}{\mu^2 + 1} \end{aligned} \quad (2.11)$$

In order to determine the limits on the applied force so as not to break contact, we shall solve these two expressions for  $f_1 \geq 0$  and  $f_2 \geq 0$  respectively to obtain

$$\begin{aligned} \alpha &\geq \arctan\left(-\frac{1}{\mu}\right) \\ \alpha &\geq \arctan(\mu) \end{aligned} \quad (2.12)$$

We notice that Equations 2.12 are in the form of inequalities but Equations 2.10 is an equality. In order to determine which values of  $M$  and  $F$  will jam and which will slide in terms of these equations, we shall have to introduce some inequalities. Recall from Figure 2.3 the definition of the friction cone. If we have a reaction force on the edge of the cone with no net applied force, we know that we are in a state of impending motion. If we imagine a small variation in the tangential component of the reaction force of the form  $f_{\text{tangential}} \rightarrow f_{\text{tangential}} + \delta$  where  $0 < \delta \ll 1$ , then the resulting equilibrium equations will determine the applied force and moment necessary to balance this extra tangential component of the reaction force. If we include this  $\delta$  into our equations and solve them for  $\delta \geq 0$ , the resulting inequality expressions will determine which way to depart from impending motion equilibrium for sliding to occur. By replacing  $\mu f_1$  with  $\mu f_1 + \delta$ , and  $\mu f_2$  with  $\mu f_2 + \delta$ , and re-evaluating Equations 2.9, we obtain the following expression

$$\begin{aligned} M &= F \sin \alpha \left[ L + D \frac{\mu \cos \theta - \sin \theta}{\mu^2 + 1} \right] + F \cos \alpha \left[ -\frac{d}{2} + D \frac{\cos \theta + \mu \sin \theta}{\mu^2 + 1} \right] \\ &\quad + \delta \left[ D \frac{\mu(\sin \theta - \cos \theta) + \sin \theta + \cos \theta}{\mu^2 + 1} \right] \end{aligned} \quad (2.13)$$

Since the parts are assumed perfectly rigid, the magnitude of the applied force will not affect the resulting equilibrium except to scale the associated moment. We can therefore normalize our force and moment balance equations with respect to the applied force.<sup>7</sup> If we solve the above equation for  $\delta \geq 0$  and divide through by  $F$ , we obtain the inequality

$$\frac{M}{F} \geq \sin \alpha \left[ L + D \frac{\mu \cos \theta - \sin \theta}{\mu^2 + 1} \right] + \cos \alpha \left[ -\frac{d}{2} + D \frac{\cos \theta + \mu \sin \theta}{\mu^2 + 1} \right] \quad (2.14)$$

where we restrict the expression multiplying  $\delta$  to be greater than zero. This will be guaranteed if  $\theta > \arctan\left(\frac{\mu-1}{\mu+1}\right)$ .

The two sets of Equations 2.14 and 2.12 together represent the constraints on the applied force and moment that will successfully slide the peg in the direction of insertion for a given position of configuration 3.

We can follow a similar process to determine the constraint inequality equations for configuration 5. Figure 2.14 shows the peg in two point contact within the hole. Again the reaction forces are assumed to be on the edges of their respective friction cones and the system is assumed to be in an equilibrium state of impending motion. The senses of the frictional components of the reaction forces are opposing the motion of the peg into the hole. The quasi-static equilibrium equations are

$$\begin{aligned} \sum F_x &= -f_2 + f_1 \cos \theta - \mu \sin \theta + F \sin \alpha = 0 \\ \sum F_y &= \mu f_1 \cos \theta + f_1 \sin \theta + \mu f_2 - F \cos \alpha = 0 \\ \sum M_1 &= M + \mu f_2 d + f_2 l - F \left(\frac{d}{2}\right) \cos \alpha - FL \sin \alpha = 0 \end{aligned} \quad (2.15)$$

where the moments have been taken around point 1 for simplicity. Again making the substitutions  $\mu f_1 \rightarrow \mu f_1 + \delta$ , and  $\mu f_2 \rightarrow \mu f_2 + \delta$  and solving we obtain the following

$$\begin{aligned} f_1 &= \frac{F(\mu \sin \alpha - \cos \alpha)}{(\mu^2 - 1) \sin \theta - 2\mu \cos \theta} \\ f_2 &= \frac{F(\cos \alpha(\mu \sin \theta - \cos \theta) - \sin \alpha(\mu \cos \theta + \sin \theta))}{(\mu^2 - 1) \sin \theta - 2\mu \cos \theta} \end{aligned}$$

---

<sup>7</sup>Note that wedging may require this assumption to be revised under certain conditions. See Section 2.2.2.

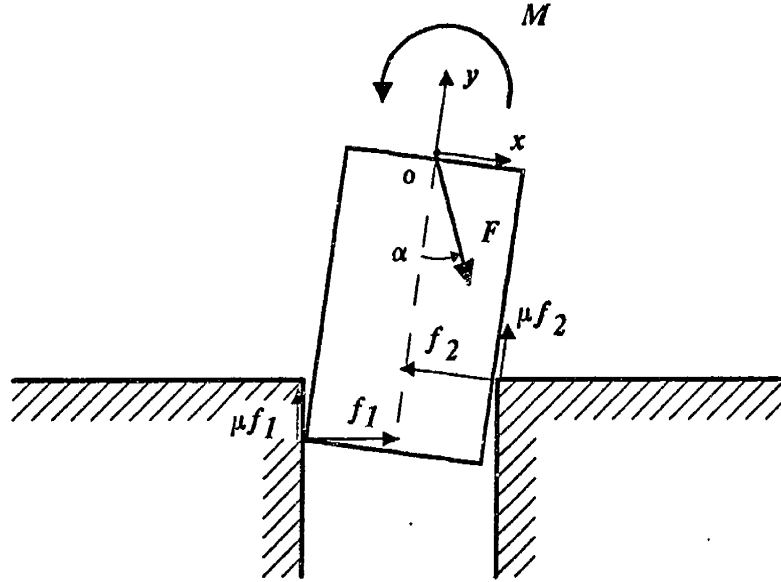


Figure 2.14: Force Equilibrium: Configuration 5

$$\begin{aligned}
 M = F \sin \alpha & \left[ L - \frac{D(\mu \cos \theta + \sin \theta) + d \left( \frac{\mu^2 - 1}{2} \sin 2\theta - \mu \cos 2\theta \right)}{1 - (\mu \sin \theta - \cos \theta)^2} \right] \\
 & + F \cos \alpha \left[ \frac{d + D(\mu \sin \theta - \cos \theta)}{1 - (\mu \sin \theta - \cos \theta)^2} - \frac{d}{2} \right] \\
 & + \delta \left[ \frac{(D - d)(1 + \cos \theta - \mu \sin \theta)}{1 - (\mu \sin \theta - \cos \theta)^2} \right] \quad (2.16)
 \end{aligned}$$

If we solve the above equations for  $f_1 \geq 0$ ,  $f_2 \geq 0$ , and  $\delta \geq 0$  respectively, we obtain the following constraint expressions

$$\alpha \leq \arctan \left( \frac{1}{\mu} \right)$$

$$\alpha \geq \arctan \left( \frac{\mu \sin \theta - \cos \theta}{\mu \cos \theta + \sin \theta} \right)$$

$$\frac{M}{F} \geq \sin \alpha \left[ L - \frac{D(\mu \cos \theta + \sin \theta) + d \left( \frac{\mu^2 - 1}{2} \sin 2\theta - \mu \cos 2\theta \right)}{1 - (\mu \sin \theta - \cos \theta)^2} \right] +$$

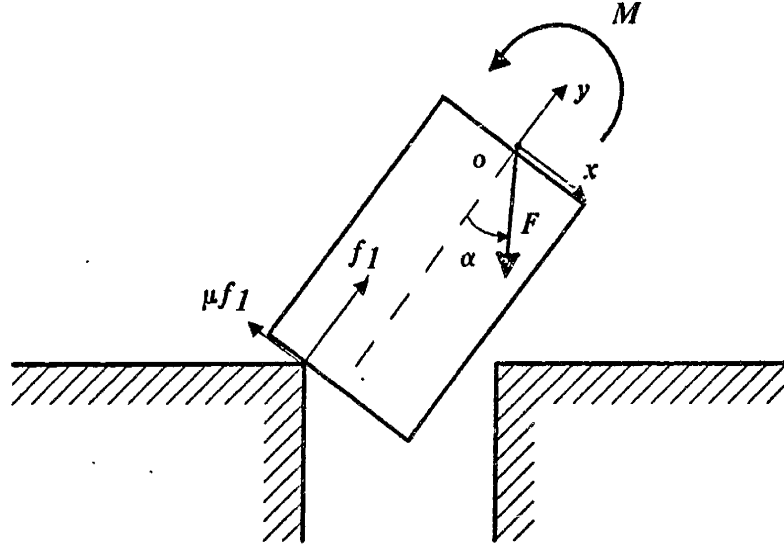


Figure 2.15: Force Equilibrium: Configuration 4

$$\cos \alpha \left[ \frac{d + D(\mu \sin \theta - \cos \theta)}{1 - (\mu \sin \theta - \cos \theta)^2} - \frac{d}{2} \right] \quad (2.17)$$

The assumptions made in evaluating the above inequalities are that  $\theta \geq 0$ , which must be true to be in this configuration, and  $\mu < 1$ . For a case where  $\mu > 1$ , the relation  $\theta \geq \arctan\left(\frac{2\mu}{\mu^2-1}\right)$  must be true for the inequalities to remain the same. If this last relation is not true, then some of the inequalities will be reversed.

For the peg in one point contact with the hole, as shown in Figure 2.15, we can carry out a similar impending motion analysis. In this case, however, the force and moment balance equations will be decoupled. For the system shown we can write

$$\begin{aligned} \sum F_x &= F \sin \alpha - \mu f_1 = 0 \\ \sum F_y &= f_1 - F \cos \alpha = 0 \\ \sum M_1 &= M - FL \sin \alpha - F \left( \frac{d}{2} - s \right) \cos \alpha = 0 \end{aligned} \quad (2.18)$$

Here the peg will not break contact for  $-\frac{\pi}{2} \leq \alpha \leq \frac{\pi}{2}$ . The peg will slide towards

the hole if

$$\alpha \geq \arctan(\mu) \quad (2.19)$$

and the peg will rotate towards two point contact if

$$\frac{M}{F} \leq L \sin \alpha + \left( \frac{d}{2} - s \right) \cos \alpha \quad (2.20)$$

We notice that the moment has no direct effect on jamming in one point contact.

Only if the applied force lies within the friction cone will the peg jam.

The constraint equations for the rest of the contact configurations are:

Configuration 1

$$\alpha \leq \arctan \left( \frac{\sin \theta - \mu \cos \theta}{\cos \theta + \mu \sin \theta} \right) \quad \text{For sliding}$$

$$\alpha \geq \left( \theta - \frac{\pi}{2} \right) \quad \text{For maintaining contact}$$

$$\frac{M}{F} \geq L \sin \alpha - \frac{d}{2} \cos \alpha \quad \text{For rotating towards vertical} \quad (2.21)$$

Configuration 2

$$\alpha \leq \arctan \left( \frac{1}{\mu} \right) \quad \text{For sliding}$$

$$\alpha \geq 0 \quad \text{For maintaining contact}$$

$$\frac{M}{F} \geq (L - l) \sin \alpha - \frac{d}{2} \cos \alpha \quad \text{For regaining two point contact} \quad (2.22)$$

Configuration 6

$$\alpha \geq \arctan \left( \frac{\mu \sin \theta - \cos \theta}{\mu \cos \theta + \sin \theta} \right) \quad \text{For sliding}$$

$$\alpha \leq \theta \quad \text{For maintaining contact}$$

$$\frac{M}{F} \leq L \sin \alpha + \frac{d}{2} \cos \alpha \quad \text{For regaining two point contact} \quad (2.23)$$

Configuration 7

$$\alpha \geq \arctan \left( \frac{\sin \theta - \mu \cos \theta}{\cos \theta + \mu \sin \theta} \right) \quad \text{For sliding}$$



$$\alpha \geq \left( \theta - \frac{\pi}{2} \right) \quad \text{For maintaining contact}$$

$$\frac{M}{F} \geq L \sin \alpha - \frac{d}{2} \cos \alpha \quad \text{For rotating towards vertical} \quad (2.24)$$

For the 'primed' configurations we shall define  $\phi$  as being the opposite angle of tilt. All other conventions will remain unchanged.

Configuration 1'

$$\alpha \geq \arctan \left( \frac{\mu \cos \phi - \sin \phi}{\cos \phi + \mu \sin \phi} \right) \quad \text{For sliding}$$

$$\alpha \leq \left( \frac{\pi}{2} - \phi \right) \quad \text{For maintaining contact}$$

$$\frac{M}{F} \leq L \sin \alpha + \frac{d}{2} \cos \alpha \quad \text{For rotating towards vertical} \quad (2.25)$$

Configuration 2'

$$\alpha \geq \arctan \left( -\frac{1}{\mu} \right) \quad \text{For sliding}$$

$$\alpha \leq 0 \quad \text{For maintaining contact}$$

$$\frac{M}{F} \leq (L - l) \sin \alpha + \frac{d}{2} \cos \alpha \quad \text{For regaining two point contact} \quad (2.26)$$

Configuration 3'

$$\alpha \leq \arctan \left( \frac{1}{\mu} \right) \quad \text{For } f_1 > 0$$

$$\alpha \leq \arctan(-\mu) \quad \text{For } f_2 > 0$$

$$\frac{M}{F} \leq \sin \alpha \left[ L + D \frac{\mu \cos \phi - \sin \phi}{\mu^2 + 1} \right] + \cos \alpha \left[ \frac{d}{2} - D \frac{\cos \phi + \mu \sin \phi}{\mu^2 + 1} \right] \quad (2.27)$$

For no jamming

where  $\phi > \arctan \left( \frac{\mu-1}{\mu+1} \right)$  as for configuration 3.

Configuration 4'

$$\alpha \leq \arctan(-\mu) \quad \text{For sliding}$$

$$-\frac{\pi}{2} \leq \alpha \leq \frac{\pi}{2} \quad \text{For maintaining contact}$$

$$\frac{M}{F} \geq L \sin \alpha - \left( \frac{d}{2} - s \right) \cos \alpha \quad \text{For regaining two point contact} \quad (2.28)$$

Configuration 5'

$$\begin{aligned} \alpha &\geq \arctan\left(-\frac{1}{\mu}\right) \quad \text{For } f_1 > 0 \\ \alpha &\leq \arctan\left(\frac{\cos\phi - \mu\sin\phi}{\mu\cos\phi + \sin\phi}\right) \quad \text{For } f_2 > 0 \\ \frac{M}{F} &\leq \sin\alpha \left[ L - \frac{D(\mu\cos\phi + \sin\phi) + d\left(\frac{\mu^2-1}{2}\sin 2\phi - \mu\cos 2\phi\right)}{1 - (\mu\sin\phi - \cos\phi)^2} \right] \\ &+ \cos\alpha \left[ \frac{d}{2} - \frac{d + D(\mu\sin\phi - \cos\phi)}{1 - (\mu\sin\phi - \cos\phi)^2} \right] \quad \text{For no jamming} \end{aligned} \quad (2.29)$$

where  $\phi \geq 0$ , and  $\mu < 1$ . For a case where  $\mu > 1$ , the relation  $\theta \geq \arctan\left(\frac{2\mu}{\mu^2-1}\right)$  must again be true. If this relation is not true, then some of the inequalities will be reversed.

Configuration 6'

$$\begin{aligned} \alpha &\leq \arctan\left(\frac{\cos\phi - \mu\sin\phi}{\mu\cos\phi + \sin\phi}\right) \quad \text{For sliding} \\ \alpha &\geq -\phi \quad \text{For maintaining contact} \\ \frac{M}{F} &\geq L\sin\alpha - \frac{d}{2}\cos\alpha \quad \text{For regaining two point contact} \end{aligned} \quad (2.30)$$

Configuration 7'

$$\begin{aligned} \alpha &\leq \arctan\left(\frac{\sin\phi + \mu\cos\phi}{\mu\sin\phi - \cos\phi}\right) \quad \text{For sliding} \\ \alpha &\leq \left(\frac{\pi}{2} - \phi\right) \quad \text{For maintaining contact} \\ \frac{M}{F} &\leq L\sin\alpha + \frac{d}{2}\cos\alpha \quad \text{For rotating towards vertical} \end{aligned} \quad (2.31)$$

### 2.4.5 The Applied Force-Moment Constraint Space

Now that we have derived the equations that govern the constraints on our applied forces and moments, we need a means by which we can represent these constraints collectively for developing an assembly strategy. Figure 2.16 illustrates an example of one such visual representation. The vertical axis is the ratio

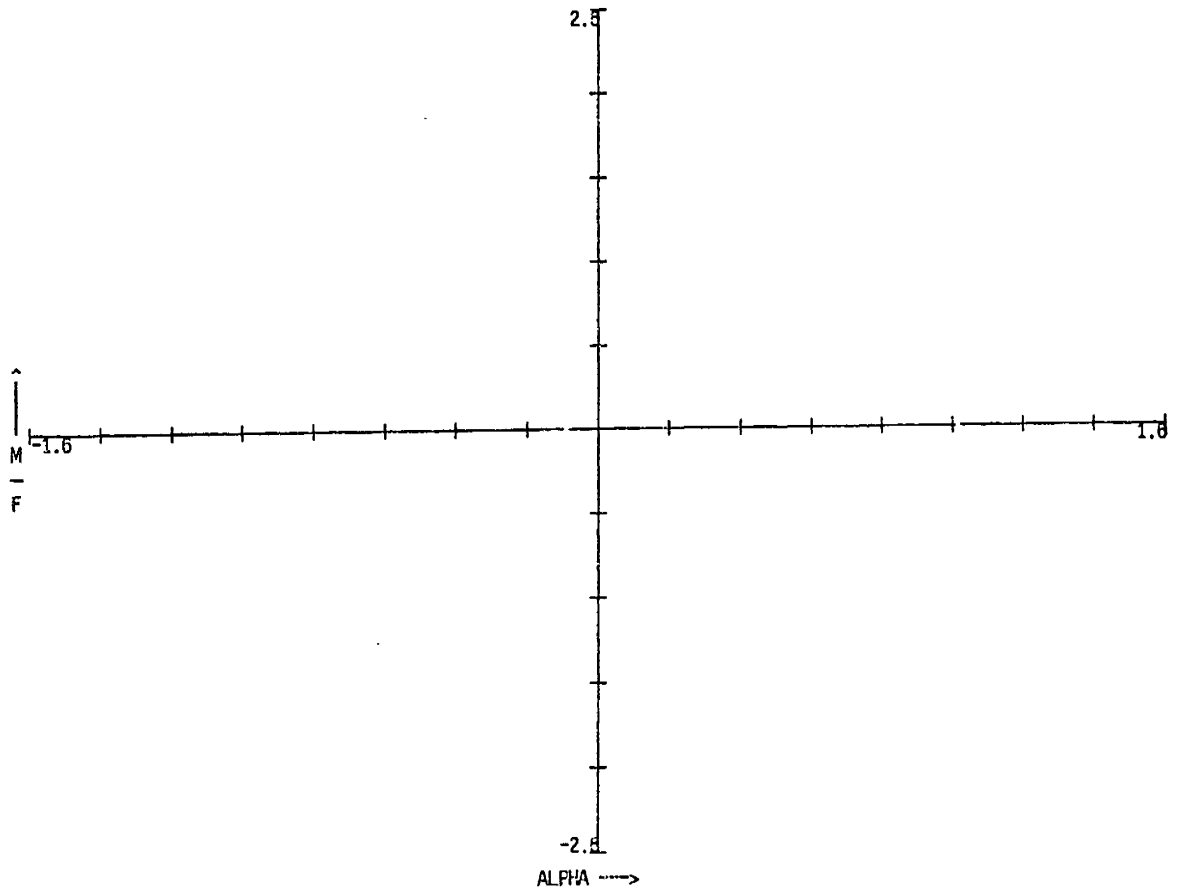


Figure 2.16: The Applied Force-Moment Constraint Space

of the magnitude of the applied moment to the applied force, and the horizontal axis is the angle  $\alpha$  that the applied force takes with respect to the centerline of the peg. If we plot as curves in this space the constraint equations just derived for given positions of the peg, we will be able to determine graphically the allowable  $\frac{M}{F}$  and  $\alpha$  values to control our system.

Since the constraint curves are functions of the configuration as well as the position of the peg relative to the hole, we will need to determine a *path* of the peg. This path will consist of a set of configurations and positions that will take the peg from its initial condition to being fully inserted into the hole. In order to determine what force and moment will move the peg along such a path, we will determine the intersection regions bounded by the constraint curves of each position. If an intersection region exists that is valid over all points of the path, then any force-moment combination within this region will constitute a valid control input for an assembly strategy.

#### 2.4.5.1 Plotting the Constraint Curves

In order to obtain a feeling for the types of solution regions we will be dealing with, we shall plot a few constraint curves for various configurations of the peg and hole. As an example, we will consider a peg and hole system with the following parameters

$$\begin{aligned} d &= 0.995 \text{ in} \\ D &= 1.000 \text{ in} \\ L &= 2.000 \text{ in} \\ \mu &= 0.9 \quad \text{aluminum on aluminum} \end{aligned}$$

Figures 2.17 and 2.18 show some pairs of constraint curves from case 3 and its complement 3' for various values of  $\theta$  and  $\phi$ . As we can see, there is no intersection region for two cases of the peg outside of the hole with opposite tilts. We interpret this to mean that the initial tilt of the peg must be determined as part of an insertion strategy since a force and moment that will slide for one tilt will jam for the other. We also see that the regions bounded by the constraint

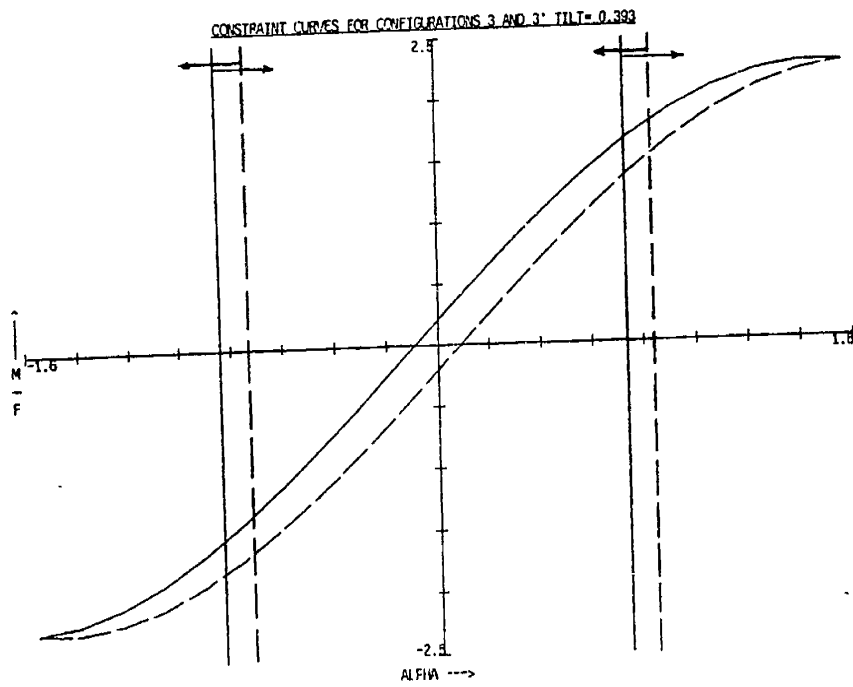
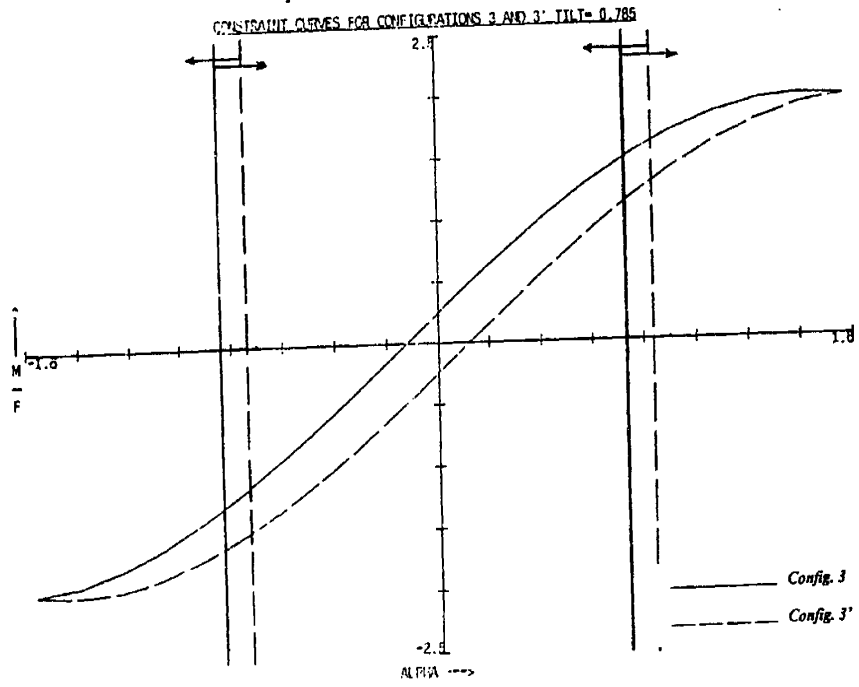


Figure 2.17: Peg Out of the Hole

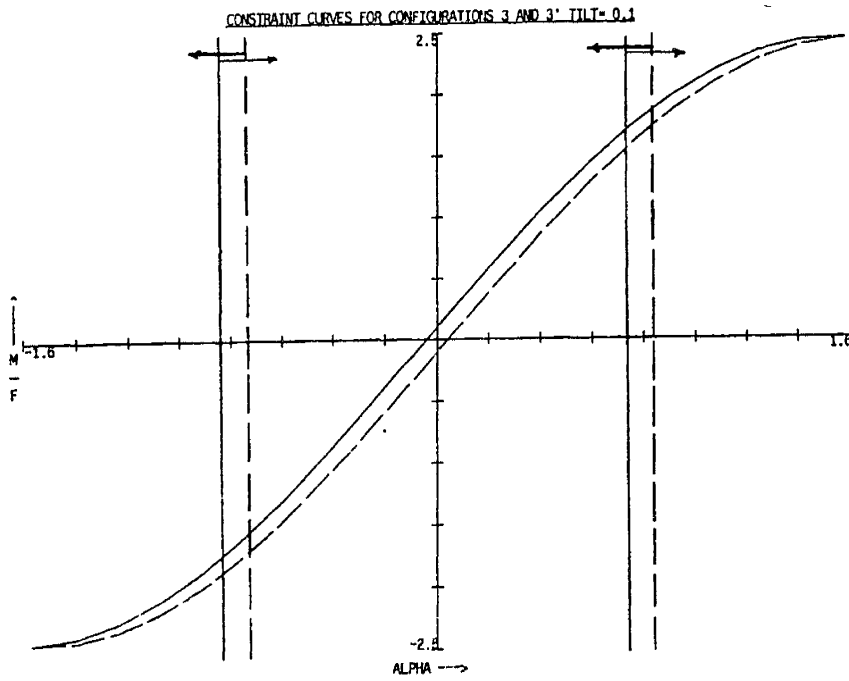
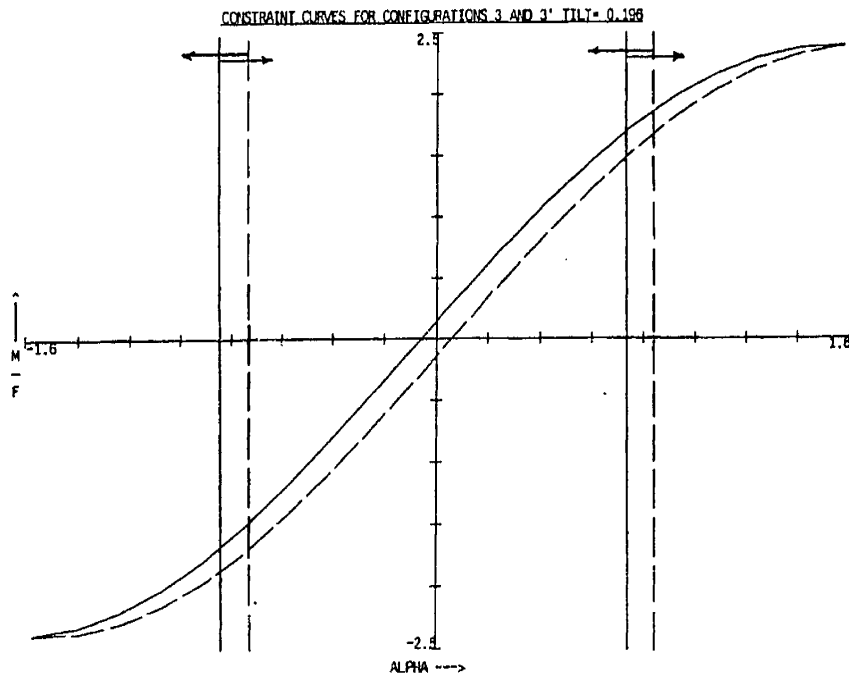


Figure 2.18: Peg Out of the Hole, (Cont.)

curves become *steeper* as the angle of tilt ( $\theta$  or  $\phi$ ) is decreased, i.e. for a given  $\alpha$  the magnitude of the corresponding  $\frac{M}{F}$  becomes larger.

Figures 2.19 and 2.20 show some pairs of constraint curves from cases 5 and 5' for various values of  $\theta$  and  $\phi$ . For this case of the peg *in* the hole we see that there *is* an intersection region for the peg in opposite tilts.<sup>8</sup> Here the intersection region becomes larger as the tilt of the peg (positive or negative) is decreased, corresponding to the peg sliding deeper into the hole. We interpret this to mean that unlike the case of the peg outside of the hole, there exists a set of forces and moments which will slide the peg inside the hole regardless of which way the peg is tilted. In addition, as the peg slides deeper into the hole the range of these sliding forces and moments increases.

The set of constraint curves for the peg outside of the hole supports our selection of the peg tilted relative to the hole as an initial condition. If we tilt the peg before we begin the insertion then we know in advance which way it will be tilted as it enters the hole. Therefore, we will choose to begin our assembly path with the peg in configuration 3. From configuration 3 we will slide in two point contact until the bottom corner of the peg crosses the top corner of the hole. From this transition point we will continue to slide in two point contact inside the hole (configuration 5) until the peg is fully inserted. As we mentioned in Section 2.3.2, we shall avoid breaking contacts in order to reduce the uncertainty of where the peg would resume two point contact with the hole.

#### 2.4.5.2 Constraints at the Configuration Transition

As we saw in Figures 2.17 through 2.20, the constraint curves changed as a function of the position of the peg relative to the hole. Since these changes were rather gradual, we can say that the constraint curves are well behaved for small changes in position within a given configuration. The next question is how do these curves behave in a transition *between* configurations. In particular, given our proposed assembly path, a transition of configurations will occur when the

---

<sup>8</sup>Except for  $\theta = \theta_c$  and  $\phi = \phi_c$  where the curves lie on top of one another.

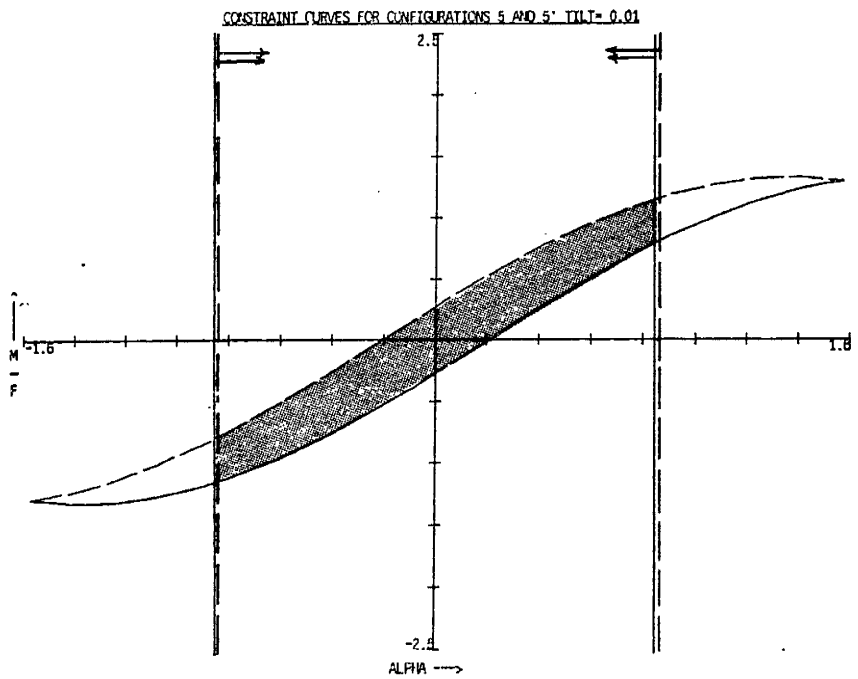
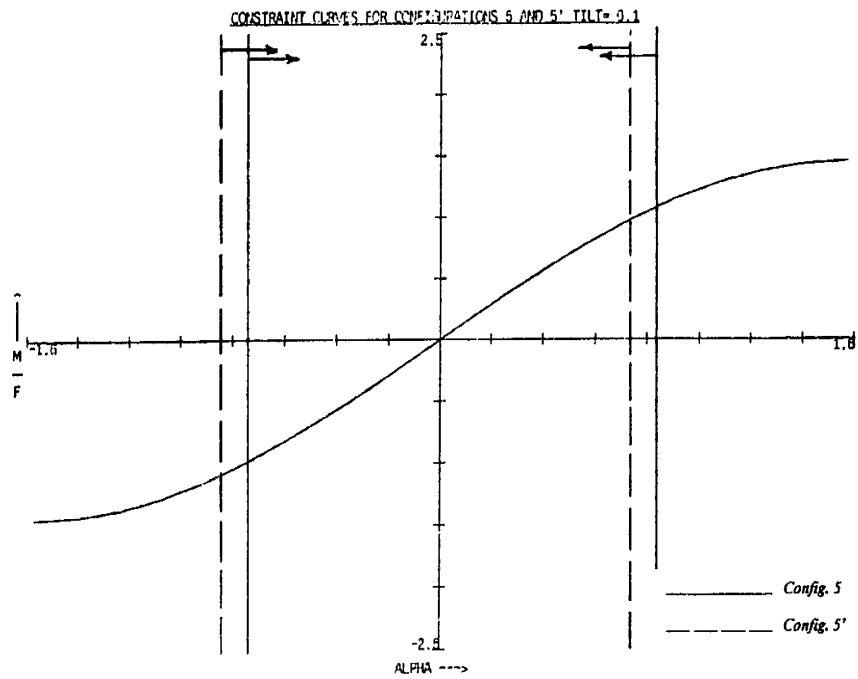


Figure 2.19: Peg in the Hole



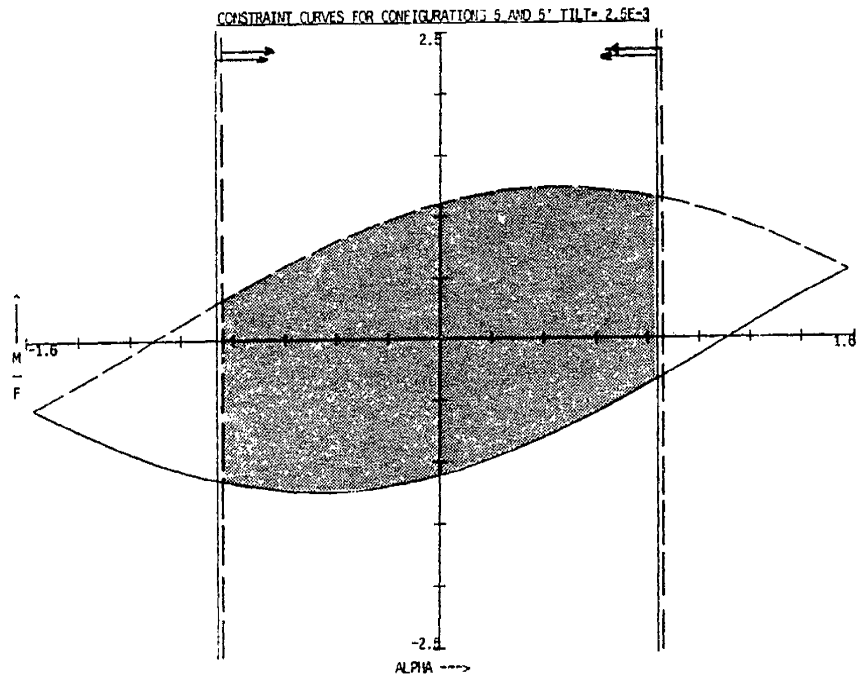
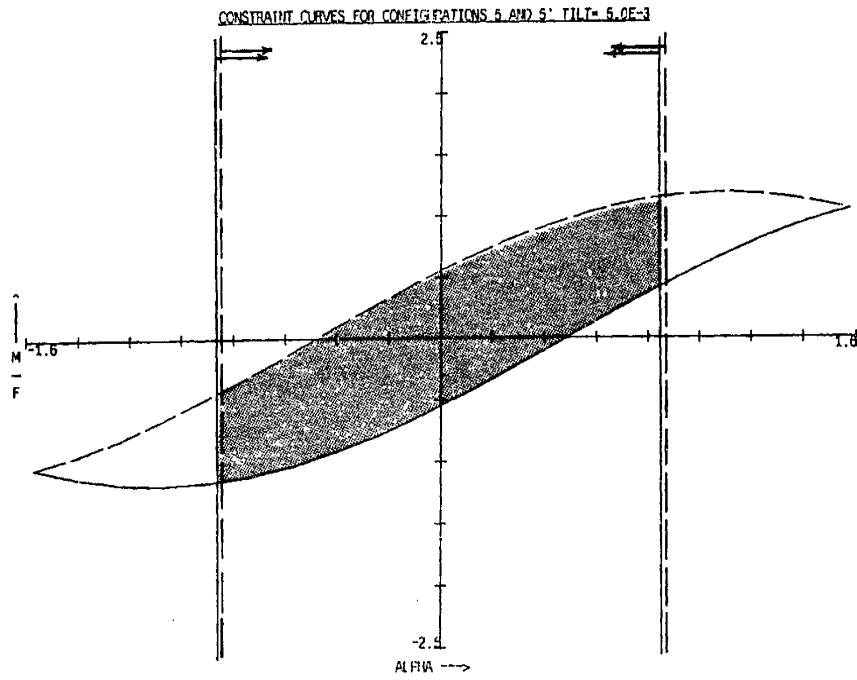


Figure 2.20: Peg in the Hole, (Cont.)

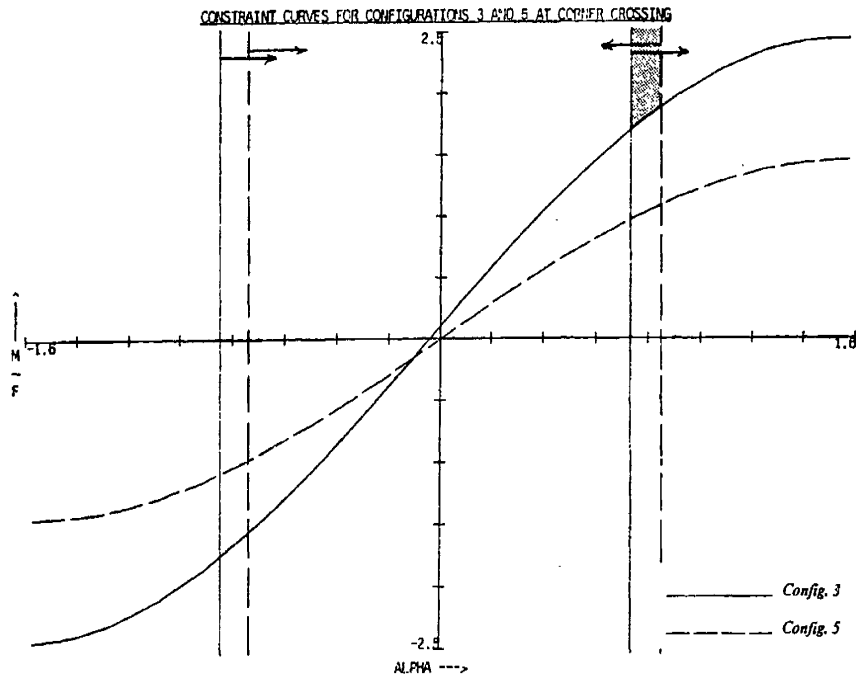


Figure 2.21: Corner Crossing

bottom of the peg crosses the top edge of the hole. It will therefore be necessary to establish the behavior of the constraint curves during this transition.

We can determine at precisely what angle of tilt corner crossing will occur by examining Equation 2.4. At corner crossing the linear parameter  $s$  will become zero. So by solving Equation 2.4 for  $s = 0$  we obtain

$$\theta_c = \arccos\left(\frac{d}{D}\right) \quad (2.32)$$

By evaluating the constraint equations for configurations 3 and 5 with  $\theta = \theta_c$  and superimposing the resulting curves in the  $(\frac{M}{F} vs \alpha)$  plane, we obtain the regions shown in Figure 2.21.

Whereas the constraint curves changed only slightly for variations in position within a configuration, from this figure we see that there is a significant and discontinuous change in solution regions in the transition between configurations 3 and 5. Since the curves will vary only slowly within a configuration, forces and

moments that are chosen from within a force-moment constraint region for one position will in general be valid for a range of positions within the configuration. Therefore, as the peg slides along a portion of an assembly path that lies within one configuration, we can represent the general force-moment constraints for that portion of the path in terms of just one position.

In addition to the realization that configuration transitions represent the points at which a given assembly strategy would be most likely to fail, we can see that for the corner crossing transition between configurations 3 and 5, there is a region of overlap between the constraint regions. This overlap region, shown shaded in the figure, represents those force-moment values that are guaranteed to carry the peg through corner crossing while satisfying the no-jamming and no-breaking contact constraints.

### 2.4.5.3 Effects of Parameters on Constraint Boundaries

In order to gain a better understanding of the factors that will affect the likelihood of our finding a solution that will neither break contacts nor jam the assembly, we will determine how the constraint curves behave with different chosen parameters. In other words, if we choose to assemble a peg with a different clearance relative to the hole ( $D - d$ ), a different friction  $\mu$ , or choose a different value of  $L$ , does the force-moment solution region change significantly? Since the corner crossing region is the sight of the greatest discontinuity in constraints within the range of positions we've examined, we shall concentrate our investigation here.

Figures 2.22 and 2.23 show the corner crossing constraints with four values of  $(D - d)$ . We see that as the clearances are decreased, all other parameters remaining constant, the constraint curves of configurations 3 and 5 change only slightly. In fact, the jamming constraint curve of configuration 3 does not change at all, while that of configuration 5 (peg in the hole) becomes slightly steeper and shifts to the left. These changes indicate that for the peg outside of the hole, the clearance has little or no effect on the conditions necessary for sliding while for the peg in the hole, the ratio of the moment to force needed for sliding is only slightly greater for tighter fits. In terms of the breaking contact constraints,

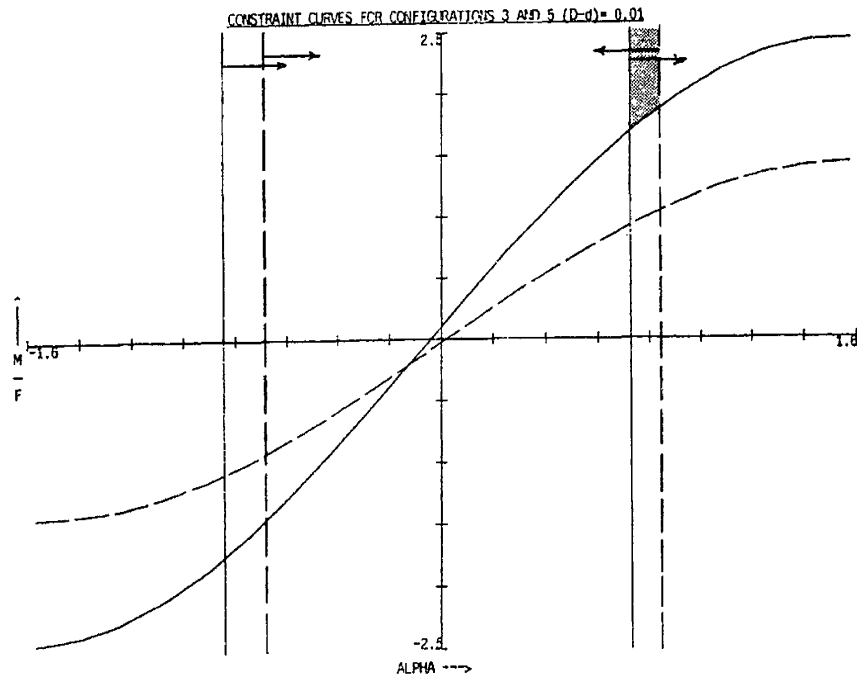
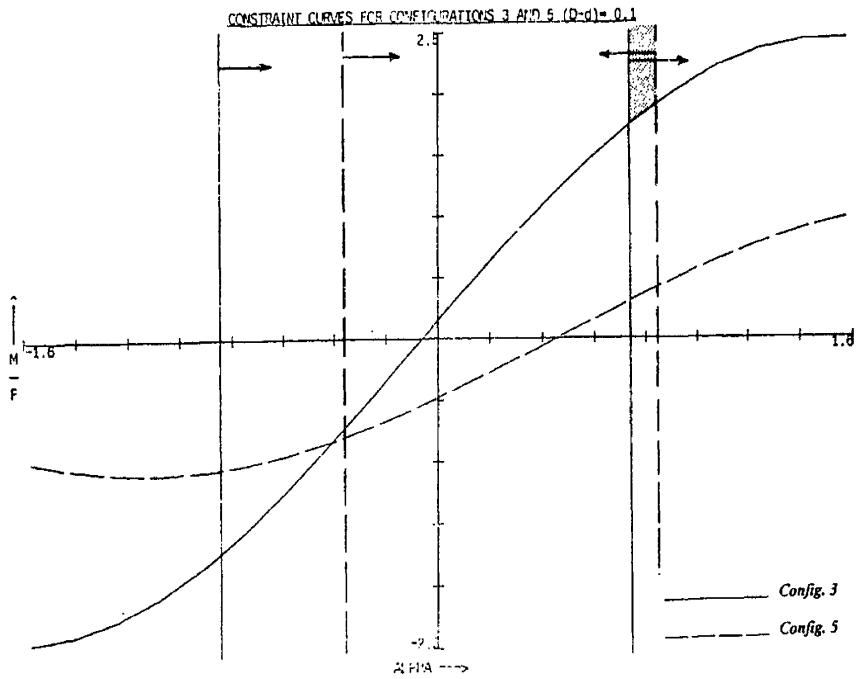


Figure 2.22: Corner Crossing: Various Clearances ( $D-d$ )

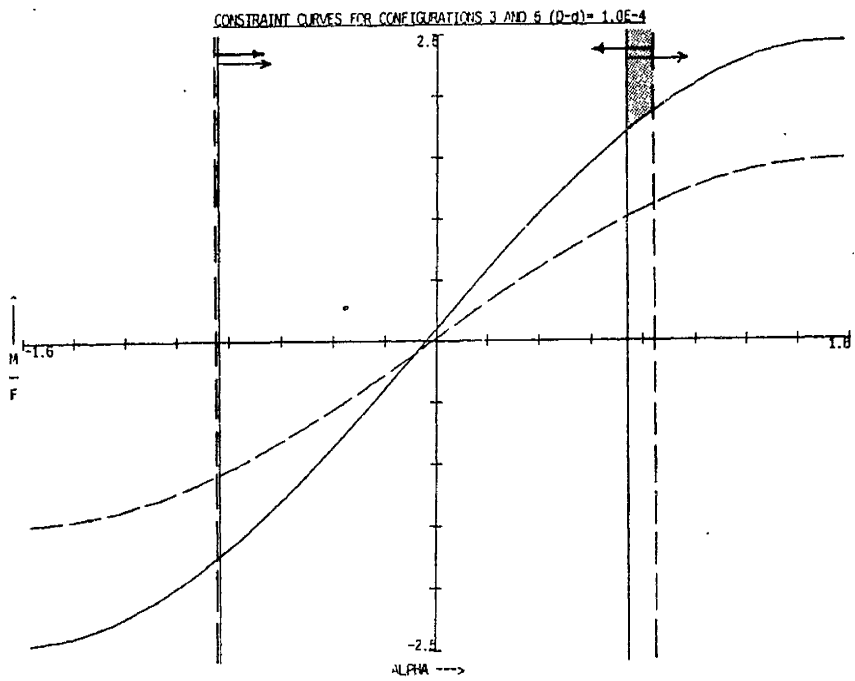
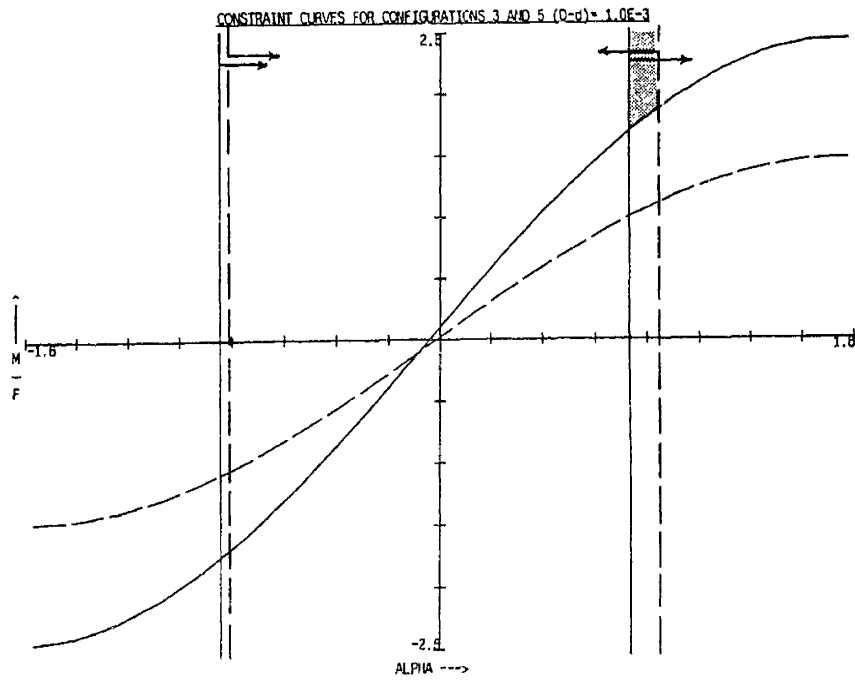


Figure 2.23: Corner Crossing: Various Clearances ( $D-d$ ), (Cont.)

since the value of  $\theta_c$  becomes smaller for tighter tolerances, the peg will have a wider range of angles of the allowable applied force for configuration 5 as the clearance is decreased.

In terms of the curves that define the boundaries of the allowable applied force-moment intersection region at corner crossing, i.e. the jamming and lower bound breaking contact curves for configuration 3 and the upper bound breaking contact curve for configuration 5, the clearance has little or no effect on the size of the resulting solution space.

Figures 2.24 and 2.25 show the corner crossing constraint regions with four different values of the coefficient of friction  $\mu$ . Here we see that the value of the friction has a rather significant effect on the size and location of the resulting intersection of constraint regions. For the case of no friction, any angle  $\alpha$  of the applied force between 0 and  $\frac{\pi}{2}$  will maintain two point contact. Thus, almost any combination of the applied force and moment that lies in the upper right quadrant of the constraint space will successfully slide the peg through the corner crossing region. As the value of the coefficient of friction is increased towards one, we see that the intersection of constraint spaces becomes smaller. At a value of  $\mu = 1$ , according to our analysis, there is no intersection region at corner crossing, indicating that it is impossible to slide and maintain two point contact at the same time during this transition phase. Under these conditions, then, it could be necessary to allow contacts to be broken at corner crossing.

Figures 2.26 and 2.27 show the corner crossing constraint regions for four values of the parameter  $L$ . We recall that  $L$  is distance from the bottom of the peg of the point at which the applied force  $F$  intersects the axis of the peg. In terms of the sliding constraint equations, the value of  $L$  will determine the magnitude of the equilibrium moment associated with a given applied force. For a value  $L$  equal to zero (applying the force at the tip of the peg), the jamming constraint curves lie near the horizontal axis. This implies that for an applied force at the tip of the peg, little moment need be applied to guarantee the peg will slide at corner crossing.<sup>9</sup> As the value of  $L$  is increased, the corresponding magnitude of the applied moment must also increase to guarantee sliding. We

<sup>9</sup>See [Whitney 82].

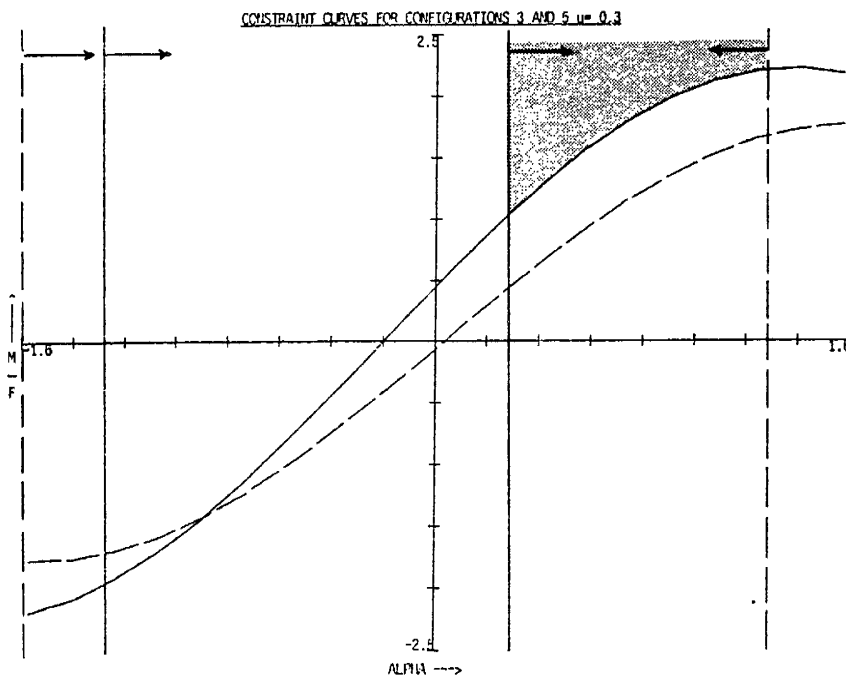
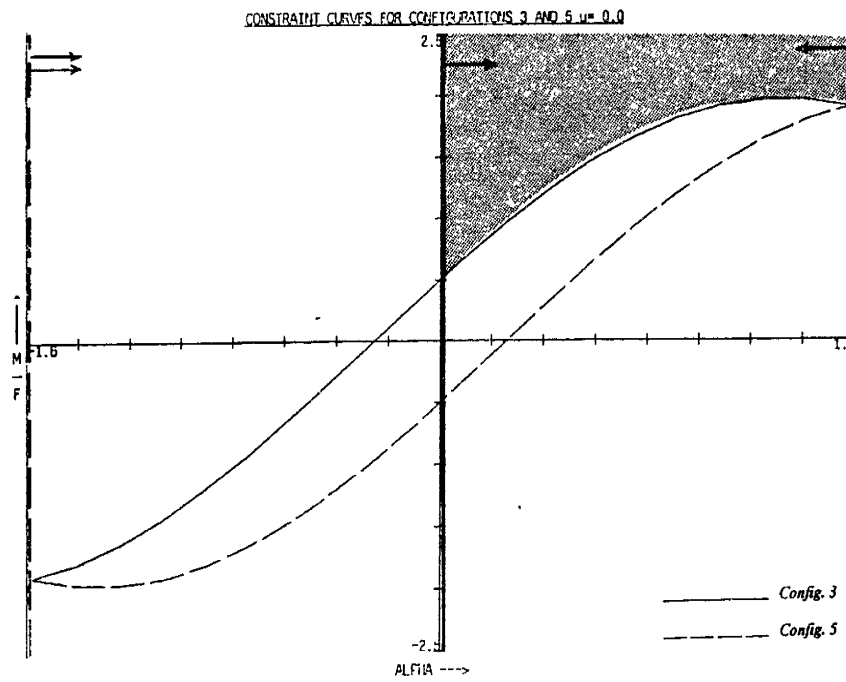


Figure 2.24: Corner Crossing: Various Coefficients of Friction  $\mu$

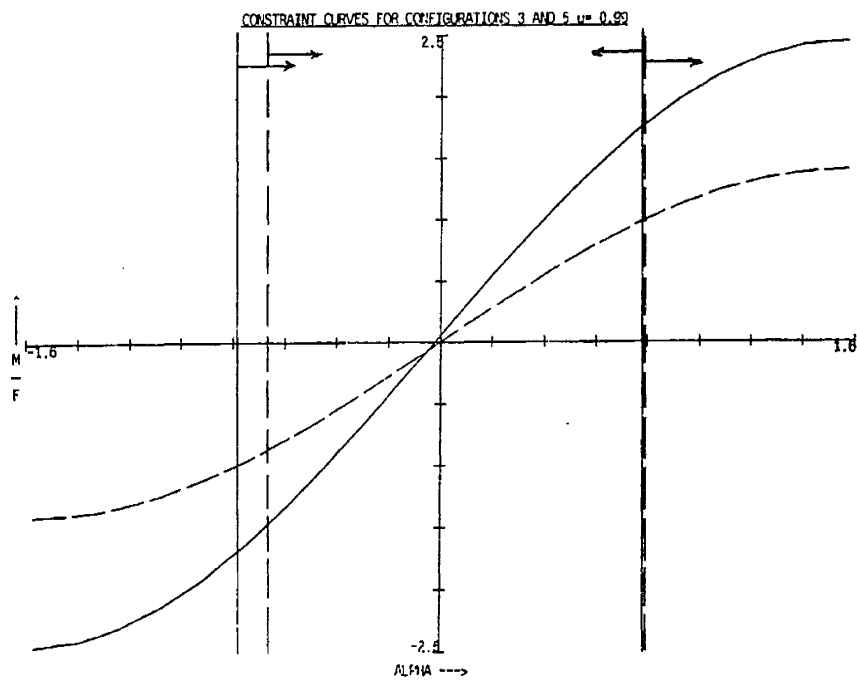
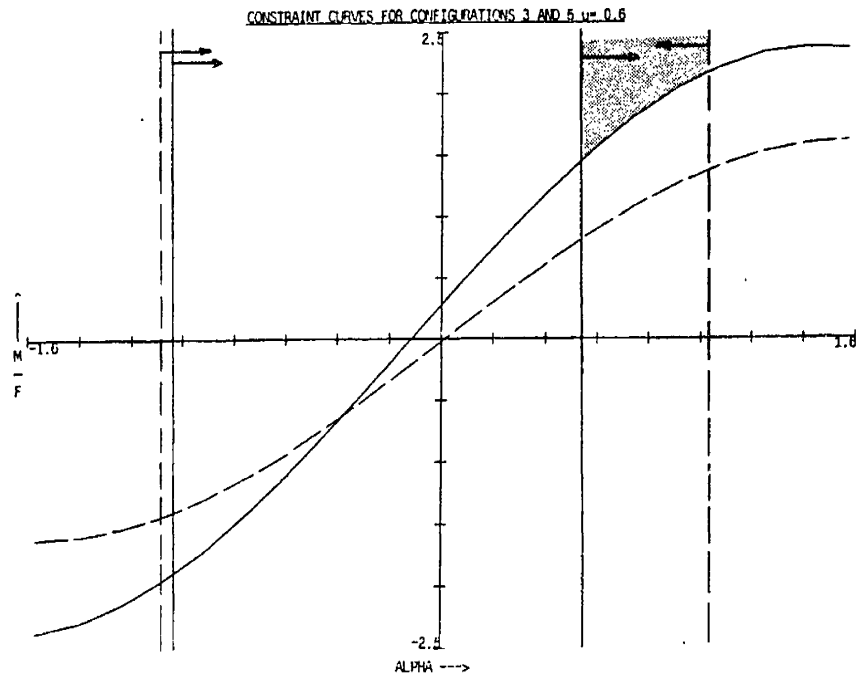


Figure 2.25: Corner Crossing: Various Coefficients of Friction  $\mu$ , (Cont.)



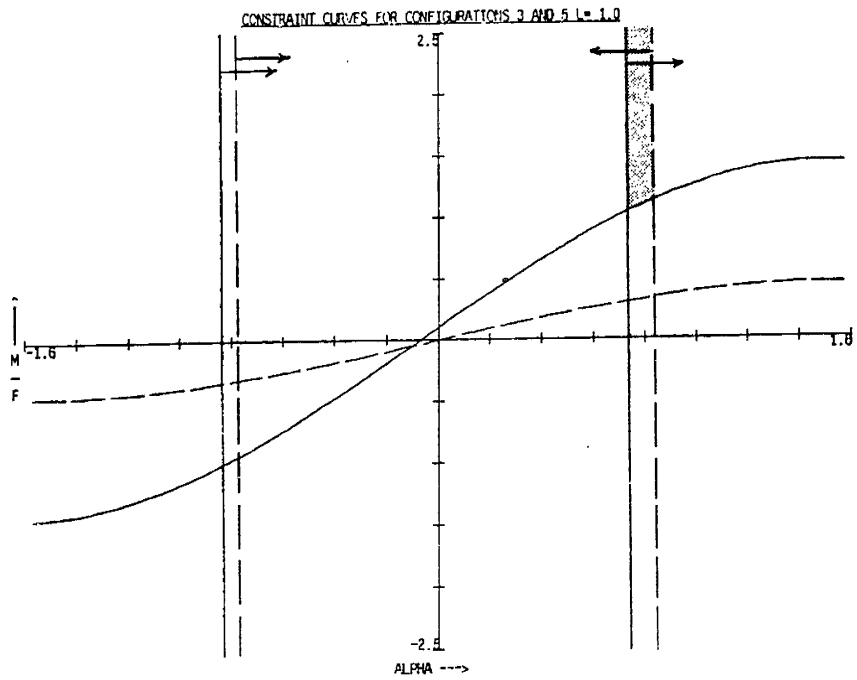
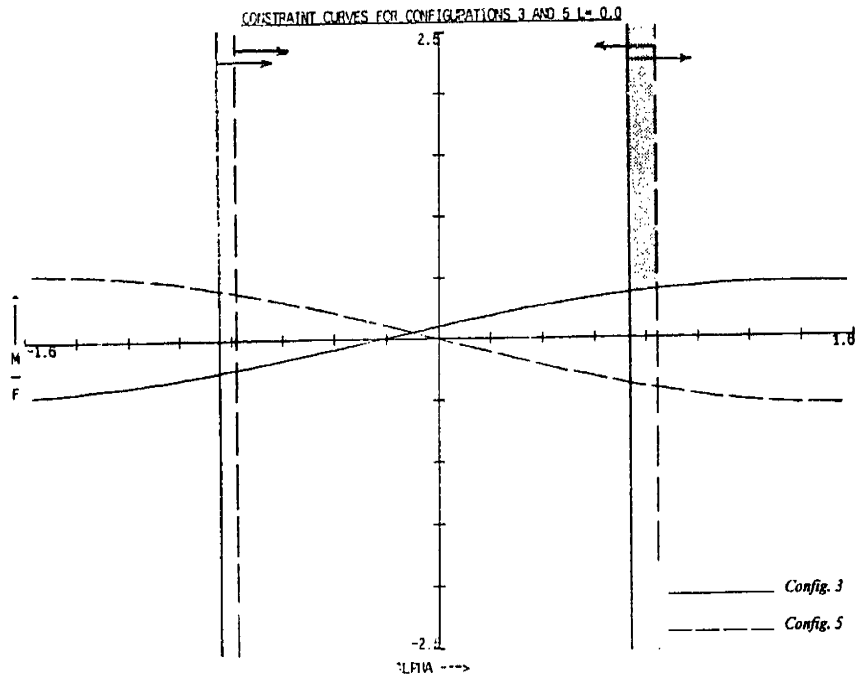


Figure 2.26: Corner Crossing: Various  $L$ 's

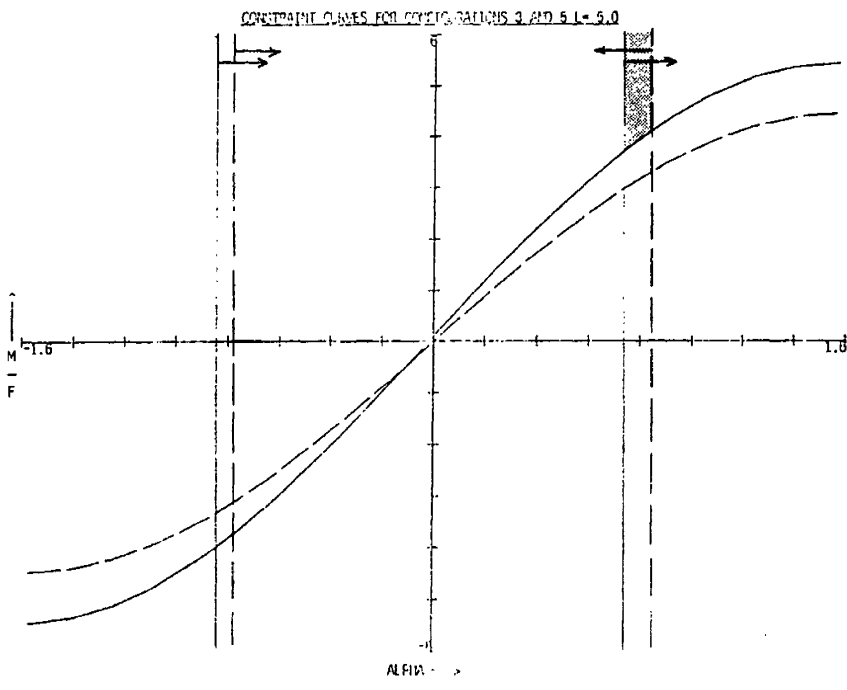
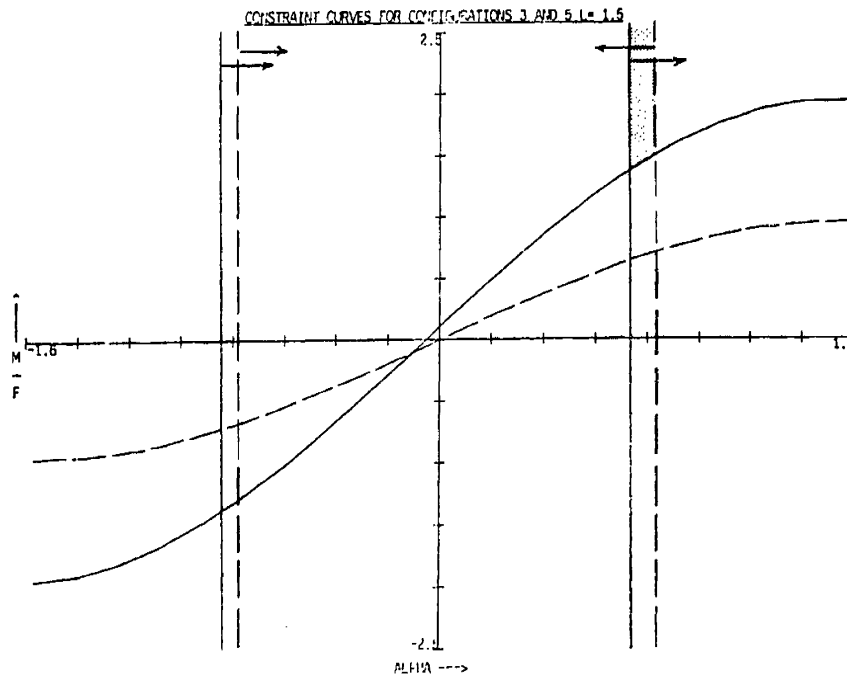


Figure 2.27: Corner Crossing: Various  $L$ 's, (Cont.)

also notice that the value of  $L$  has absolutely no effect on the breaking contact constraints on  $\alpha$ . Thus, the distance  $L$  at which the force  $F$  is applied from the tip of the peg serves only to scale the corresponding value of the minimum applied moment  $M$ .

### 2.4.6 The Resulting Insertion Strategy

Using the results of the previous sections, we can now specify a strategy that will successfully insert the peg into the hole.

- We begin by specifying the initial conditions from which to start our compliant strategy. For the chamferless peg and hole, we begin by tilting the peg relative to the hole and establishing two point contact, as outlined in Section 2.3.1.
- Next, we determine the quasi-static constraints that will allow the peg to slide in two point contact outside of the hole from its initial tilted position to corner crossing. We do this by identifying a single region in the applied force-moment constraint space that covers that portion of the assembly path.
- From corner crossing, we determine the applied forces and moments that will continue to slide the peg in two point contact inside the hole. Since the constraint regions for the case of the peg in the hole become larger as the peg slides deeper into the hole, the corner crossing constraints will represent the most conservative bounds on the allowable force and moment.
- Finally, a single applied force-moment vector is chosen from the resulting single intersecting constraint region that will slide the peg along the entire assembly path.

If all of these conditions can be met, the peg will be guaranteed to slide in two point contact, into the hole, without jamming.

## 2.5 Summary

In this chapter we have modeled the effects of friction and part geometry on the insertion of a two dimensional chamferless peg into a chamferless hole. We have outlined a reliable initial condition that marks the starting point for subsequent assembly motions in the presence of uncertainty. In addition, we have shown how the need for physical modifications to the geometry of parts to aid in their assembly can be avoided in many cases by choosing the proper strategy.

We have chosen the applied force and moment as our primary control variables and represented the constraints that the presence of friction in our system places on the determination of these variables. In particular, we have presented the  $(\frac{M}{F} vs \alpha)$  plane as a visual means in which to represent these frictional constraints. By intersecting constraint regions within this force-moment plane we have provided a means to search for the proper moments and forces that will guide an assembly in sliding motion. We have also seen how the various parameters that describe an assembly affect the range of applied forces and moments that will determine the resulting strategy.

## Chapter 3

# Assembly in Three Dimensions

The previous chapter examined assembly in a two-dimensional planar domain. We now turn our attention to the more general *and considerably more difficult* case of assembly in three dimensions. Our goals in this chapter will be to extend the assumptions and conclusions of the last chapter and from these develop a set of techniques sufficient for planning assembly strategies in three dimensions. As we shall see, many of the factors that affect planar assembly are similarly present in three dimensions, yet modeling these factors will in general be more difficult and the resulting models more cumbersome to manipulate. In addition, factors that did not explicitly appear in the two-dimensional case will also have to be examined.

As a result of these added complexities, the assumptions and simplifications required to adequately model and analyze three-dimensional assembly will be greater in number. Although these additional restrictions indicate the resulting solutions will be of a less general nature, enough generality remains to make such an analysis useful.

We shall use as an example the rectangular peg and hole described in Chapter 1 and shown in Figure 3.1. As noted, the rectangular peg and hole provides a simple yet non-trivial example of a general three-dimensional assembly. Many of the techniques and conclusions drawn from this example can be extended to more complex parts modeled as collections of polyhedra (see Section 1.4).

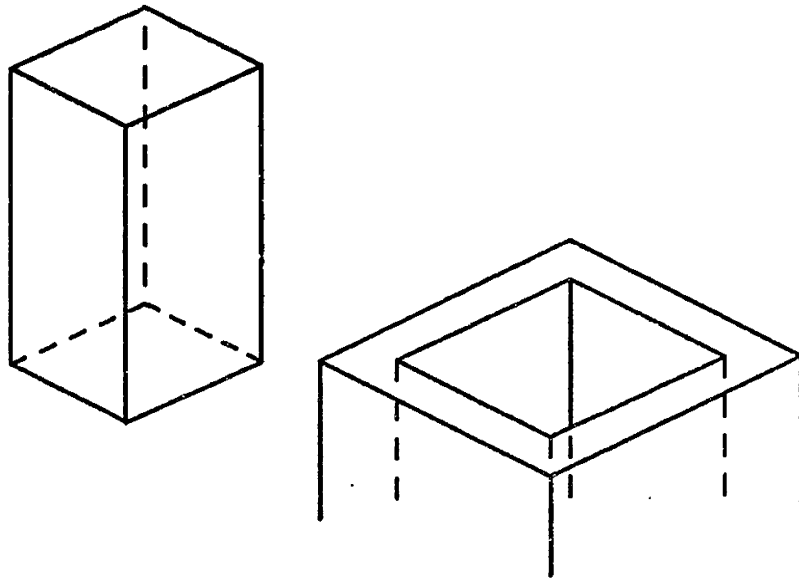


Figure 3.1: The Rectangular Peg and Hole

### 3.1 Brief Overview

The basic approach of planning assembly strategies in three dimensions follows closely that of the last chapter. Namely, a set of possible configurations between parts is identified, and from this set, a trajectory is chosen which moves a part from some initial state to a final desired configuration.

As before, the overall goal will be to reduce the level of uncertainty at each stage in the assembly and thereby reduce the likelihood of premature or unanticipated motion termination. This will be accomplished by initiating contact between parts and sliding along their edges and surfaces, allowing them to act as guides. By establishing and maintaining contact between parts, the degrees of freedom of the system and their associated positional uncertainties will gradually be reduced.<sup>1</sup>

The modes of failure identified in the two-dimensional case will be extended into three dimensions and represented as constraints on the applied forces and

---

<sup>1</sup>See Section 3.2.1.3

moments that are the control variables of the system. Initial conditions will be specified and the requirements for reliably transcending various configurations encountered during assembly will be outlined. In addition, properties of the system that are useful for planning strategies will be identified and applied in the specific example of the rectangular peg and hole.

Geometric models of the parts being assembled will be developed and the governing quasi-static force balance equations derived. Techniques for evaluating and manipulating the resulting solution spaces will be developed, and a means for visualizing the effects of various parameter changes on those solutions will be provided.

Finally, a means of evaluating the resulting strategies in terms of various design criteria will be detailed. In general, the process of establishing configurations and initial conditions, selecting applied forces and moments, and evaluating the resulting strategies will be iterative in nature. As noted in the first chapter, the designer of an assembly strategy, just as any other designer, will have to make a number of assumptions and choices during the design process. Often, these choices will need to be reviewed and revised as more details about the design become apparent.

## 3.2 Outlining a Strategy

An underlying assumption of the strategy development process is that the simplifications, assumptions, and conservative approximations that are made will still result in a viable solution region being identified. One risk of this assumption is that an approach that is *too conservative* may fail to find any solution at all, even if one actually exists. This dilemma of *'keeping the baby and throwing away the bath water'* demonstrates the need to carefully evaluate the assumptions that are made in each step of the design process. With this in mind, we will now outline the assumptions and conclusions that will be made in developing a strategy to insert a rectangular peg into a rectangular hole.

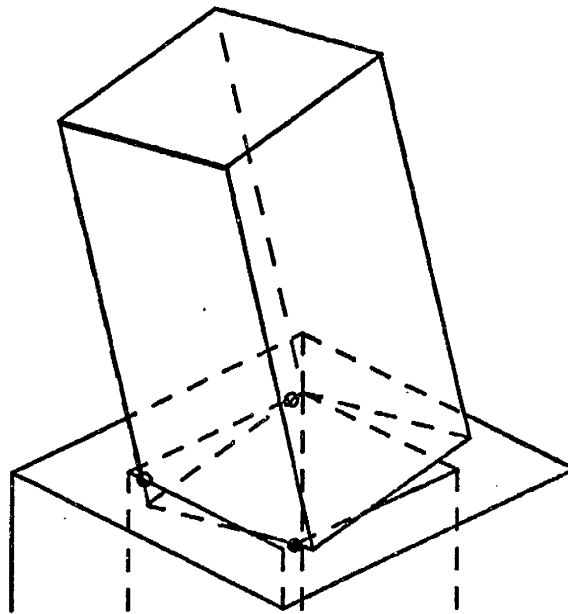


Figure 3.2: Edge-Edge Contact Between Peg and Hole

### 3.2.1 Identifying Useful Properties

The first step in developing a strategy is to identify the important factors that affect the assembly process. Here we will present a list of *Heuristics* that can be applied in the design process. Many of these heuristics result from the conclusions of the last chapter, while others are derived from observations made while examining the parts and subassemblies discussed briefly in chapter one. While some of these rules may not apply to all assemblies, hence the name heuristics, many of them are general enough that they will be useful in most situations encountered.

#### 3.2.1.1 Dominance of Edge-Edge Contact Between Convex Polyhedral Parts

Most of the interactions involving polyhedral parts involve edges contacting edges. In fact, in the case of the rectangular peg and hole, all contacts between the peg and the hole are edge-edge until all four bottom corners of the peg have



cleared the rim of the hole, as can be seen in Figure 3.2.<sup>2</sup> This information will be useful when determining the set of possible configurations of the peg and hole since it will define the class of allowable contacts that determine those configurations.

Since many parts can be represented as a collection of polyhedra, as shown in Chapter 1, this observation has an additional level of importance and generality. As we shall see later, modeling parts in terms of interacting edges provides a convenient framework in which to represent the effects of geometry on an assembly strategy.

### 3.2.1.2 A Reliable Initial Condition

A fine motion strategy will often begin with initial conditions inherited from a series of gross motions required to place a part in or near contact with another part. These initial conditions will have a significant effect on the motions that follow. It is therefore desirable to have an initial configuration, i.e. a set of contacts, which is relatively insensitive to the positioning errors associated with gross motions.

In the case of the rectangular peg and hole, one such configuration is the three edge-edge contact configuration shown in Figure 3.2. In this configuration, slight positioning errors do not result in any significant changes in the resulting motion constraints since the contacts all remain edge-edge. In other words, by tilting and rotating the peg we are establishing contact between large edges of the peg and large edges of the hole. Since it is assumed that the lengths of these edges are considerably greater than any positioning errors we are likely to encounter, we can safely guarantee that the desired contacts will be made. This initial condition is exactly analogous to the technique of tilting the two-dimensional peg relative to the hole to obtain the kinematic equivalent of chamfers (see Section 2.3.1).

---

<sup>2</sup>The one exception to this example is the case of a bottom corner of the peg sliding on the surface surrounding the edge of the hole. Here it is assumed that the peg has been started with one corner already in the hole.

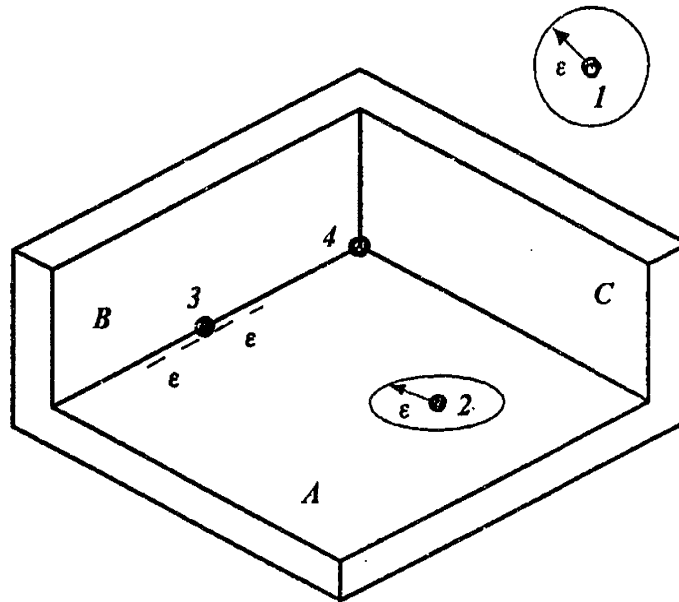


Figure 3.3: Uncertainty of Various Degrees of Freedom

### 3.2.1.3 Relationship Between Degrees of Freedom and Uncertainty

Consider the points shown in Figure 3.3. Point one is above the surfaces, point two is on surface A, point three is along the intersection of surfaces A and B, and point four is in the corner formed by the intersection of all three surfaces. In the first case, the position of point one relative to the surfaces is known within some uncertainty *ball* of radius  $\epsilon$ , where  $\epsilon$  is the maximum allowable *linear* positional uncertainty.<sup>3</sup> In the second case, the position of point two is known within an uncertainty *circle*, also of radius  $\epsilon$ . Since point two is constrained to lie on the horizontal surface A, there is no uncertainty associated with its vertical position. In the third case, the position of point three is known along an uncertainty *line* of length  $2\epsilon$  which lies on the intersection of surfaces A and B. Finally, in the fourth case, the position of point four is known with respect to the surfaces with no uncertainty at all, since it is constrained to lie at the intersection of all three surfaces.

<sup>3</sup>We say *linear* because a point can have no uncertainty associated with an orientation.

This example serves to illustrate the direct relationship between the degrees of freedom of a system and the positional uncertainty associated with those degrees of freedom. To put it another way, the positional uncertainty of a system with  $n$  degrees of freedom can be represented by an  $n$ -dimensional *region*. As the number of degrees of freedom is reduced, the dimensions of the uncertainty region are also reduced.<sup>4</sup> The velocity uncertainty associated with  $n$  degrees of freedom can also be represented in a similar fashion as will be shown in Section 3.2.3.

The fact that positional uncertainty is reduced as the degrees of freedom are reduced is an important consideration when planning an assembly strategy. By reducing the degrees of freedom of a system it is possible to reduce the likelihood of encountering an unanticipated motion terminating configuration, i.e. a configuration that was not anticipated while planning the assembly strategy and that would cause the assembly to jam or otherwise fail. In addition, fewer degrees of freedom implies fewer position (and velocity) parameters to be specified by the designer. This is desirable since one of the assumptions a designer must make during motion planning is what positions and velocities are required to move towards a given goal configuration. As the degrees of freedom are reduced, more of these parameters are automatically specified by the kinematic constraints of the assembly and are hence less ambiguous than those that must be *plucked out of the air* or otherwise determined.

This relationship between the degrees of freedom of a system and its associated positional uncertainty is the primary reason behind the specified requirement that parts must slide along each other during assembly. It should be noted that this requirement is a conservative one since a more general approach would allow contacts to be broken in some instances.

#### 3.2.1.4 Robustness of the Edge-Corner Contact for Constrained Sliding Motions

---

<sup>4</sup>Note that this reduction is equivalent to taking lower dimensional *cross-sections* of the uncertainty region.

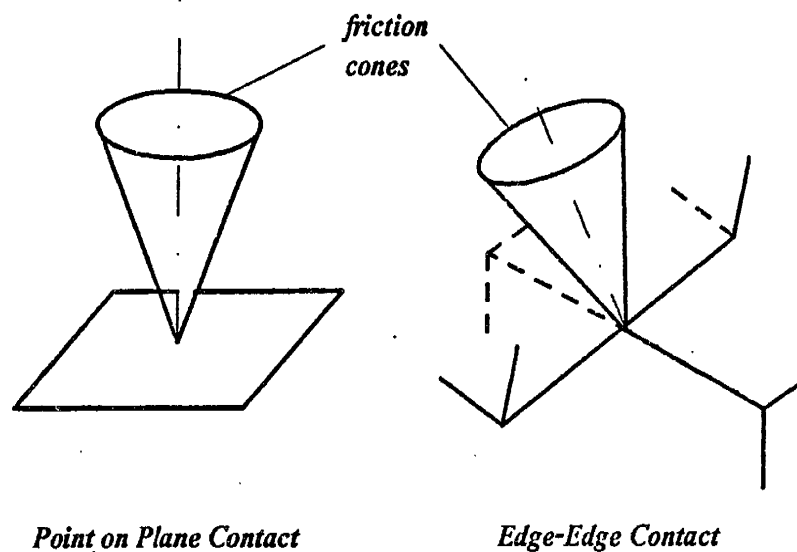


Figure 3.4: The Three-Dimensional One Point Friction Cone

In Section 2.2.1 we defined the one point friction cone as the set of all possible reaction forces that a surface (or edge) could exert at that point. Figure 3.4 shows an extension of the one point friction cone into three dimensions. As we can see, the three-dimensional one point friction cone is a direct and obvious extension of its two-dimensional counterpart.<sup>5</sup> In addition, a one point friction cone can be defined in the interaction between two edges. In this case, also shown in Figure 3.4, the axis of the cone is defined by the common normal of the two edges. Section 3.3.4.1 provides a more detailed explanation of how these normals (both sense and magnitude) are determined.

Figure 3.5 shows the friction cone corresponding to an edge-corner contact. In particular, the contact of an edge of the peg with a corner of the hole. As we can see by comparing Figure 3.5 and Figure 3.4, the corner friction cone is considerably larger than the simple one point friction cone. This indicates that the range of reaction forces possible in an edge-corner contact is considerably

<sup>5</sup>In fact the word 'cone' suggests its three-dimensional form

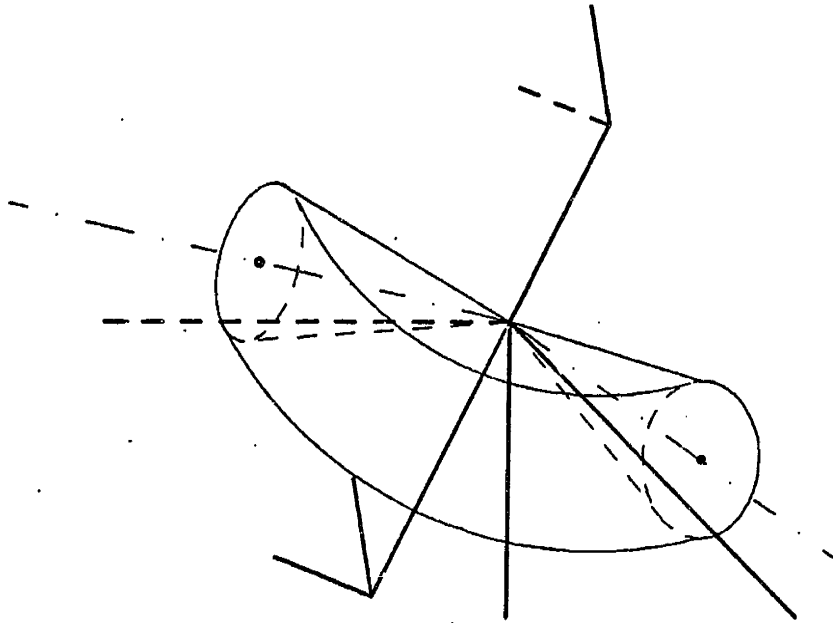


Figure 3.5: The Three-Dimensional *Corner Friction Cone*

greater than for a simple one point (or edge-edge) contact. This is a useful fact to consider in planning a strategy since the larger range of reaction forces means that a larger range of applied forces and moments can be made without breaking contact.<sup>6</sup>

In addition, a single corner contact represents a constraint on two degrees of freedom rather than just one, as in a single edge-edge contact. This can be easily seen by considering a corner as being comprised of two edges. Satisfying the constraint of being on both edges simultaneously, i.e. in the corner, means that two one degree of freedom constraints are being imposed simultaneously. Therefore a single edge-corner contact reduces the total degrees of freedom by two. From the previous section, we know that fewer degrees of freedom means fewer uncertainties to contend with. With these facts in mind, a strategy that contains edge-corner contacts will tend to be more ‘robust’, i.e. less susceptible to the effects of uncertainty.

---

<sup>6</sup>There is also a greater set of forces and moments that can result in jamming, but in developing our strategies we intend to avoid those sets.

### 3.2.1.5 Transition of Constraints at Intersections of Configurations

We have used the term *configuration* a number of times up to this point. In order to describe the interactions between parts in three dimensions, we shall take the term *configuration* to mean a set of contact *types*, i.e. edge-edge, edge-corner, vertex-face, etc.. In addition, each edge, corner, vertex, etc. is specified uniquely, i.e. top right corner of hole, bottom left vertex of peg, etc., as are the contacts between them. For example, the particular three point edge-edge contact shown in Figure 3.2 is an example of *one* possible configuration of the peg and hole under this definition. The peg can have any set of contact positions along the corresponding six edges (three on the hole, three on the peg) and still remain within the given configuration. Therefore a configuration represents an infinite range of possible positions within which a uniquely specified set of contact types is maintained.

In the last chapter we saw that the greatest changes in the sliding and breaking contact solution regions of the 2D peg and hole occurred at a transition of configurations. Namely, when the bottom edge of the peg crossed the top corner of the hole, the curves bounding the solution space shifted discontinuously. In the configurations both preceding and following the corner crossing point, i.e. out of the hole and in the hole respectively, the solution space regions were comparatively well behaved and changed only slowly with the position of the peg. It was for this reason that most of the analysis was centered around this transition point (see Section 2.4.5.2).

In the three-dimensional case, we shall expect the same sort of behavior from the corresponding solution regions. As we shall see later, the transition points between configurations will experience the greatest changes in the resulting solution space. We shall use this assumption to limit our search for solutions to the regions immediately surrounding changes in configuration. In addition, we shall assume that the solution regions change only slowly both before and after these transitions. Therefore an applied force and moment that works just before or just after a transition point will be assumed to work over a range of

positions within the vicinity of those configuration transitions.<sup>7</sup> This assumption is important since the modeling and analysis procedure is discrete in nature in that it represents a force and moment balance only at a particular position within a configuration. If we can extend the analysis to include the contact positions surrounding the ones explicitly specified then we can reduce the number of these positions that must be examined.

### 3.2.2 Determining the Set of Possible Contact Cases

Having listed our assumptions and conclusions about three-dimensional assembly, we now begin the process of actually developing a strategy for inserting a rectangular peg into a rectangular hole. We start this process by determining a set of contact configurations from which we can construct a path, i.e. a sequence of positions of the peg and hole within the chosen configurations, that will connect an initial state to a desired end state.

We will begin by considering all of the possible configurations given an allowable set of contact types and from these select a subset large enough to specify a complete trajectory. As we shall see, the set of configurations that are possible grows rapidly with the geometric complexity of the parts being modeled.

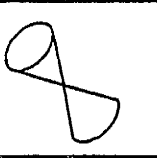
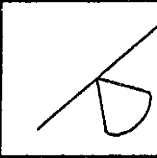
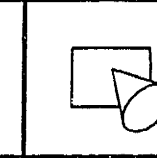
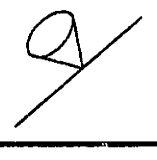
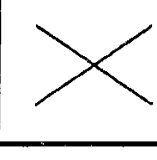
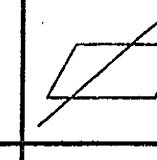
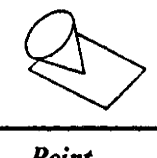
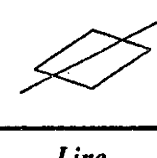
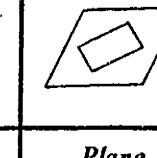
#### 3.2.2.1 Types of Contact Considered

Figure 3.6 shows the set of nine single contact types possible between two rigid bodies [Salisbury 82]. Of this set of contacts, three are unstable and only appear as transient states. These three are the point on point, point on line, and line on point contacts. We also see that some contacts are simply the inverse of another, namely the point on plane vs. the plane on point, and the line on plane vs. the plane on line. Each of these pairs represent identical constraints and hence can be combined. This leaves us with four distinct types of contact. These are:

- *line on line* contact,
- *point on plane* contact,

---

<sup>7</sup>Just how broad these ranges are will be determined later.

|                                  |   |   |  |                                  |
|----------------------------------|---|---|--|----------------------------------|
| <i>PART A</i>                    |   |   |  |                                  |
| <i>P<br/>A<br/>R<br/>T<br/>B</i> |  |  |  | <i>P<br/>o<br/>i<br/>n<br/>t</i> |
|                                  |  |  |  | <i>L<br/>i<br/>n<br/>e</i>       |
|                                  |  |  |  | <i>P<br/>l<br/>a<br/>n<br/>e</i> |
|                                  | <i>P<br/>o<br/>i<br/>n<br/>t</i>  | <i>L<br/>i<br/>n<br/>e</i>  | <i>P<br/>l<br/>a<br/>n<br/>e</i>   |                                  |

*Note: The 'cones' shown here represent solid objects, not friction.*

Figure 3.6: General Contact Types

- *line on plane* contact, and
- *plane on plane* contact.

Of these remaining types of contact, the plane on plane contact is comparatively rare in the rectangular peg and hole since it requires the two planes to be exactly parallel, by definition, which can only occur when the axes of the peg is exactly aligned with those of the hole.<sup>8</sup> The line on plane contact is also less common than the first two cases in the rectangular peg and hole example.

We have now left ourselves with two basic types of contact with which to represent the set of possible contact configurations of the rectangular peg and hole. These are the *line on line*, which we shall refer to as edge-edge, and the *point on plane* contacts, which we shall refer to as vertex-face. We note that the other types of contact which we have just eliminated can still be represented as combinations of these two contacts types. For example, a line on plane contact

<sup>8</sup>In other words all of the edges have been cleared and the peg has only one degree of freedom along its vertical axis.



could be represented as two vertex-face contacts or a plane on plane contact as three vertex-face contacts acting simultaneously, etc.. Therefore we have not completely eliminated the possibility of including these contacts in our strategy.<sup>9</sup> We shall also consider the corner contact case mentioned earlier to be included as part of this extendable representation.

### 3.2.2.2 The Complete Set of Contact Configurations

From the allowable contacts just derived we can construct a list containing the complete set of contacts between the rectangular peg and hole. To do so it is first useful to specify a notation with which to represent these configurations. It should be noted that the notation we shall use is not the only possible or even the best notation for describing contacts, but is intended only to serve as a consistent format in which to combine contact types. First the peg and hole are separated and the contacts on each listed separately. For the peg, the contacts are noted as:

- SE for a Side Edge of the peg,
- BE for a Bottom Edge of the peg, and
- V for a Vertex of the peg (bottom vertex).

In addition, the specification of the number of edges or vertices involved is given as well as the relationship to similar edges or vertices on the same part. Namely, A for adjacent and O for opposite. So for example BE2O refers to a contact involving two opposite bottom edges of the peg. The Contact types for the peg are:

|      |      |     |
|------|------|-----|
| SE1  | BE1  | V1  |
| SE2A | BE2A | V2A |
|      | BE2O | V2O |

For the hole, the notation is:

---

<sup>9</sup>See also Section 2.4.2.

- **TE** for a Top Edge of the hole, and
- **S** for a Side of the hole (inside).

As for the peg, we specify the number of each contact type included and the relationship between them, i.e. **A** for adjacent and **O** for opposite. So the notation **TE2A** refers to a contact involving two adjacent top edges of the hole. The contact types for the hole are:

TE1    S1  
 TE2A   S2A  
 TE2O  
 TE3  
 TE4

By combining the types of contact listed and matching the contacts of the hole to those of the peg, the total number of contact types possible for the peg and for the hole can be determined. An assumption made during this combination process is that four is the maximum number of single independent contacts allowed. A few relatively straightforward rules are used to govern this process. They are:

- The number of contacts of the peg equal the number of the corresponding contacts of the hole.
- A vertex of the peg can only contact a face of the hole (vertex-face contact).
- An edge of the peg can only contact an edge of the hole (edge-edge contact).

Applying these rules to the contact types listed, we obtain a list of 23 contact types. However, in order to specify the complete list of *configurations* we must further specify exactly what edges are in contact with what other edges and what vertices are in contact with what faces. For example, to determine each configuration of the contact case **SE1:TE1** (*contacts of the peg : contacts of the hole*), we determine all possible combinations of these single edges possible. For

this case there are  $4 \times 4$  or 16 possible combinations. Therefore the SE1:TE1 case has 16 associated configurations. A similar process can be used to determine the configurations for the rest of the contact cases generated. Such an analysis produces 1060 possible configurations of the rectangular peg and hole, given the two basic contact representations of edge-edge and vertex-face.

As we can see from the previous development, the number of possible contact configurations for a pair of relatively simple three-dimensional parts such as the rectangular peg and hole is quite large, even after making a number of simplifications regarding the types of contact allowed.<sup>10</sup> One can imagine the number of configurations that would have to be modeled for a more geometrically complicated pair of parts, such as those shown in Chapter 1. From this type of analysis it seems clear that just mindlessly generating and analyzing all of the possible configurations would not be the way to approach the problem. Instead, as was the case for the two-dimensional assembly of Chapter 2, it makes more sense to select a reasonable subset of configurations that will sufficiently describe an assembly path. In Section 3.3.2 we will present such an approach.

### 3.2.3 The Independent Parameters Specifying an Assembly State

In the second chapter we outlined the assumptions underlying a quasi-static model of assembly. Among these were that the effects of velocities and accelerations on the resulting reaction forces were negligible. In fact we considered these terms to be zero and assumed the peg to be in a state of impending motion. Implicit in this assumption was the fact that if the peg *were* to move, it would do so in a pre-determined direction. This sense of direction of the impending motion allowed us to determine what direction to assign the frictional reaction forces at the points of contact. In fact what we were doing was specifying a velocity of the peg *in sense only*.

When we introduced the friction cone in Section 2.2.1 we indicated that the frictional force opposed the motion (or impending motion) of the point of

<sup>10</sup>Compare this to the 14 configurations of the two-dimensional peg and hole of Chapter 2.

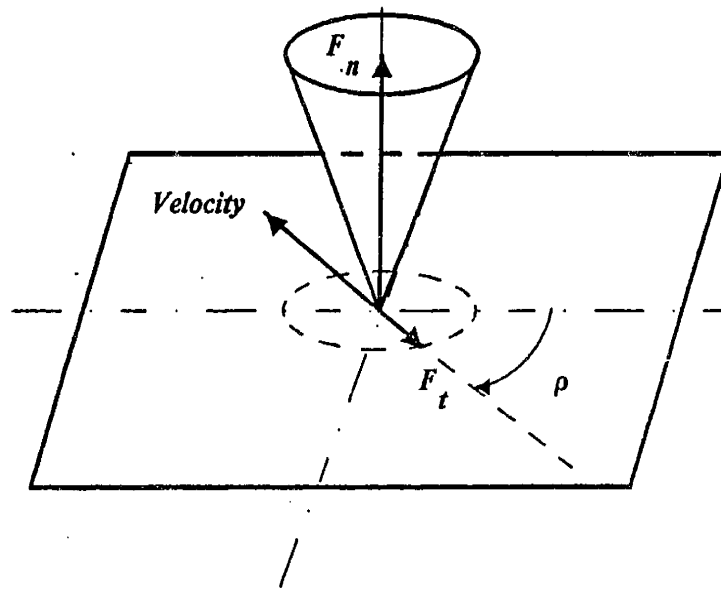


Figure 3.7: Velocity Related to Friction

contact. This sense of motion allowed us to determine which way the frictional component of force should point. In the three-dimensional friction cone this is still true. However, as shown in Figure 3.7, the sense of the frictional component of the reaction force is no longer a binary choice. For the three-dimensional friction component to be specified, the direction of the velocity of the contact point must be specified in the plane. Again this contact velocity is in sense only, but now its specification requires an additional (orientation) parameter, in this case  $\rho$ . As before, the frictional component of the reaction force will be opposite in sense to this velocity.

For a quasi-static analysis, the parameters required to specify the state of an assembly are the position of the peg relative to the hole and the sense of the velocity of the peg relative to the hole. The number of parameters required to specify the position of the peg is equal to the number of degrees of freedom remaining in a given configuration. If the velocity of the peg is considered in sense only, i.e. a vector of unit magnitude, then the number of parameters required to specify it is equal to the degrees of freedom minus one. Therefore, for a given assembly state the number of independent parameters that must be specified by the planner is given by:

$$N_{param} = 2 \times N_{dof} - 1 \quad (3.33)$$

where  $N_{param}$  is the number of parameters and  $N_{dof}$  is the number of degrees of freedom of the system. As an example of this consider again the two-dimensional peg and hole. For the peg in two point contact the number of parameters was  $2 \times 1 - 1 = 1$ , namely the position of the peg (in terms of either the angle of tilt or the insertion depth).<sup>11</sup>

Given the addition of unit velocity to our list of parameters specifying an assembly state, we can again see the relationship between the number of these parameters and the number of associated uncertainties in our system. Specifically, we can extend our conclusions regarding positional uncertainty from Section 3.2.1.3 and state that the *total* number of uncertainties associated with a

<sup>11</sup>As mentioned earlier, the velocity required only a binary ( $\pm$ ) specification and therefore did not require any extra parameters.

given assembly state (assuming a quasi-static model) is equal to the number of parameters necessary to specify that state,  $N_{param}$ . Therefore, each time we remove a degree of freedom from our system, we are removing two associated uncertainties. This lends further support to our strategy of reducing the degrees of freedom of our system wherever possible.

### 3.2.4 Determining Velocity Trajectories Between States

In the overall task of planning an assembly trajectory, the unit velocity of the peg relative to the hole will serve the dual purpose of specifying the sense of the frictional forces at the points of contact as well as specifying the motions that will be used to connect the various contact configurations. In this second function, the unit velocity vector will (just as the position parameters will) represent certain bounds on the set of applied forces and moments that will move the peg between the desired configurations. Figure 3.8 shows a representation of this bounding function in terms of a velocity *error cone*. The positional uncertainty of point  $P_1$  is again represented as a ball of radius  $\epsilon$ . From this uncertainty ball we extend a cone to the goal region  $P_{goal}$ , which represents the desired range of goal positions of our system. This *goal region* may be the final desired position of the peg or some intermediate goal in the overall trajectory. Velocity vectors originating in  $P_1$  and pointing towards  $P_{goal}$  whose directions lie on or within the bounds formed by the edges of the cone represent those unit velocities that are guaranteed to reach the goal region.<sup>12</sup> Similarly, unit velocities that lie outside the bounds of this cone may not always end in a desired configuration (we recall that, by definition, a unit velocity represents a *sense* only and has no associated magnitude).

Figure 3.8 also illustrates an interesting relationship between the uncertainties associated with position and those associated with unit velocity. Given a positional uncertainty ball at position  $P_1$  and an allowable range of positions in  $P_{goal}$ , the width of the velocity cone that connects them will be determined

---

<sup>12</sup>See [Erdmann 84] for a more formal description of the error cone as related to fine motion planning.

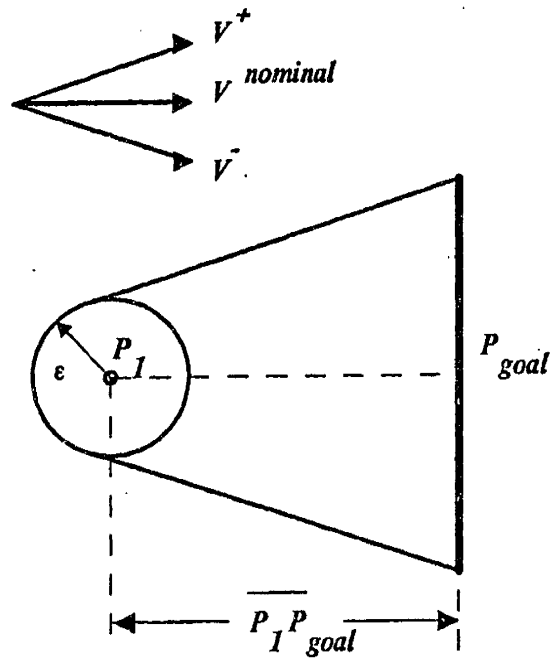


Figure 3.8: The Velocity Error Cone

by the distance between the two states. Therefore a unit velocity between two regions that have a large separation will be more constrained, i.e. fall within a narrower range, than a unit velocity between two regions that are closer together, assuming the relative sizes of the two regions remain constant. This relationship between the distance of a move and the size of the associated uncertainty regions places a lower bound on the number of discretely modeled positions that will sufficiently describe an assembly path.

In this discussion we have used a symmetric cone to represent the bounds on unit velocity for illustrative purposes only. Although the actual unit velocity bounds between given states will in general be non-symmetric and non-linear functions of position, the qualitative conclusions will still be valid.

### 3.2.5 Selecting the Proper Forces and Moments

As was the case with the two-dimensional peg and hole, the control variables of the three-dimensional rectangular peg and hole will be the applied forces and moments. Aside from the initial conditions and states describing an assembly path, the only variables we shall be specifying will be the forces and moments applied to the peg. Given this, we shall want to represent the constraints of our system: geometry, tolerancing, friction, etc., in terms of constraints on the applied forces and moments. We shall do this in a way similar to that of the last chapter, namely, we shall construct a *force-moment space* in which we shall search for solution regions. These regions will determine the range of forces and moments that are guaranteed to move the peg between desired configurations with a minimum chance of premature motion termination.

The two major constraints that we shall be representing in this force-moment space are again jamming and breaking contact. Jamming as a mode of failure of an assembly was outlined in Chapter 1 and treated in detail in Chapter 2. Breaking of contact between two edges or surfaces, although not specifically in itself a mode of failure, represents an increase in the degrees of freedom of our system and, given the relationship to uncertainty, is considered undesirable.

Searching for a set of allowable forces and moments in the last chapter



amounted to graphically finding intersection regions between various constraint curves in a two-dimensional ( $\frac{M}{F}$  vs  $\alpha$ ) space. We shall adopt a similar procedure in this chapter but, given the greater complexity and size of the new force-moment space, we shall have to adopt new techniques for breaking the solution regions down into more manageable form.

### 3.3 Modeling and Analysis

Given our outline of Section 3.2, we now begin the process of modeling the rectangular peg and hole, analyzing these models, and finding acceptable values for the applied forces and moments. Most of the assumptions and conclusions developed up to this point will be applied to these new models. As they become necessary, we shall note any new assumptions we make.

#### 3.3.1 Geometric Representation of Parts

As we have mentioned a number of times, geometry will be a major factor in determining the behavior of a system during assembly. The models we use in developing an assembly strategy must therefore be sufficiently flexible to allow easy representation of this important factor. Figure 3.9 shows the rectangular peg and hole with the three edge-edge contacts of the configuration from Section 3.2.1 indicated on both. The dimensions of the peg: length, width, and height, have been assigned the variables:  $l$ ,  $w$ , and  $h$  respectively. We have assigned to the peg a right handed coordinate frame relative to which we can specify any point on the peg. The  $z$ -axis of the coordinate frame lies along the centerline of the peg and the  $x$ -axis and  $y$ -axis are aligned with the sides of the peg as shown. The origin of the frame has been displaced along the centerline by a distance  $L$  from the bottom of the peg.<sup>13</sup> In this example there are three points of contact between the peg and the hole which are numbered as shown. The positions of these points in the peg's frame are represented by the position vectors  $\vec{R}_1$ ,  $\vec{R}_2$ , and  $\vec{R}_3$  respectively.

---

<sup>13</sup>Notice that the value of  $L$  may be greater or less than the value of  $h$ .

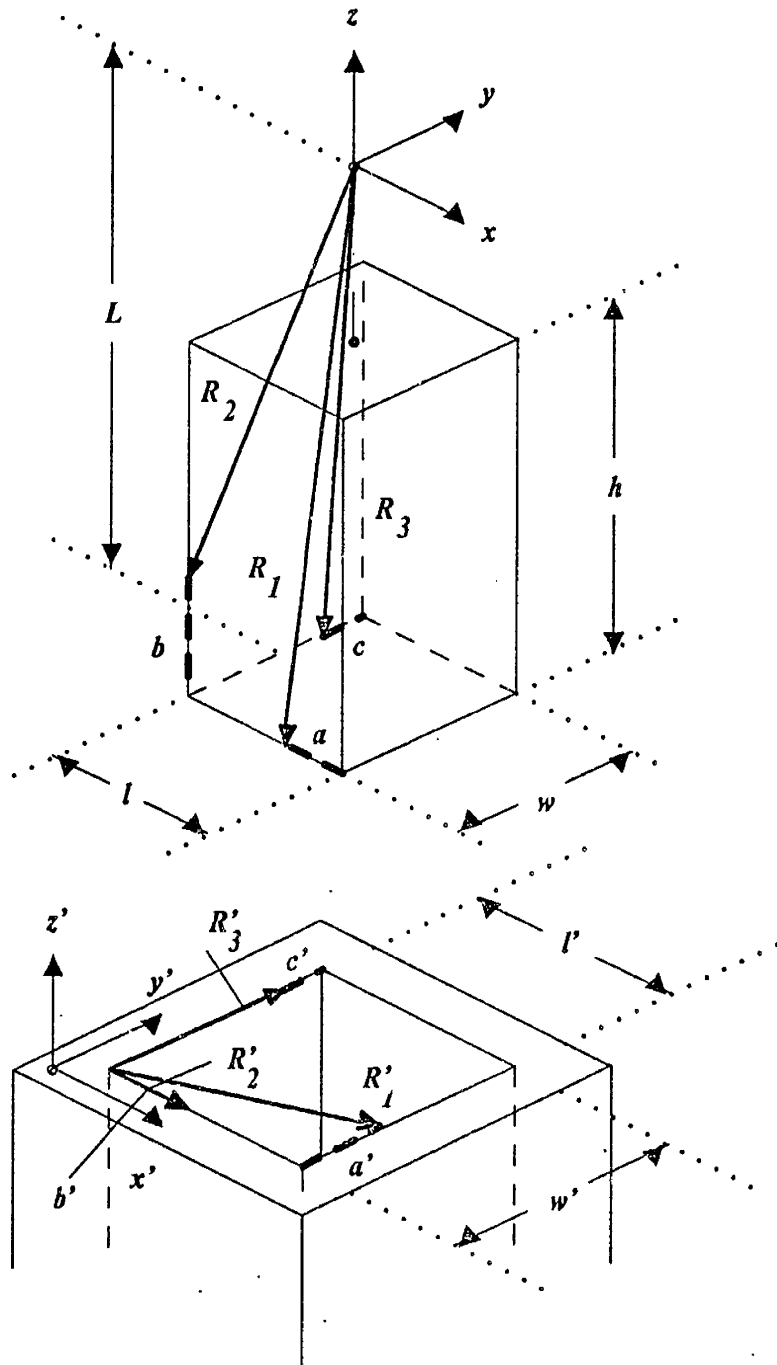


Figure 3.9: Representation of Part Geometry

The dimensions of the top of the hole, length and width, have been assigned the variables  $l'$  and  $w'$  respectively. Since we are primarily concerned with the problem of having the peg enter the hole and not how far it can then be inserted, we shall not bother to specify the depth of the hole. Again we have assigned a right handed coordinate frame to the hole in which we can specify points of contact. The  $z'$ -axis points vertically out of the hole and the  $x'$ -axis and  $y'$ -axis are aligned with the edges of the hole. The origin of the frame is located in the corner shown in Figure 3.9 and lies in the plane formed by the top edges of the hole. The three points of contact relative to the hole coordinates are then represented by the vectors  $\vec{R}'_1$ ,  $\vec{R}'_2$ , and  $\vec{R}'_3$  respectively.

By aligning the coordinate frames with the edges of the peg and the hole we are able to take advantage of the simplicity inherent in representing edges of a right parallelepiped. Namely, each edge-edge contact can be represented by a position vector with only one independent unknown. For example, the three components of the position vector  $\vec{R}'_1$  are:  $l/2 - a$  along the  $x$ -axis,  $-w/2$  along the  $y$ -axis, and  $-L$  along the  $z$ -axis. We see that the  $y$  and  $z$  components of  $\vec{R}'_1$  are simply constant dimensional parameters of the peg. Only the  $x$  component contains an unknown variable  $a$ , which in this case represents the displacement of contact point one along the edge which is aligned with the  $x$ -axis. Therefore any motions of the peg within the configuration will result in changing only the value of  $a$  for vector  $\vec{R}'_1$ .

Vertex-face contacts, although not as easily represented as edge-edge contacts, can also be represented in a relatively straightforward way. For a point of contact on a vertex of the peg, the components of the position vector contain no unknowns since the position of the vertex relative to the rest of the peg remains constant. The corresponding position vector in hole coordinates will lie in the plane formed by the side of the hole in contact and will therefore contain an unknown in those two of its three components that lie in the plane.

### 3.3.2 Selecting a Subset of Configurations

From the list of configurations developed in the last section we will now select a subset from which to build an assembly strategy. As was mentioned in the introduction to this chapter, the development of an assembly strategy is an iterative process. Therefore, the set of configurations initially chosen may prove to be insufficient and therefore need to be revised later in the design process. In fact we shall see some specific examples of how the geometry of a specific case under study, including factors such as tolerancing, will have a significant impact on the final configurations chosen to describe an assembly path.

In generating our configuration subset we shall make use of the heuristics of Section 3.2.1. In particular, we shall select configurations that tend to increase the number of constraints, contain edge/corner contacts, and require only small motions to reach the next configuration in the assembly path. This last criterion, i.e. that of considering only small motions between configurations, follows directly from our discussion of the relationship between unit velocity and position uncertainties. It is designed to help us to reduce the associated errors of sampling only a few points per configuration to represent the path of motion through the configuration. In other words, it is intended to support our assumption that the solution spaces will not change appreciably between configuration transition points. Our aim then is to make the assembly problem more tractable by analyzing as sparse a set of actual positions as possible.

#### 3 pt. Edge-Edge Contact: [Case 1]

The first contact configuration we shall use, shown in Figure 3.10, is the 3 pt. edge-edge contact outlined in Section 3.2.1.4. As was mentioned earlier, this edge-edge configuration provides a very robust initial condition for our strategy since it requires a low positioning resolution to achieve reliably. With this configuration providing the starting point for the following configurations that will be achieved through compliant motions, the positioning resolution of our system becomes less important in planning our strategy. Since each edge-edge contact removes one degree of freedom from the peg, it has three remaining degrees of freedom in this configuration. In terms of our contact notation of

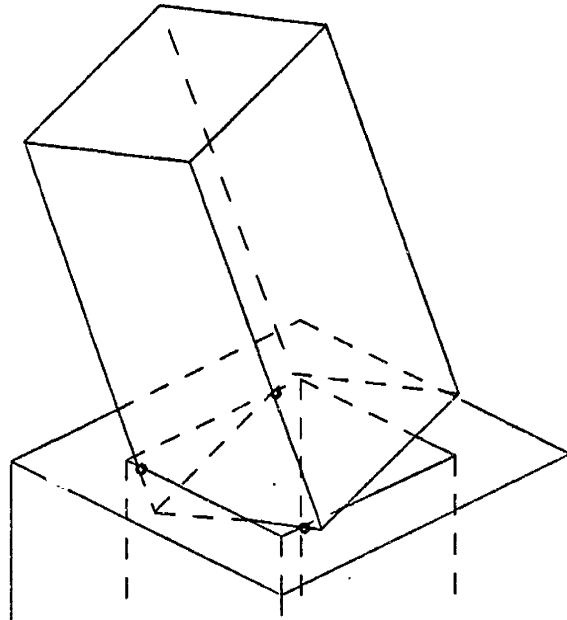


Figure 3.10: Case One

Section 3.2.2.2, Case 1 is of the form (SE1.BE2A:TE3).

**Bottom Edge in Corner: [Case 2]**

From the three edge-edge configuration we shall next establish contact between a corner of the hole and a bottom edge of the peg. This contact, shown in Figure 3.11, represents a reduction from three to two degrees of freedom for the peg. Case 2 also represents a transition of configurations since the bottom edge of the peg is now touching a new edge of the hole, as well as the previous one. In other words, this transition case represents two configurations of the peg and hole occurring simultaneously. These configurations are both also of the form (SE1.BE2A:TE3), where a different top edge of the hole is specified in each.

**Bottom Edge-Corner, Side Edge-Corner: [Case 3]**

From Case 2, we will establish another edge-corner contact, this time between an adjacent side edge of the peg and an adjacent corner of the hole, as shown in Figure 3.12. Here again, Case 3 represents a transition of configurations. The first two configurations of Case 2 are still valid, and to them we have

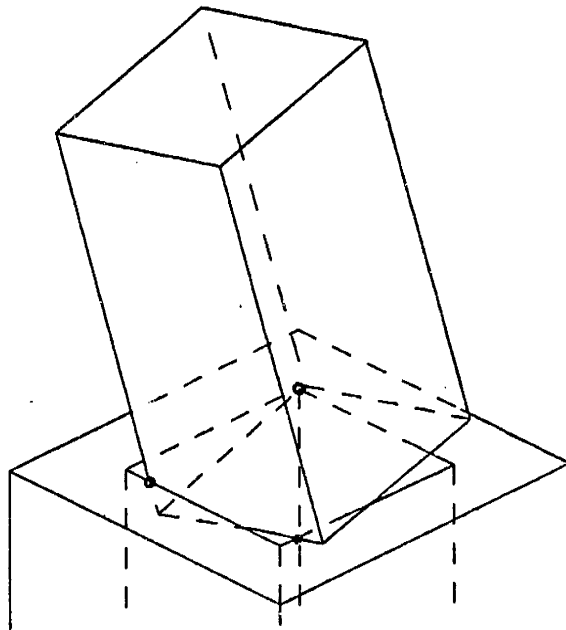


Figure 3.11: Case Two

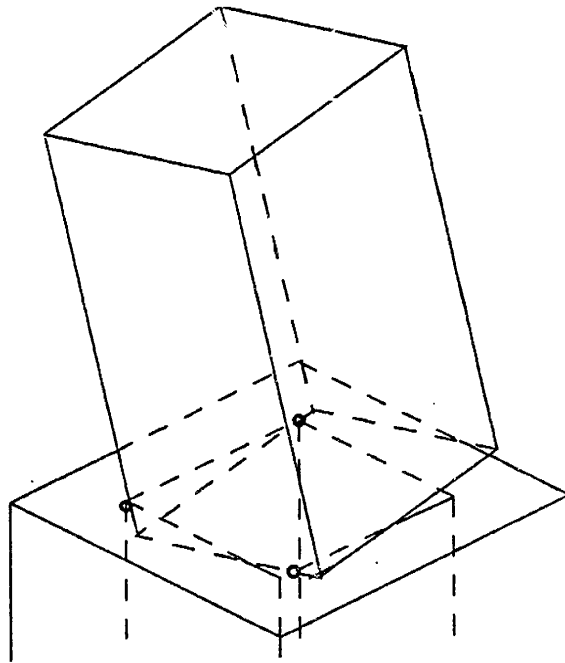


Figure 3.12: Case Three

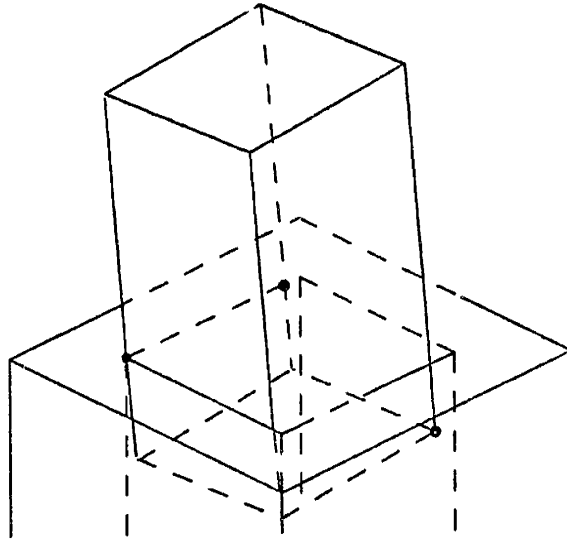


Figure 3.13: Case Four

added a third simultaneously occurring configuration. The models used to define the equilibrium of this case will represent a combination of models that would represent those three configurations, each defined separately. Here again the three configurations are all of the form (SE1.BE2A:TE3).

#### **Peg in the Hole: [Case 4]**

From Case 3, we will slide until the peg has cleared the top edges of the hole. As the peg slides in to the hole, contact will be established between a vertex of the peg and a face of the hole, as shown in Figure 3.13. The configurations used to describe Case 4 are of the form (SE2A.V1:TE2A.S1) and (SE1.V1:TE1.S1).

### **3.3.3 Kinematic Constraints**

We recall that an unconstrained three-dimensional rigid body can have a maximum of six degrees of freedom and will therefore require six independent parameters to describe its position uniquely. As we reduce the degrees of freedom by imposing motion constraints, we reduce the number of these parameters that

may be specified independently. We shall now develop representations for these constraints for the set of configurations just chosen.

### 3.3.3.1 Coordinate Transformations

Given a pair of coordinate frames, one moving and the other fixed, we can determine a transformation that relates the translational and rotational displacement of the moving frame relative to the fixed reference frame. This transformation is of the form:

$$\vec{U}^{fixed} = [C] \vec{U}^{moving} + \vec{R}_0 \quad (3.34)$$

where  $\vec{U}^{fixed}$  represents the axes of the *fixed* reference frame,  $\vec{U}^{moving}$  represents the moving coordinate frame,  $[C]$  is a  $3 \times 3$  rotational transformation matrix representing the rotational displacement of the two coordinate frames in the fixed frame, and  $\vec{R}_0$  is the column vector representing the linear displacement of the origins of the two frames in the fixed frame. Therefore to represent any vector  $\vec{R}_m^{moving}$  of the moving frame in the fixed frame we write:

$$\vec{R}_m^{fixed} = [C] \vec{R}_m^{moving} + \vec{R}_0 \quad (3.35)$$

where  $\vec{R}_m^{moving}$  is any vector in the *moving* frame and  $\vec{R}_m^{fixed}$  is the corresponding vector in the *fixed* reference frame.

For the purposes of our model we shall choose the fixed coordinate frame to be that of the peg and the moving frame to be that of the hole. This may seem strange since it is the peg that shall actually be moving while the hole remains stationary, but since we shall be specifying our applied forces and moments relative to the peg it makes more sense to choose the peg to be our reference. Therefore all of our specified motions will be of the *hole moving around the peg*.<sup>14</sup>

We shall choose a *roll-pitch-yaw* representation of rotation for convenience. Namely, a rotational displacement of the hole relative to the peg will consist of a rotation  $\theta_{x'}$  about the  $x'$ -axis of the hole, followed by a rotation  $\theta_{y'}$  about the

---

<sup>14</sup>The task of transforming these motions for the purpose of implementation consists a straightforward inversion and will be detailed in Chapter 4.



$y'$ -axis of the hole, followed finally by a rotation  $\theta_{z'}$  about the  $z'$ -axis of the hole. We can represent these rotations in matrix form by:

$$ROT(x', \theta_{x'}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_{x'} & -\sin \theta_{x'} \\ 0 & \sin \theta_{x'} & \cos \theta_{x'} \end{bmatrix} \quad (3.36)$$

for the  $x'$  rotation:

$$ROT(y', \theta_{y'}) = \begin{bmatrix} \cos \theta_{y'} & 0 & \sin \theta_{y'} \\ 0 & 1 & 0 \\ -\sin \theta_{y'} & 0 & \cos \theta_{y'} \end{bmatrix} \quad (3.37)$$

for the  $y'$  rotation: and

$$ROT(z', \theta_{z'}) = \begin{bmatrix} \cos \theta_{z'} & -\sin \theta_{z'} & 0 \\ \sin \theta_{z'} & \cos \theta_{z'} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.38)$$

for the  $z'$  rotation.

To determine the elements of the rotation matrix  $[C]$  we carry out the following matrix multiplications:

$$[C] = ROT(z', \theta_{z'}) ROT(y', \theta_{y'}) ROT(x', \theta_{x'}) \quad (3.39)$$

where the matrices are multiplied in reverse order, namely the second times the third, and then the result by the first. We do this because the rotation matrices operate on vectors by *post multiplication*, hence the vector being rotated would be placed at the far right of the above expression to be operated on by the  $ROT(x', \theta_{x'})$  matrix first. What we end up with is the matrix:

$$[C] = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (3.40)$$

where the elements have the values:

$$\begin{aligned}
C_{11} &= \cos \theta_{y'} \cos \theta_{z'} \\
C_{12} &= \sin \theta_{x'} \sin \theta_{y'} \cos \theta_{z'} - \cos \theta_{x'} \sin \theta_{z'} \\
C_{13} &= \sin \theta_{x'} \sin \theta_{z'} + \cos \theta_{x'} \sin \theta_{y'} \cos \theta_{z'} \\
C_{21} &= \cos \theta_{y'} \sin \theta_{z'} \\
C_{22} &= \sin \theta_{x'} \sin \theta_{y'} \sin \theta_{z'} + \cos \theta_{x'} \cos \theta_{z'} \\
C_{23} &= \cos \theta_{x'} \sin \theta_{y'} \sin \theta_{z'} - \sin \theta_{x'} \cos \theta_{z'} \\
C_{31} &= -\sin \theta_{y'} \\
C_{32} &= \sin \theta_{x'} \cos \theta_{y'} \\
C_{33} &= \cos \theta_{x'} \cos \theta_{y'}
\end{aligned} \tag{3.41}$$

We shall represent the components of  $\vec{R}_0$  as:

$$\vec{R}_0 = \begin{Bmatrix} Rx_0 \\ Ry_0 \\ Rz_0 \end{Bmatrix} \tag{3.42}$$

### 3.3.3.2 Motion Constraints for Given Configurations

Now that we have geometric models of the peg and hole and the transformations necessary to relate them, we may begin the task of characterizing the kinematic constraints imposed by the part geometries. In particular we shall want to specify the position of the hole relative to the peg in terms of the smallest set of independent parameters allowable. The previous coordinate transformation contained the maximum set of 6 unknown position parameters  $(\theta'_x, \theta'_y, \theta'_z, Rx_0, Ry_0, Rz_0)$ . For a configuration with three degrees of freedom, for example, only three of these parameters may be specified independently, while the other three will be functions of these. Our task therefore is to generate these relationships for each of our cases.

We begin with case one described earlier in this chapter. Figure 3.14 shows this case with all three contact points specified in both coordinate systems. We note that the vector pair  $\vec{R}_1$  and  $\vec{R}'_1$  both describe the same point in their respective coordinate frames. This is also true for the other two pairs:  $\vec{R}_2$  and  $\vec{R}'_2$ ,  $\vec{R}_3$  and  $\vec{R}'_3$ . We can use these vector pairs to determine the necessary

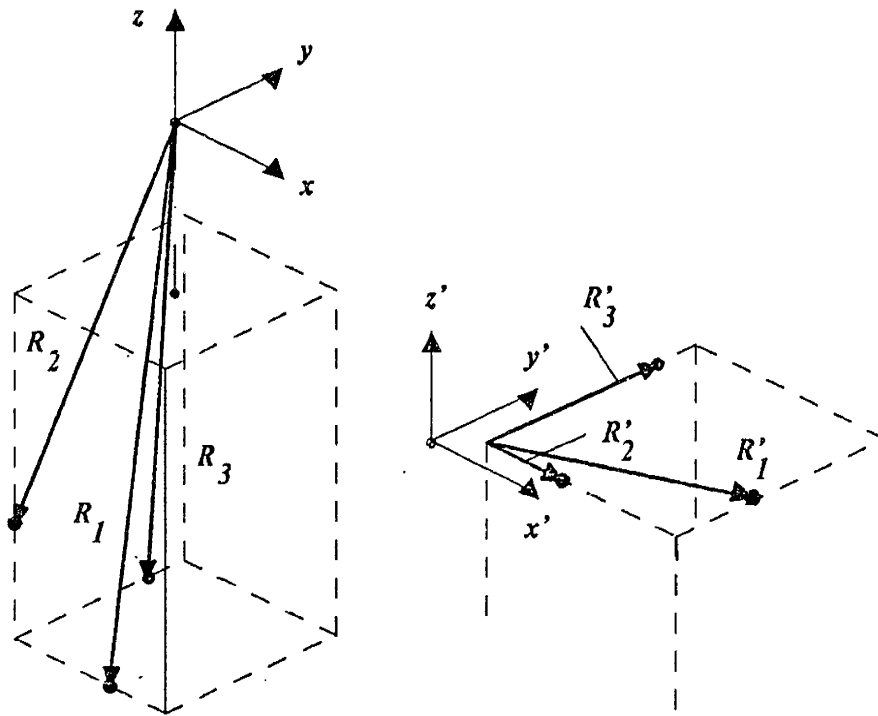


Figure 3.14: Kinematic Constraints: Case One

constraints since point one must be the same point in both coordinate frames as must also be true for points two and three. We therefore have the *three* constraints we were looking for.

We note that point one is constrained by definition to lie on the bottom edge of the peg as shown in Figure 3.14. This edge, which is parallel to the  $x$ -axis of the peg, can also be represented as the intersection of the bottom and side faces of the peg as shown. Since point one in hole coordinates, represented by the tip of vector  $\vec{R}_1'$ , must also be subject to this same constraint, we can write the following relation:

$$\begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} \cdot \left[ \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{Bmatrix} l' \\ a' \\ 0 \end{Bmatrix} + \begin{Bmatrix} Rx_0 \\ Ry_0 \\ Rz_0 \end{Bmatrix} \right] = -\frac{w}{2} \quad (3.43)$$

which translates into the scalar equation:

$$l'C_{12} + a'C_{22} + Ry_0 = -\frac{w}{2} \quad (3.44)$$

The above equation states that point one must be located a distance  $-w/2$  from the origin of the peg's coordinate frame along the  $y$ -axis, or to put it another way, the dot product of the unit vector representing the normal of the back face of the peg and the vector  $\vec{R}_1'$  transformed into peg coordinates is equal to the magnitude of the  $y$ -axis component of the vector  $\vec{R}_1'$ .<sup>15</sup>

Similarly, using a more compact vector notation, we can write:

$$\vec{U}_z \cdot [[C] \vec{R}_1' + \vec{R}_0] = -L \quad (3.45)$$

which produces the scalar equation:

$$l'C_{31} + a'C_{32} + Rz_0 = -L \quad (3.46)$$

The term  $\vec{U}_z$  is a unit vector pointing in the peg's  $+z$  direction and is of the form:

$$\vec{U}_z = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \quad (3.47)$$

---

<sup>15</sup>Refer to Figure 3.9 for  $l', a'$ , etc..

For points two and three we can similarly write the expressions:

$$\vec{U}_x \cdot [[C] \vec{R}_2' + \vec{R}_0] = -\frac{l}{2}b'C_{11} + Rx_0 = -\frac{l}{2} \quad (3.48)$$

$$\vec{U}_y \cdot [[C] \vec{R}_2' + \vec{R}_0] = -\frac{w}{2}b'C_{21} + Ry_0 = -\frac{w}{2} \quad (3.49)$$

$$\vec{U}_x \cdot [[C] \vec{R}_3' + \vec{R}_0] = -\frac{l}{2}(w' - c')C_{12} + Rx_0 = -\frac{l}{2} \quad (3.50)$$

$$\vec{U}_z \cdot [[C] \vec{R}_3' + \vec{R}_0] = -L(w' - c')C_{32} + Rz_0 = -L \quad (3.51)$$

These six equations represent the kinematic constraints between the peg and the hole for case one. From them we can choose any three independent parameters to describe the position of the hole relative to the peg in case one. For simplicity we shall choose the set of roll-pitch-yaw angles:  $(\theta_{x'}, \theta_{y'}, \theta_{z'})$ . We can now solve for all of the position variables in terms of the roll-pitch-yaw angles and the constant parameters of the peg and the hole. We do this by representing the above equations in matrix form, inverting the resulting matrix, and multiplying both sides by the result. Thus we can write:

$$\begin{bmatrix} C_{22} & 0 & 0 & 0 & 1 & 0 \\ C_{32} & 0 & 0 & 0 & 0 & 1 \\ 0 & C_{11} & 0 & 1 & 0 & 0 \\ 0 & C_{21} & 0 & 0 & 1 & 0 \\ 0 & 0 & C_{12} & 1 & 0 & 0 \\ 0 & 0 & C_{32} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a' \\ b' \\ (w' - c') \\ Rx_0 \\ Ry_0 \\ Rz_0 \end{bmatrix} = \begin{bmatrix} -\frac{w}{2} - l'C_{21} \\ -L - l'C_{31} \\ -\frac{l}{2} \\ -\frac{w}{2} \\ -\frac{l}{2} \\ -L \end{bmatrix} \quad (3.52)$$

To save space, the resulting expressions for  $(a', b', w', Rx_0, Ry_0, Rz_0)$ , are listed in Appendix A.

The other contact parameters, such as:  $a$ ,  $b$ , and  $c$ , can be solved by plugging the above expressions back into the transformation:

$$\vec{R}_m = [C] \vec{R}_m' + \vec{R}_0 \quad (3.53)$$

and solving. We should mention that for the other cases where there are fewer than three degrees of freedom, the above matrix will not be square due to the fact that the three angles  $(\theta_{x'}, \theta_{y'}, \theta_{z'})$  will no longer all be independent. The extra equations (rows of the matrix) are redundant and will represent the kinematic constraints between these angles. These extra rows will be removed from the matrix and solved separately, leaving a square matrix to be inverted for the linear parameters.

### 3.3.4 Representation of Sliding Constraints

Now that we have represented the kinematic constraints of the peg and hole, we will want to represent these constraints in terms of the forces they impart to the system during assembly. To do this we shall determine the contact normals and contact point unit velocities and use these to model the normal and tangential contact forces.<sup>16</sup>

#### 3.3.4.1 Determining Contact Normals

Figure 3.15 shows two edges in contact. If we assume these to be frictionless edges then we can see that the only forces that the edges can exert on each other in static equilibrium lie along the common normal between them. Any other forces would cause the edges to accelerate relative to each other. In the presence of friction, the two edges can exert reaction forces within a friction cone centered about this normal.<sup>17</sup> If we represent these two edges by the unit vectors  $\vec{i}$  and  $\vec{j}$  as shown then the sense of the common normal is given by the cross product:

$$\vec{N}_{normal} = \vec{i} \times \vec{j} \quad (3.54)$$

We note that the sense of the normal  $\vec{N}_{normal}$  is given by the order in which the vectors are crossed and its magnitude is determined by the cross product relation. For the case of the peg and the hole, we shall assume that all normals

---

<sup>16</sup>See Section 3.2.4

<sup>17</sup>See Section 3.2.1.4

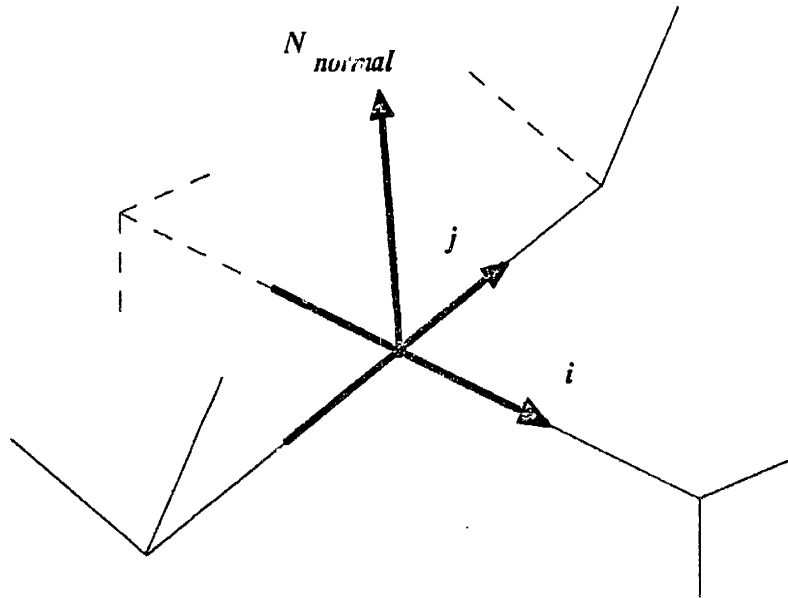


Figure 3.15: Contact Normal for Edge-Edge Contact

point into the peg, i.e. an edge of the hole can only *push* on an edge of the peg, and the magnitude of the normal shall be 1, i.e. a unit vector, since we care about the *sense* of the vector only. Therefore, to determine a normal vector of unit magnitude we write:

$$\vec{n}_{normal} = \frac{\vec{i} \times \vec{j}}{|\vec{i} \times \vec{j}|} \quad (3.55)$$

where the  $z$  component of the normal is always taken to be positive (pointing into the peg). For a vertex-face contact, the contact normal will simply be the normal of the face of the hole that the peg is touching, transformed into peg coordinates.

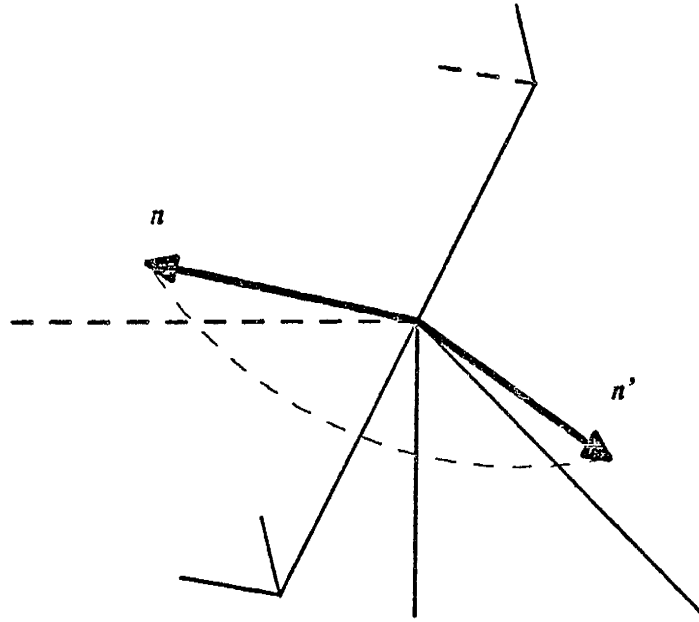


Figure 3.16: Contact Normal for an Edge-Corner Contact

For case one, the three edge-edge contact normals will be:

$$\begin{aligned}\vec{n}_1 &= \frac{\vec{U}_x \times \vec{U}'_y}{|\vec{U}_x \times \vec{U}'_y|} \\ \vec{n}_2 &= \frac{\vec{U}_z \times \vec{U}'_x}{|\vec{U}_z \times \vec{U}'_x|} \\ \vec{n}_3 &= \frac{\vec{U}'_y \times \vec{U}_y}{|\vec{U}'_y \times \vec{U}_y|}\end{aligned}\tag{3.56}$$

where the primed unit vectors of the hole would first have been transformed into the peg's coordinate frame. The contact normals for case 1 in terms of model parameters are contained in Appendix A.

For the case of an edge of the peg in contact with a corner of the hole, the definition of the contact normal becomes somewhat more complicated. As we showed in Section 3.2.1.4 we can think of the corner of the hole as consisting of two edges of the hole which are simultaneously in contact with an edge of the peg. Therefore, each of the two edges of the hole defines a different normal when crossed with the edge of the peg. If we consider the edge-corner contact



to be frictionless, then the reaction forces that can be exerted between the two bodies could lie anywhere in a  $90^\circ$  arc between these two normals in the plane they define, as shown in Figure 3.16. Here the contact normal is defined to be the quarter plane bounded by the vectors  $\vec{n}$  and  $\vec{n}'$ . If we consider a three-dimensional symmetric friction cone to be centered around the normal  $\vec{n}$  and then *sweep* the normal and the friction cone until they are aligned with the normal  $\vec{n}'$ , the volume swept out represents the set of possible reaction forces in the presence of friction. The corner friction cone mentioned in Section 3.5 was derived in this fashion.

### 3.3.4.2 Determining the Contact Velocities

As we mentioned in Section 3.2.4, the sense of the frictional reaction force will be determined by the sense of the velocity (or impending velocity) of the contact point. For the case of the peg moving around the hole, the contact velocities will be given by the velocities of the points of contact relative to the peg coordinates. Given a contact point  $m$ , specified in both coordinate frames and related by the coordinate transformation:

$$\vec{R}_m = [C] \vec{R}_m' + \vec{R}_0 \quad (3.57)$$

we can differentiate the relationship with respect to time and obtain:

$$\frac{d}{dt}(\vec{R}_m) = \frac{d}{dt}([C]) \vec{R}_m' + [C] \frac{d}{dt}(\vec{R}_m') + \frac{d}{dt}(\vec{R}_0) \quad (3.58)$$

To determine the velocities of the contacts relative to the peg coordinates, we pick a contact point which is fixed relative to the coordinates of the hole. The velocities we shall then be concerned with are the velocities of these fixed points relative to the peg's coordinate frame. Since these points are fixed relative to the coordinates of the hole, the term  $\frac{d}{dt}(\vec{R}_m')$  will be zero, therefore the velocity of contact  $m$ , relative to the peg, is given by:

$$\frac{d}{dt}(\vec{R}_m) = \frac{d}{dt}([C]) \vec{R}_m' + \frac{d}{dt}(\vec{R}_0) \quad (3.59)$$

We note that since the velocities given by the above equation are of contact points fixed to the hole and moving relative to the peg, the frictional forces

imposed on the peg will point in the *same* direction as these velocities. As was the case with the contact normals, we are concerned only with the sense of these velocities and not with their magnitude. To turn the above velocities into the *unit* velocities we want, we shall divide each of them by their respective magnitudes. Therefore the unit contact velocity at point  $m$  will be given by:

$$\vec{v}_m = \frac{\frac{d}{dt}(\vec{R}_m)}{\left|\frac{d}{dt}(\vec{R}_m)\right|} \quad (3.60)$$

In the edge-corner case we note that in order for the edge of the peg to remain in contact with the corner of the hole, no component of the unit contact velocity for that point may be parallel to the top edges of the hole that define the corner. Therefore, the unit contact velocity for the edge-corner case must be parallel to the edge of the peg that defines the contact.

### 3.3.5 Representing the Assembly State

We have now determined the kinematic constraints of our system and the unit vectors that we can use to represent the normal and tangential contact forces in our force balance equations. Before we proceed to develop these equations we shall first look at how these relationships can best be represented in terms of developing an assembly strategy. Here we shall use the term *assembly state* to define the set of position and unit velocity parameters of the peg in a given configuration.

#### 3.3.5.1 Nominal Position in Terms of Roll-Pitch-Yaw Coordinates

In Section 3.3.3.2 we chose to represent the position of case one in terms of the three roll-pitch-yaw angles  $(\theta'_x, \theta'_y, \theta'_z)$ . We could also have chosen an independent set of the edge contact parameters such as  $(a', b', c')$ . In fact such a set might be more useful in specifying a desired configuration. For example, in the transition between case one and case two we are concerned with establishing an edge-corner contact at point three. In view of this it would make more sense to use the edge contact parameters to specify position since we then could easily set

up an initial condition that would guarantee point three would reach the corner before any other contacts, i.e.  $c' \approx 0$ . However, solving for the roll-pitch-yaw angles explicitly in terms of these variables would have been extremely difficult since it would have involved inverting large trigonometric expressions and choosing among multiple solutions. In addition, the resulting expressions would have proven too unwieldy to incorporate into later equations. In the interests of computational simplicity then, we shall resign ourselves to using the roll-pitch-yaw angles for now as our independent position parameters.<sup>18</sup>

### 3.3.5.2 A Convenient Representation of Contact Velocities

In addition to the position parameters for the peg and hole system, we must also specify parameters for the unit velocity of each contact point. Again, in light of our concern for establishing a desired configuration, we might wish to specify our unit velocity in terms of the velocities of contact points along the edges. Luckily, in the case of velocity, we are able to do this. Consider again equation 3.59:

$$\frac{d}{dt}(\vec{R}_m) = \frac{d}{dt}([C]) \vec{R}_m' + \frac{d}{dt}(\vec{R}_0)$$

The derivative of the rotational transformation matrix can be represented in the form:

$$\frac{d}{dt}([C]) = [W'] [C] \quad (3.61)$$

where

$$[W'] = \begin{bmatrix} 0 & w'_z & -w'_y \\ -w'_z & 0 & w'_x \\ w'_y & -w'_x & 0 \end{bmatrix} \quad (3.62)$$

The matrix  $[W']$  represents the velocity of the hole coordinates relative to the peg coordinates in terms of an angular velocity  $w'$  about the *instantaneous axis of rotation* in the fixed frame. We notice that the elements of  $[W']$  are linear and would therefore be easy to invert. Since we are able to easily specify our

---

<sup>18</sup>Note that the number of independent roll-pitch-yaw angles will depend on the degrees of freedom of the given case, see Section 3.3.3.2.

velocity in terms of any set of parameters we desire, we shall do so as outlined below.

From the kinematic constraints of the previous section we obtained expressions for the linear contact parameters in terms of angles. By differentiating these expressions with respect to time, we obtain a set of equations relating the linear edge contact velocities to the angular velocities of the hole coordinate frame. In case one for example, if we choose our linear parameters to be  $(a', b, c')$ , then we can write:

$$\begin{aligned}\dot{a}' &= \mathcal{F}_1(\dot{\theta}'_x, \dot{\theta}'_y, \dot{\theta}'_z) \\ \dot{b} &= \mathcal{F}_2(\dot{\theta}'_x, \dot{\theta}'_y, \dot{\theta}'_z) \\ \dot{c}' &= \mathcal{F}_3(\dot{\theta}'_x, \dot{\theta}'_y, \dot{\theta}'_z)\end{aligned}$$

where  $\mathcal{F}_m()$  represent *linear* functions. Therefore we can express the above equations in matrix form and write:

$$\begin{Bmatrix} \dot{a}' \\ \dot{b} \\ \dot{c}' \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{Bmatrix} \dot{\theta}'_x \\ \dot{\theta}'_y \\ \dot{\theta}'_z \end{Bmatrix} \quad (3.63)$$

where the elements of  $[A]$  represent the coefficients of the angular velocity terms. We can now easily solve for the angular velocity components in terms of the linear contact velocities, by inverting  $[A]$  and multiplying for example. Now we have a set of parameters, three for case one, representing the velocity of the hole relative to the peg. As we stated in Section 3.2.3, we are concerned only with a velocity of *unit* magnitude. For simplicity we shall normalize the contact velocity parameters.<sup>19</sup> In case one for example, if we choose  $\dot{c}' = -1$ , then  $\dot{a}'$  and  $\dot{b}$  can be expressed in terms of  $\dot{c}'$ , thereby reducing the number of velocity parameters by one.

What we now have are expressions for the position and unit velocity parameters of the peg and hole system, in terms of the independent roll-pitch-yaw angles and a set of normalized contact velocities. This set of independent parameters, or *state variables*, are what will have to be specified in the assembly planning process. In Section 3.5.1 we shall develop procedures to incorporate

<sup>19</sup>This corresponds to dropping the magnitude parameter of a vector.

the *a priori* specification of these state variables into the strategy development process by setting limits on their allowable values.

### 3.3.6 Quasi-static Equilibrium Equations

The final step in our modeling process is to formulate the quasi-static force and moment balance equations for our system. In the previous sections we determined the unit normal and velocity vectors which will define the senses of the corresponding normal and tangential reaction forces. We shall now incorporate these expressions into the equations that govern the applied forces and moments of our system.

In their simplest form, we can represent the requirements for quasi-static equilibrium by:

$$\sum \vec{F} = \vec{F}_{ext} + \sum_{i=1}^n \vec{f}_i = 0 \quad (3.64)$$

and

$$\sum \vec{M} = \vec{M}_{ext} + \sum_{i=1}^n (\vec{R}_i \times \vec{f}_i) = 0 \quad (3.65)$$

where  $\vec{F}_{ext}$  and  $\vec{M}_{ext}$  are  $3 \times 1$  column vectors representing the applied forces and moments respectively,  $\vec{R}_i$  is the position vector of contact  $i$ , and  $\vec{f}_i$  is the reaction force (normal and tangential combined) at contact  $i$ . All vectors are expressed in the peg coordinates.

To constrain the reaction forces to lie on the edge of their respective friction cones, i.e. impending sliding motion, we can write:

$$\vec{f}_i = \vec{f}_i^{normal} + \vec{f}_i^{tangent}$$

where

$$|\vec{f}_i^{tangent}| = \mu |\vec{f}_i^{normal}|$$

In the second chapter we introduced the variable  $\delta$  to represent an infinitesimal departure from equilibrium.<sup>20</sup> In particular, we stated that by adding  $\delta$  to

---

<sup>20</sup>See Section 2.4.4

the frictional component of each reaction force and solving the equilibrium equations for  $\delta \geq 0$ , we could specify which side of the constraint curves represented jamming (reaction forces within the friction cone), and which side represented sliding. We shall adopt the same procedure for the three-dimensional equilibrium equations. Therefore, we can write:

$$\vec{f}_i^{tangent} \longrightarrow \bar{f}_i^{tangent} + \vec{\delta}_i \quad (3.66)$$

where  $\vec{\delta}_i \ll 1$  and has the same sense as  $\bar{f}_i^{tangent}$ .

In terms of the unit normal and velocity vectors developed earlier, we can rewrite our reaction forces as:

$$\vec{f}_i^{normal} = f_i \vec{n}_i \quad (3.67)$$

$$\vec{f}_i^{tangent} = \mu f_i \vec{v}_i \quad (3.68)$$

$$\vec{\delta}_i = \delta \vec{v}_i \quad (3.69)$$

where  $f_i$ ,  $\mu$ , and  $\delta$  are scalar quantities. For the special case of the edge-corner contact of Section 3.3.4.1, where the normal force will lie somewhere within a  $90^\circ$  arc between two bounding normals, we shall represent the normal force as:

$$\vec{f}_i^{normal} = f_i (\vec{n}_i \sin \phi_i + \vec{n}'_i \cos \phi_i) \quad (3.70)$$

where the actual sense of the normal vector is defined by the parameter  $\phi_i$ . Determining the value of  $\phi_i$  shall be done in Section 3.5.2.

We can now rewrite our equilibrium equations as:

$$\sum \vec{F} = \vec{F}_{ext} + \sum_{i=1}^n (f_i \vec{n}_i + \mu f_i \vec{v}_i + \delta \vec{v}_i) = 0 \quad (3.71)$$

and

$$\sum \vec{M} = \vec{M}_{ext} + \sum_{i=1}^n ((\vec{R}_i \times f_i \vec{n}_i) + (\vec{R}_i \times \mu f_i \vec{v}_i) + (\vec{R}_i \times \delta \vec{v}_i)) = 0 \quad (3.72)$$

The above equations will be used to define the boundaries of our applied force-moment solution regions. All of the factors that we have identified as significantly affecting the assembly process are embodied in these two equations.

Factors such as friction, geometry, tolerances, unit velocity, and position, are all represented as constraints on the applied forces and moments that will be used to control the assembly process. Our next task will be to represent and analyze these constraints in as manageable and straightforward a way as possible.

### 3.4 The Applied Force-Moment Solution Space

Recall from the second chapter the solution space of  $(\frac{M}{F}$  vs  $\alpha$ ) for the planar peg and hole. In it we represented the limits on the applied force and moment that would result in sliding, jamming, or breaking contact. In this chapter we shall follow a similar approach for the three-dimensional rectangular peg and hole. Our aim will be to develop a visual interpretation of the effects of the various factors that control assembly, and to develop graphical techniques that will aid in the search for forces and moments that will guarantee successful assembly. In this manner the process of interrelating the constraints from all the various factors in assembly will be reduced to a process of searching for solution regions within a well defined geometric domain.

#### 3.4.1 The 5 Dimensional Force-Moment Space

In the two-dimensional case the set of applied forces and moments took the form  $(F_x, F_y, M_z)$ . In addition, since the system model was quasi-static, the magnitudes of the applied forces and moment could be normalized. In other words, the applied force was transformed from cartesian to polar coordinates and the resulting equations divided by its magnitude  $F$ , so that the resulting terms were of the form  $(\frac{M}{F}, \alpha)$ . We were thereby able to reduce the order of the solution space from 3 to 2.<sup>21</sup>

In the three dimensional case we have applied forces and moments of the form  $(F_x, F_y, F_z, M_x, M_y, M_z)$ . Since the three-dimensional system is also governed by quasi-static equations, we can similarly normalize the above terms. Figure 3.17 shows the applied force represented in spherical coordinates. To represent the

---

<sup>21</sup>See Section 2.4.5.

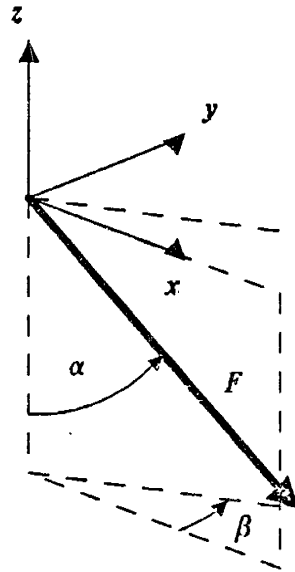


Figure 3.17: Applied Force in Spherical Coordinates

force in the coordinates of the peg we write:

$$\begin{aligned}
 F_x &= F \sin \alpha \cos \beta \\
 F_y &= F \sin \alpha \sin \beta \\
 F_z &= -F \cos \alpha
 \end{aligned}
 \tag{3.73}$$

where  $\alpha$  and  $\beta$  are defined as shown. By dividing the equations through by the term  $F$ , the resulting terms are  $(\frac{M_x}{F}, \frac{M_y}{F}, \frac{M_z}{F}, \alpha, \beta)$ . We therefore have a 5 dimensional force-moment solution space for the three-dimensional system.

### 3.4.2 Making the Solution Space More Tractable

As we stated earlier, our aim is to develop representations of the constraints on our applied forces and moments that will help us visualize and hence better understand the relationship between the various factors that affect assembly. In the case of the two-dimensional peg and hole, this visualization took the form of the  $(\frac{M}{F} \text{ vs. } \alpha)$  solution plane. In the three-dimensional case, the 5 dimensional solution space does not seem to fulfill this goal directly since it is in general very difficult to plot, much less conceptualize, regions that are defined in 5



dimensions. We therefore must find ways in which to represent the solution regions in lower dimensional, and hence more manageable, domains.

### 3.4.2.1 Subdividing the 5D Solution Space

In order to be able to visualize the solution regions defined by the quasi-static equilibrium equations, they must be represented into domains whose dimensions are less than or equal to three. If we re-examine the equilibrium equations of Section 3.3.6 in terms of their scalar components, we see that there are six scalar equations of the form:

$$\begin{aligned}
 \sum F_x &= 0 \\
 \sum F_y &= 0 \\
 \sum F_z &= 0 \\
 \sum M_x &= 0 \\
 \sum M_y &= 0 \\
 \sum M_z &= 0
 \end{aligned}
 \tag{3.74}$$

Within these equations are four sets of variables. Namely, the set of applied forces and moments in their new form,  $(\frac{M_x}{F}, \frac{M_y}{F}, \frac{M_z}{F}, \alpha, \beta)$ , the set of normal reaction forces,  $(f_i \text{ for } i = 1, 2, 3, \dots)$ , and  $(\phi_i)$  where applicable, the set of state variables,  $(\theta_{x'}, \theta_{y'}, \theta_{z'}, \dot{a}', \dot{b}, \dots)$ , and the set of constant system parameters,  $(l, w, l', w', L, \mu)$ .<sup>22</sup> Of these four sets, two, namely the state variables and the system parameters, must be predetermined before the solution regions can be defined. The remaining two sets of variables, normal reaction forces and applied forces and moments, will be determined by the equilibrium equations. Of these last two sets, a given number of variables in each will be independent. For example, in case one there are three points of edge-edge contact. Therefore there are three normal reaction forces,  $f_1, f_2,$  and  $f_3$ . If we solve the force balance equations for  $(f_1, f_2, f_3)$  in terms of functions of  $\alpha$  and  $\beta$  and substitute these into the three moment balance equations, we will end up with three equations

---

<sup>22</sup>The one other variable that appears is  $\delta$  which is defined as being infinitesimal and therefore not considered in the above sets.

of the form:

$$\begin{aligned}\frac{M_x}{F} &= \mathcal{G}_1(\alpha, \beta) \\ \frac{M_y}{F} &= \mathcal{G}_2(\alpha, \beta) \\ \frac{M_z}{F} &= \mathcal{G}_3(\alpha, \beta)\end{aligned}\tag{3.75}$$

where  $\mathcal{G}_m()$  represent not necessarily linear functions. From this we see that, for case one, the solution regions are determined by three *independent* functions of  $\alpha$  and  $\beta$ . We can therefore represent our solution regions in three separate solution *subspaces* of the form:

$$\begin{aligned}\frac{M_x}{F} \text{ vs } (\alpha, \beta) \\ \frac{M_y}{F} \text{ vs } (\alpha, \beta) \\ \frac{M_z}{F} \text{ vs } (\alpha, \beta)\end{aligned}$$

Figure 3.18 shows constraint surfaces represented in these three subspaces that were generated for a particular state of case one. We note that the two horizontal axes of each subspace represent the same  $(\alpha, \beta)$  plane. In other words, the three (three-dimensional) subspaces have two of their three dimensions in common with each other. The third dimension in each, namely  $(\frac{M_x}{F}, \frac{M_y}{F}, \frac{M_z}{F})$  are independent of the other two. When the equilibrium equations are solved as inequalities for  $\delta \geq 0$ , the surfaces represent sliding constraints. Namely, one side of each surface will represent applied forces and moments that will jam the assembly, while the other side represents those that result in sliding.

To determine the breaking contact constraints on the applied forces and moments, we take the expressions for  $f_i$  in terms of  $\alpha$  and  $\beta$  and solve them for  $f_i \geq 0$ , assuming  $\delta = 0$ . In the example of case one, the breaking contact constraints take the form of curves in the  $(\alpha, \beta)$  plane. These curves, shown in Figures 3.19 and 3.20, represent limits on the direction of the applied force  $F$  that will break contacts. If we superimpose the curves, as shown, then a region is defined in which all of the reaction forces are guaranteed to be positive, i.e. pushing on the peg. If we extend these constraints into our three subspaces, then the portions of the sliding constraint regions that lie within these breaking contact bounds represent valid solution regions for the case being considered. Sliding regions that lie outside these bounds are not valid since they assume negative reaction forces which are not physically realizable in our system.

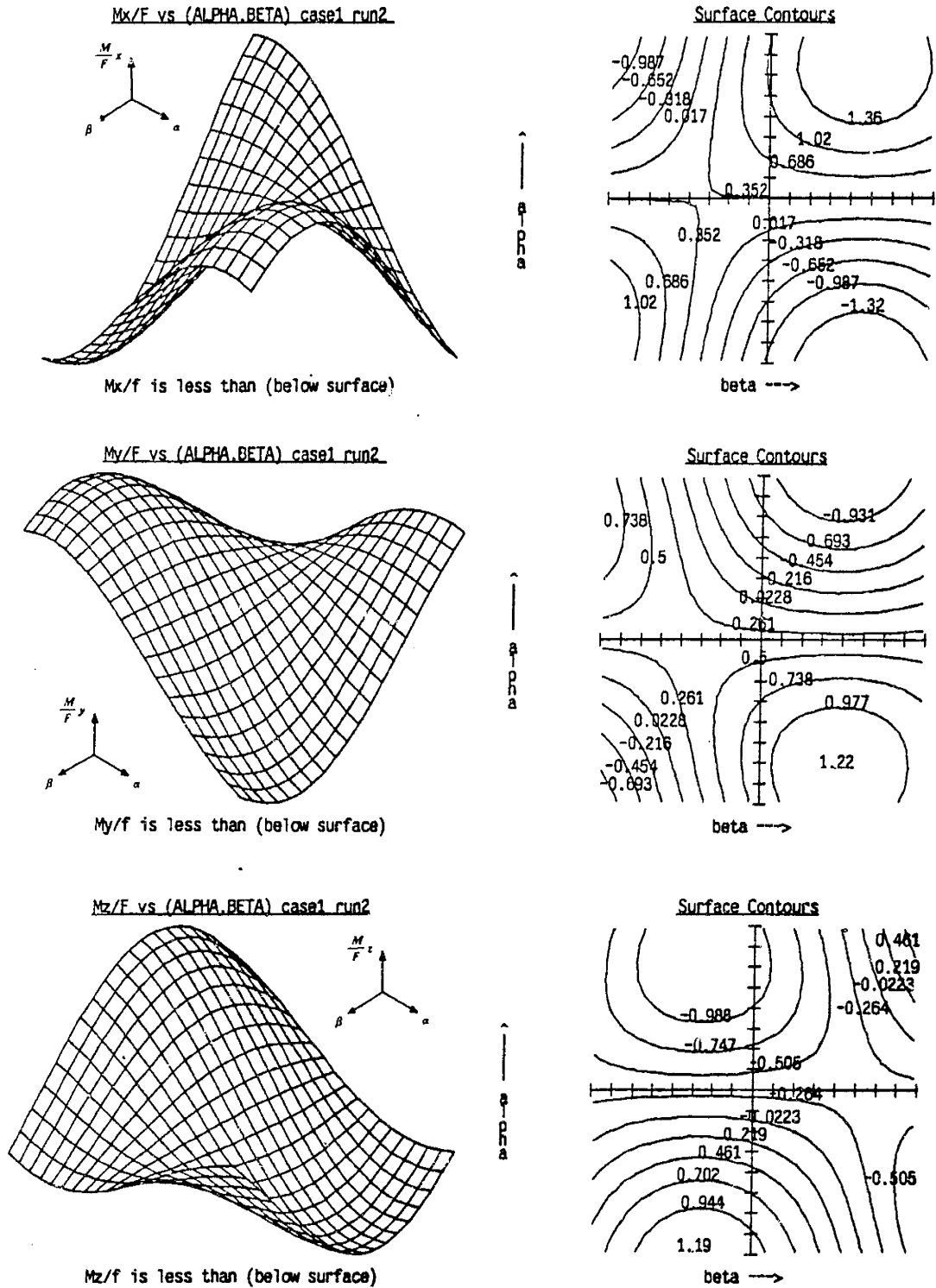


Figure 3.18: Sliding Constraints in 3 Three-Dimensional Subspaces

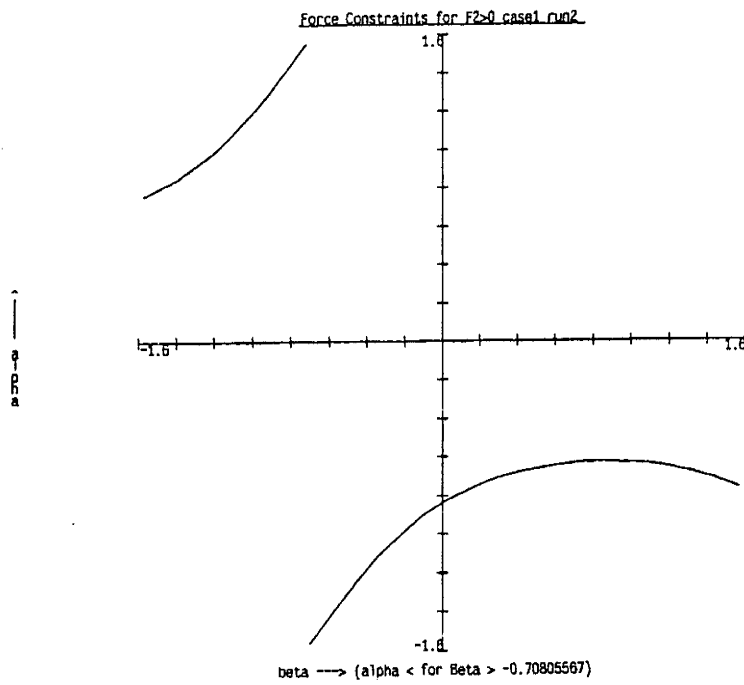
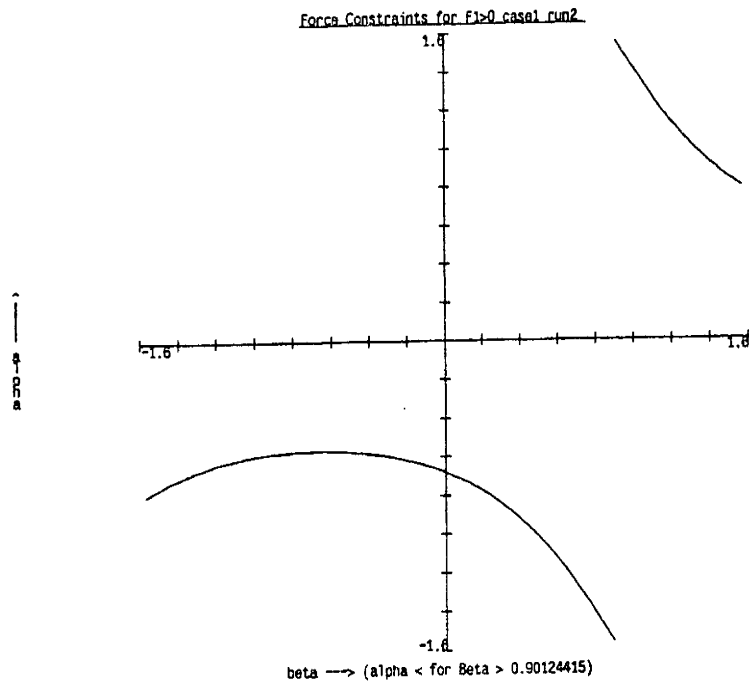


Figure 3.19: Breaking Contact Constraint Curves

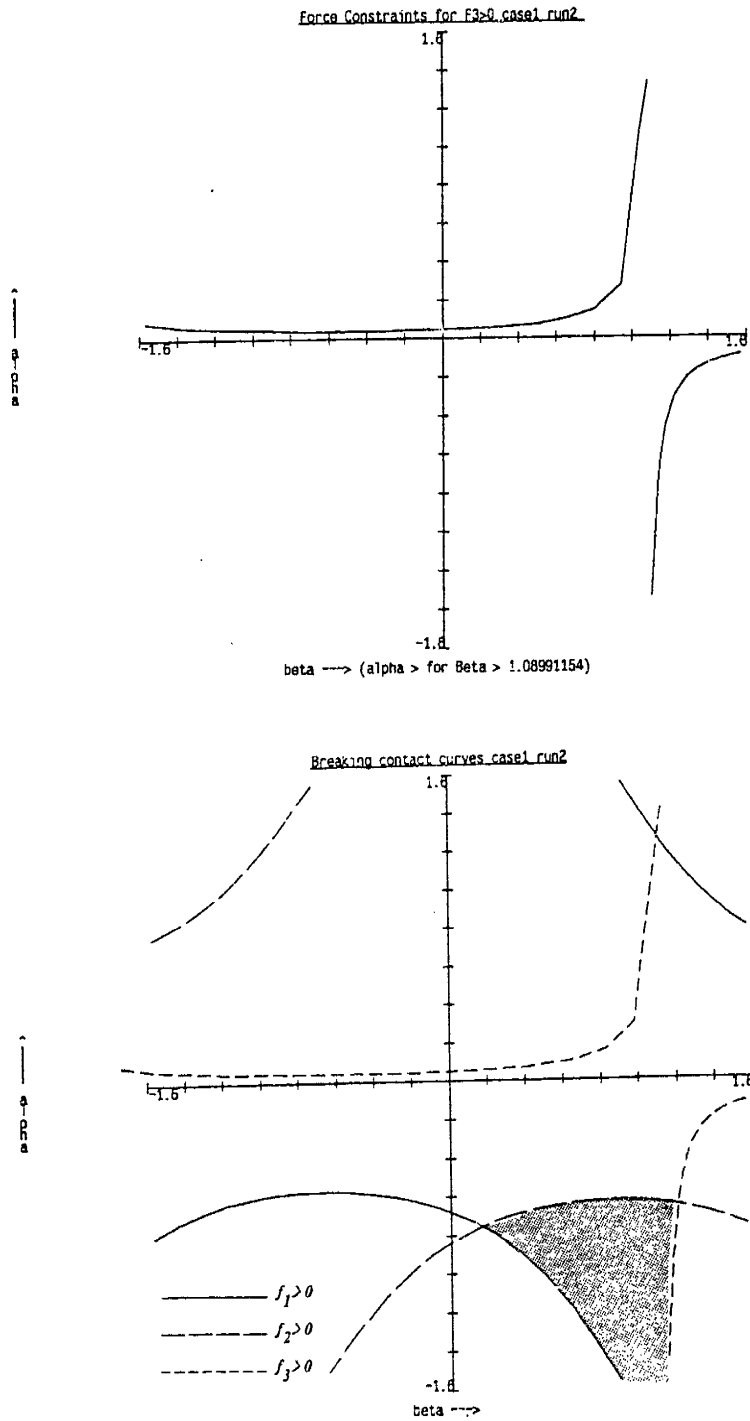


Figure 3.20: Breaking Contact Constraint Curves, (Cont.)

This particular representation of the solution space as a collection of three-dimensional subspaces makes it considerably easier to visualize the force and moment constraints and will therefore prove to be particularly useful in the strategy development process. The one catch to this representation lies in the fact that for contact cases with fewer than three degrees of freedom, the  $(\frac{M_x}{F}, \frac{M_y}{F}, \frac{M_z}{F})$  dimensions of the solution space will no longer be independent of each other. To deal with this problem, we will first examine the relationship between the degrees of freedom of a system and the order of the resulting force-moment constraint regions.

### 3.4.2.2 Number of Contacts and the Resulting Constraint Curves

We recall that the degrees of freedom a system has is determined by the number of independent parameters necessary to specify its position uniquely. For a three-dimensional rigid body a maximum of six degrees of freedom is possible, namely three position parameters and three orientation parameters. As we add constraints we introduce dependency relationships between some of these parameters. Just as we defined degrees of freedom, we can similarly define degrees of *constraint*, namely the number of position or orientation parameters that are no longer independent. We can therefore write the following relationship for a three-dimensional rigid body.

$$d.o.f + d.o.c = 6$$

This expression illustrates the complementary relationship between the degrees of freedom and degrees of constraint of a system.

Consider again the six equilibrium equations of Section 3.4.2.1. For the example of case one, the peg and hole system had three contact points each representing one degree of constraint and therefore had three  $(6 - 3) = 3$  degrees of freedom remaining. By solving for and eliminating the three reaction forces, we were able to reduce the six equations to three, each one defining a surface that varied in three dimensions (see Figure 3.18). In other words, each surface represents a constraint simultaneously relating three of the five force-moment parameters (the *three* surfaces together constrain all five parameters).

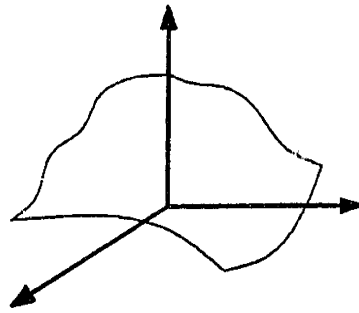
In addition, we were able to derive and solve equilibrium equations, in terms of the normal reaction forces, that guaranteed contacts would not be broken. These constraints took the form of three surfaces that varied in two dimensions of our three dimensional subspaces. Each surface simultaneously represents the constraints on two of the five force parameters  $(\alpha, \beta)$ . For any one given surface, three of the five force-moment parameters are independent and hence not constrained by that surface. We note that none of the breaking contact constraint curves determine any constraints on the three  $\frac{M_i}{F}$  dimensions of our applied force-moment space. This is represented by the breaking contact curves in the  $(\alpha, \beta)$  plane being *swept* into each of the  $\frac{M_i}{F}$  dimensions. This representation serves to illustrate why we chose to decouple our five-dimensional force-moment space into the 3 three-dimensional subspaces.

Let us for the moment imagine the general *six*-dimensional applied force-moment space, for example consisting of the dimensions  $(F_x, F_y, F_z, M_x, M_y, M_z)$ .

If we now imagine a contact case of the peg and hole containing only two contact points, therefore possessing four degrees of freedom, then by eliminating the two reaction force terms we will be left with four equations. Each of these equations will relate (constrain) three of the six force-moment parameters, thereby defining a surface that varies in three of the six dimensions. We can also solve the two equations for the reaction forces greater than zero which will define two surfaces that each vary in two dimensions of our three-dimensional subspaces.

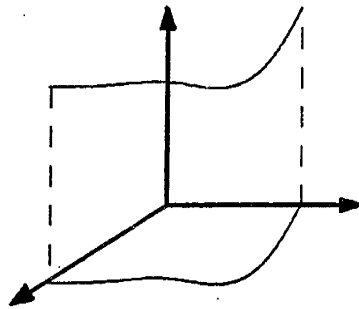
For the case of a system with only one contact point (five degrees of freedom) the result of eliminating from the six equilibrium equations the single normal reaction force term will be five equations, each relating two force-moment parameters. Each equation will define a surface that varies in two dimensions. The single breaking contact constraint equation will represent a plane fixed by a point on (and perpendicular to) one of the axes in each of the subspaces.

Figure 3.21 illustrates the meaning of the term '*a surface varying in n-dimensions*'. We can summarize the relationship between the degrees of freedom a system has and the dimensionality of its constraint boundaries in the general six-dimensional force-moment space as follows:



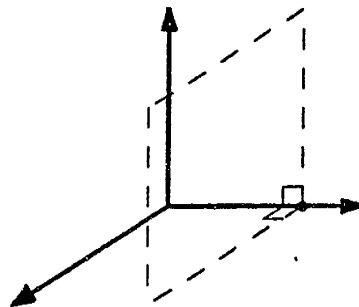
*Surface Varying  
in Three Dimensions*

*a general three  
dimensional surface*



*Surface Varying  
in Two Dimensions*

*a two dimensional curve  
swept into the third  
dimension*



*Plane Perpendicular  
to One Dimension*

*a point swept into  
the other two  
dimensions*

Figure 3.21: Variation of Surfaces in the Six-Dimensional Force-Moment Constraint Space



| Degrees<br>of<br>Freedom | Sliding<br>Constraint<br>Boundaries          | Breaking-Contact<br>Constraint<br>Boundaries    |
|--------------------------|--|---|
| five                     | five surfaces varying in<br>two dimensions   | one (plane) perpendicu-<br>lar to one dimension |
| four                     | four surfaces varying in<br>three dimensions | two surfaces varying in<br>two dimensions       |
| three                    | three surfaces varying in<br>four dimensions | three surfaces varying in<br>three dimensions   |
| two                      | two surfaces varying in<br>five dimensions   | four surfaces varying in<br>four dimensions     |
| one                      | one surface varying in six<br>dimensions     | five surfaces varying in<br>five dimensions     |

The two cases we omitted from the above list are six degrees of freedom and zero degrees of freedom. A system with six degrees of freedom is by definition unconstrained and will therefore have no constraint boundaries. A system with zero degrees of freedom will be fully constrained and therefore be unable to move, i.e. unable to break contacts or slide.

Since our constraint space is a five-dimensional subspace of the general six-dimensional force-moment space, we will have to reduce the dimensions of the corresponding constraint surfaces accordingly.<sup>23</sup> In addition, by choosing to represent the applied force in polar form using the angles  $\alpha$  and  $\beta$ , we have introduced three nonlinear equations in three unknowns.<sup>24</sup> Some of the resulting constraint surfaces for few degrees of freedom, therefore, will not appear to be directly swept into higher dimensions due to the presence of these trigonometric functions.<sup>25</sup>

---

<sup>23</sup>See Section 3.4.1.

<sup>24</sup>See Equation 3.73.

<sup>25</sup>By *directly swept* we mean along a straight line parallel to the axis of the dimension into which

What we can conclude from the above discussion is that contact cases with three or fewer contacts, i.e. three or more degrees of freedom, can be represented in a straightforward way in our three force-moment constraint subspaces. In general, the fewer degrees of freedom that are constrained, the more decoupling of the resulting constraint equations is possible. Therefore, the constraint regions are bounded by lower dimensional surfaces.

As we mentioned briefly in the previous section, the cases that have more constrained degrees of freedom pose a slight problem. Just as more degrees of freedom resulted in lower dimensional constraint boundaries, fewer degrees of freedom will represent *higher* dimensional constraint boundaries. To represent these more constrained cases in our three subspaces, we shall have to select *a priori* some values related to the applied force-moment parameters. This will be equivalent to taking *cross-sections* of the higher dimensional constraint boundaries and projecting them into our lower dimensional subspaces. In Section 3.5.2 we shall illustrate a method for doing this that can be easily incorporated into the strategy development process.

At this point we should pause and review exactly what the constraint surfaces previously derived actually represent. We recall that the equilibrium equations of Section 3.3.6 represent the peg and hole system in a given *state* defined by the state variables representing both the position and unit velocity of the peg. Therefore the resulting solution space surfaces actually represent equilibrium only for that particular state. What we will be assuming is that the regions bounded by these surfaces are well behaved in the neighborhood of this state and therefore may be used to represent a larger portion of the assembly trajectory.<sup>26</sup> The scope of our solutions will therefore depend on the scope and validity of these assumptions, as well as the other assumptions we've made up to now. In Section 5.2.1 of Chapter 5 we shall examine the relationship of the 5 dimensional solution space for a discrete state to higher dimensional spaces that would represent a more general solution for all states.

---

the curve is being extended.

<sup>26</sup>See Section 3.2.1.5.

### 3.5 Selecting a Strategy

We have now completed the task of modeling the rectangular peg and hole system and have developed a means of representing valid solution regions in terms of the imposed constraints. Our task now will be to utilize these tools in the development of a compliant assembly strategy. In effect we shall be pulling together the previous work of this chapter and integrate it into an overall plan of attack for developing such a strategy.

As we mentioned in the beginning of this chapter, we have adopted a design approach to assembly planning. In order to make the problem more tractable, we have been forced to make a number of assumptions and simplifications. Our approach then will combine the analysis techniques just developed and the assumptions made throughout this and previous chapters. As we mentioned earlier, the process of planning an assembly strategy will by necessity be somewhat iterative in nature. It will not be surprising then if we are forced to respecify certain parameters within our models. Our hope is that the tools provided here will make that respecification process more straightforward and therefore reduce the total number of iterations necessary during the planning process.

We can break down the planning process into five basic steps. The first two, namely selecting configurations and then modeling them, have already been completed. The next two steps are to select an assembly *trajectory* consisting of a number of state variable sets with which to discretely approximate the motion of our system, and to select a suitable set, or sets, of applied forces and moments with which to control the assembly process. We shall deal with these two steps next. The last step will be to evaluate the resulting assembly plan to determine if another iteration is necessary. This last step will be dealt with in Section 3.6. The overall planning process is illustrated in Figure 3.22. The implementation of a resulting strategy, here shown as a separate step, comes about after the iterations within the loop have been completed.<sup>27</sup>

---

<sup>27</sup>See Chapter 4.

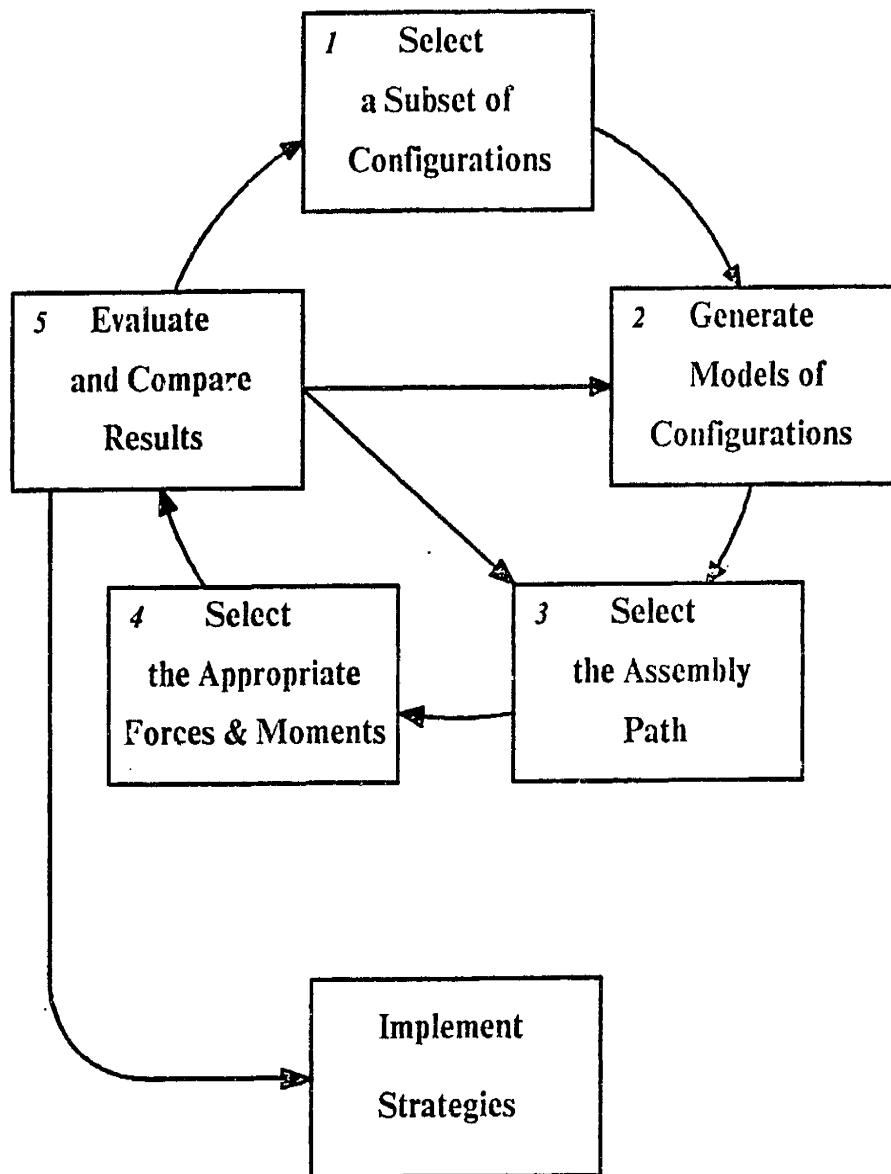


Figure 3.22: Assembly Planning Loop

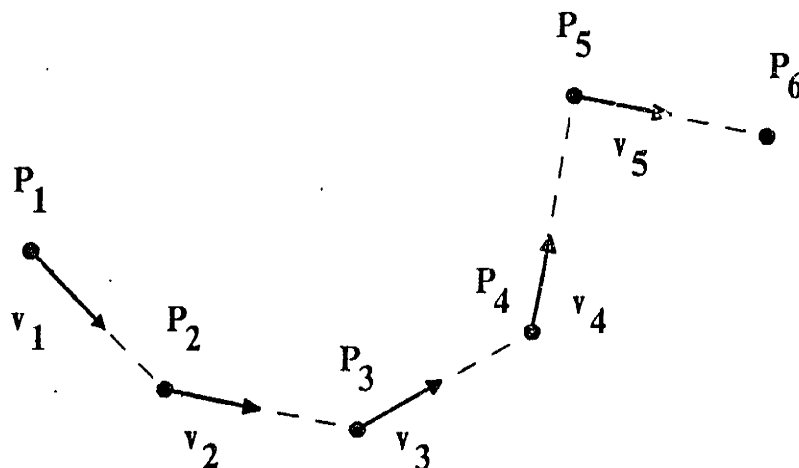


Figure 3.23: Relating Position and Unit Velocity State Variables

### 3.5.1 Determining State Parameters

Given the peg and hole system in a state with  $n$  degrees of freedom, the number of state variables that define that state will be  $2 \times d.o.f. - 1$ .<sup>28</sup> We also notice that this is again equal to the number of *total* uncertainties of our system defined in Section 3.2.3. We can break down the number of these parameters into those that specify position ( $n$ ) and those that specify the unit velocity ( $n - 1$ ). One way that we can think of these two groups of state variables is to say that the position variables represent discrete points along a particular trajectory of the peg, and the unit velocity variables represent a way to connect one position to the next along this trajectory. This is illustrated in Figure 3.23

Recall the velocity error cone of Section 3.2.4. With it we examined qualitatively the relationship between position and unit velocity errors, as well as the relationship between allowable unit velocity error and the distance between discrete positions. We will now use a slightly modified version of this concept to establish bounds on the unit velocity state variables in terms of position state

<sup>28</sup>See Section 3.3.5.2

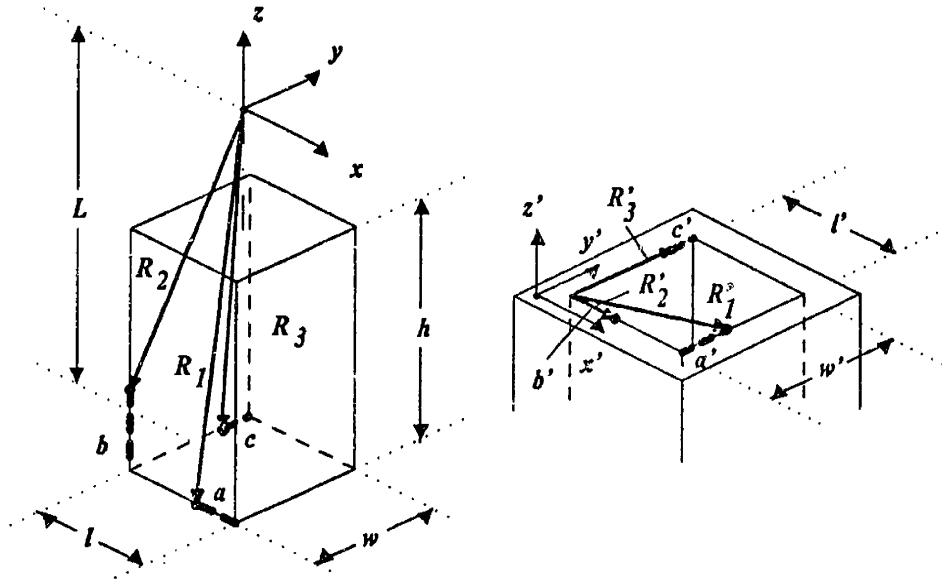


Figure 3.24: Case One Revisited

variables. For the purposes of illustration, we will again use case one as an example, and have repeated an earlier figure for reference in Figure 3.24. For case one, we chose the state variables  $(\theta_{x'}, \theta_{y'}, \theta_{z'}, \dot{a}', \dot{b}')$ .

As we mentioned in Section 3.3.5.1, the choice of roll-pitch-yaw angles to explicitly represent position results from the computational complexity of determining position in another equivalent set of variables. If we are willing to do a little numerical iteration, we can determine what angles would give us the desired set of linear edge contact parameters. We can therefore, for a given state, implicitly represent our state variables as  $(a', b, c', \dot{a}', \dot{b}')$ . Note that the position and velocity terms now correspond except for the absence of the term  $\dot{c}'$ . This is due to the fact that we normalized  $\dot{c}'$  to be  $-1$ .<sup>29</sup>

In general, the velocities of the contact points will be nonlinear functions of the position of the hole relative to the peg. In other words, if we assume

<sup>29</sup>See Section 3.3.5.2

a set of instantaneous contact velocities at one position, at another position these velocities are likely to have different values. If we assume *small* motions, however, then the velocities of the contact points can be assumed to be constant over those motions. Therefore, for small motions of the hole we can write:

$$p_1 = p_0 + \dot{p}_0 \Delta t \quad (3.76)$$

where  $p_0$  represents an edge position at state 0,  $p_1$  represents the same variable at state 1,  $\dot{p}_0$  is the *linearized* edge contact velocity, and  $\Delta t$  is an arbitrary time step. So for the state variables we can write:

$$\begin{aligned} a'_1 &= a'_0 + \dot{a}'_0 \Delta t \\ b_1 &= b_0 + \dot{b}_0 \Delta t \\ c'_1 &= c'_0 + \dot{c}'_0 \Delta t \end{aligned} \quad (3.77)$$

For the third equation we remember that  $\dot{c}' = -1$ . In addition, our strategy calls for the contact defined by the variable  $c'$  to reach the corner of the hole first, i.e.  $c'_1 = 0$  while  $a'_1 > 0$  and  $b_1 > 0$ . We can therefore solve the third equation for  $\Delta t$  as:

$$\Delta t = \frac{-c'_0}{\dot{c}'_0} = c'_0 \quad (3.78)$$

and substitute the above expression back into the first two equations to obtain:

$$\begin{aligned} a'_1 &= a'_0 + \dot{a}'_0 c'_0 \\ b_1 &= b_0 + \dot{b}_0 c'_0 \end{aligned} \quad (3.79)$$

In order to ensure that  $c'_1$  reaches zero first from a given starting position, we can solve the above two equations for  $a'_1 \geq 0$ , and  $b_1 \geq 0$  respectively to obtain:

$$\begin{aligned} \dot{a}'_0 &\geq -\frac{a'_0}{c'_0} \\ \dot{b}_0 &\geq -\frac{b_0}{c'_0} \end{aligned} \quad (3.80)$$

We now have expressions that represent the lower bounds of our two unit velocity state variables in terms of our initial position state variables that will guarantee the desired transition configuration. To represent upper bounds on the unit

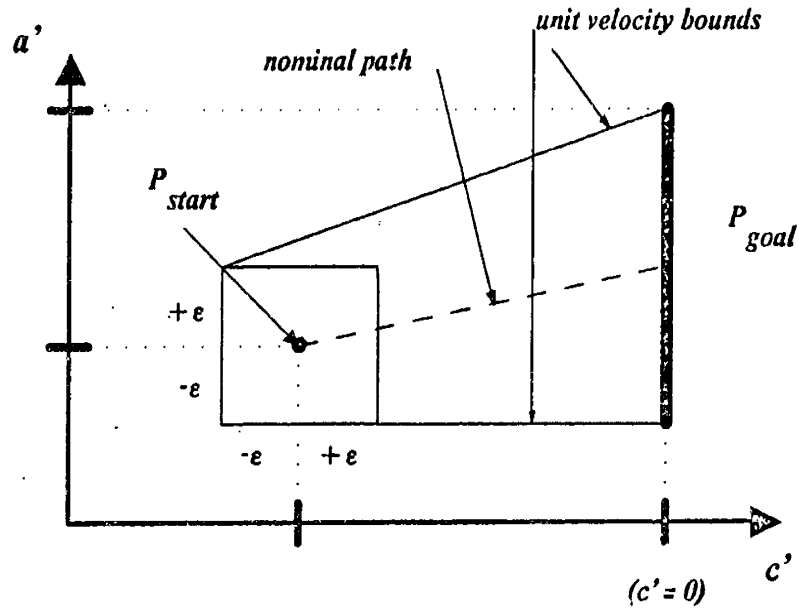


Figure 3.25: The Linearized Velocity Error Cone

velocity state variables, we could constrain the direction of sliding to be towards aligning the peg and the hole. In this case that would translate to:

$$\begin{aligned} \dot{a}'_0 &\leq 0 \\ \dot{b}'_0 &\leq 0 \end{aligned} \tag{3.81}$$

which constrains the position variables  $(a', b, c')$  to become smaller with time.

We could also generate similar expressions that would provide unit velocity bounds with an associated position error  $\pm\epsilon$  for each position variable, which would guarantee the desired configuration transition in the presence of positional uncertainty. A way to represent this process graphically is shown in Figure 3.25, where the axes of the figure represent the contact position state variables, and the slopes of the lines connecting the error regions represent the bounding values of the normalized edge contact velocity variables, and the dotted line represents a nominal trajectory assuming no uncertainty.

This figure represents the unit velocity *error cone* and *uncertainty ball* in terms of our state variables. It is a *linearized* analogy to the more general error



cone mentioned earlier because we have assumed constant edge velocities over our range of motion. With this figure we can visualize the relationships between our state variables as well as uncertainty. Since these state variables must be provided *a priori* in order to construct the force-moment regions of our solution space, it will serve as an important visual tool in our planning process, as well as in evaluating the sensitivity to uncertainty of the resulting strategy.<sup>30</sup>

### 3.5.2 Choosing ‘Cross-sections’ of Higher Dimensional Regions

In Section 3.4.2.2 we illustrated how a system with greater than two degrees of freedom could be represented in our three force-moment subspaces. For cases with two or fewer degrees of freedom, however, we are not as easily able to decouple the  $\frac{M_i}{F}$  dimensions of the five-dimensional solution space. We mentioned that one way around this problem would be to choose a value for one or more of the force-moment dimensions and solve the resulting equations for the remaining variables, which could then be represented in our three subspaces. As we said in Section 3.4.2.2, this would be equivalent to taking *cross-sections* of the higher dimensional constraint regions and projecting them into our lower dimensional subspaces. One problem with this approach is that if we chose a value that lies outside of the higher dimensional solution region then no solution cross-section will be found. It would therefore be necessary to iterate until the range over which such a variable is valid could be determined.

From our list of chosen contact cases of Section 3.3.2, we see that the highly constrained cases quite often involve edge-corner contacts. We recall from Section 3.3.4.1 that one of the variables appearing in the force balance equations for these cases was the angle  $\phi_m$ , which represents the direction of the normal contact vector,  $\vec{n}_m$ , at point  $m$  within a  $90^\circ$  range. We could think of this variable as if it were another normal reaction force. In other words, if we chose not to use the representation of the corner contact with the angle  $\phi_m$ , we could have added an additional force at point  $m$  such that the corner was constrained by

---

<sup>30</sup>See Section 3.6.2

two normal forces perpendicular to each other. One advantage of using the  $\phi_m$  representation instead is that we know its value must be between 0 and  $90^\circ$  to have any physical significance. Therefore, if we choose  $\phi_m$  to be the variable by which we will take cross-sections of our higher dimensional constraint regions, then we know the resulting regions will be valid if  $0 \leq \phi_m \leq 90^\circ$ . By using the  $\phi$  variables in our equations, we will therefore be able to take cross-sections of the solution regions for highly constrained cases and 'fit' them into our three-dimensional subspaces for subsequent evaluation.

One issue that arises with the *a priori* specification of  $\phi_m$  is that of *causality*. Specifically, since  $\phi_m$  is not one of the variables whose value is directly under our control, it is not clear that specifying its value is a sufficient condition to determine a unique cross-section of a higher dimensional solution region. Work on determining the uniqueness of this sectioning technique has not yet been completed. For the present we shall assume that, where necessary, we can specify the value of  $\phi_m$  in our strategies. If this assumption later proves to be incorrect, then it will be necessary to choose fixed values of the applied force-moment parameters, as mentioned earlier.

### 3.5.3 Finding Solution Regions

Having developed the techniques to specify consistent sets of state variables and represent all of the cases we shall be considering in our three-dimensional subspaces, we will now establish the techniques with which to search for solution regions within these spaces. As stated earlier, we shall favor techniques that allow us to visualize the effects of various parameters on the solution regions and therefore allow us to gain an intuitive feel for these effects.

#### 3.5.3.1 Intersecting Solution Spaces

For each state of our assembly we are now able to generate surfaces that represent the constraints on our applied forces and moments that will result in the assembly continuing in the manner desired. Each state, therefore, has associated with it a set of constraint surfaces unique to that state. These surfaces are all

represented in a common set of three 3 dimensional force-moment subspaces. If we intersect the regions defined by the constraint surfaces of these various states, the resulting intersection regions will represent solutions common to all of the states. In other words, we begin by picking a series of vectors, whose components represent the variables of a given state, in such a way that they form a discrete set of points along an assembly *path*. The intersection region of all the force-moment solution spaces associated with those points, then, will represent the forces and moments that will successfully guide the assembly along that path. We can best illustrate this process with an example.

Recall the surfaces for case one shown in Figures 3.18 through 3.20. The combined breaking contact curves of Figure 3.20 can be thought of as a collection of surfaces extending vertically into each of the three  $\frac{M_i}{F}$  dimensions.<sup>31</sup> In a similar fashion, Figures 3.26 and 3.27 show the constraint surfaces (represented in the same subspaces) corresponding to a given state of case two. In Figure 3.28 we have intersected the breaking contact solution regions for the two cases in the  $(\alpha, \beta)$  plane. The shaded region, then, represents values of the force direction angles that will not break contacts in either case.

### 3.5.3.2 Superposition of Curves in the Alpha-Beta Plane

If we intersect two or more sliding constraint surfaces in the  $\frac{M_i}{F}$  vs  $(\alpha, \beta)$  subspaces, it would be rather difficult to picture the exact shape of the resulting intersection region, as it would constitute a rather complicated three-dimensional volume. Fortunately, there is an alternative way of representing this form of intersection. Figure 3.29 shows the curves defined by the intersection of the two  $\frac{M_x}{F}$  constraint surfaces, for cases one and two, projected into the  $(\alpha, \beta)$  plane. The shaded regions in this figure determine the values of  $\alpha$  and  $\beta$  that lie within non-jamming regions common to cases one and two. By performing similar operations for the surfaces in the  $\frac{M_y}{F}$  vs  $(\alpha, \beta)$  and  $\frac{M_z}{F}$  vs  $(\alpha, \beta)$  subspaces and projecting all of the curves, including those for breaking contacts, into the same

---

<sup>31</sup>Corresponding to surfaces that vary in two dimensions and are swept into the third. See Section 3.4.2.2

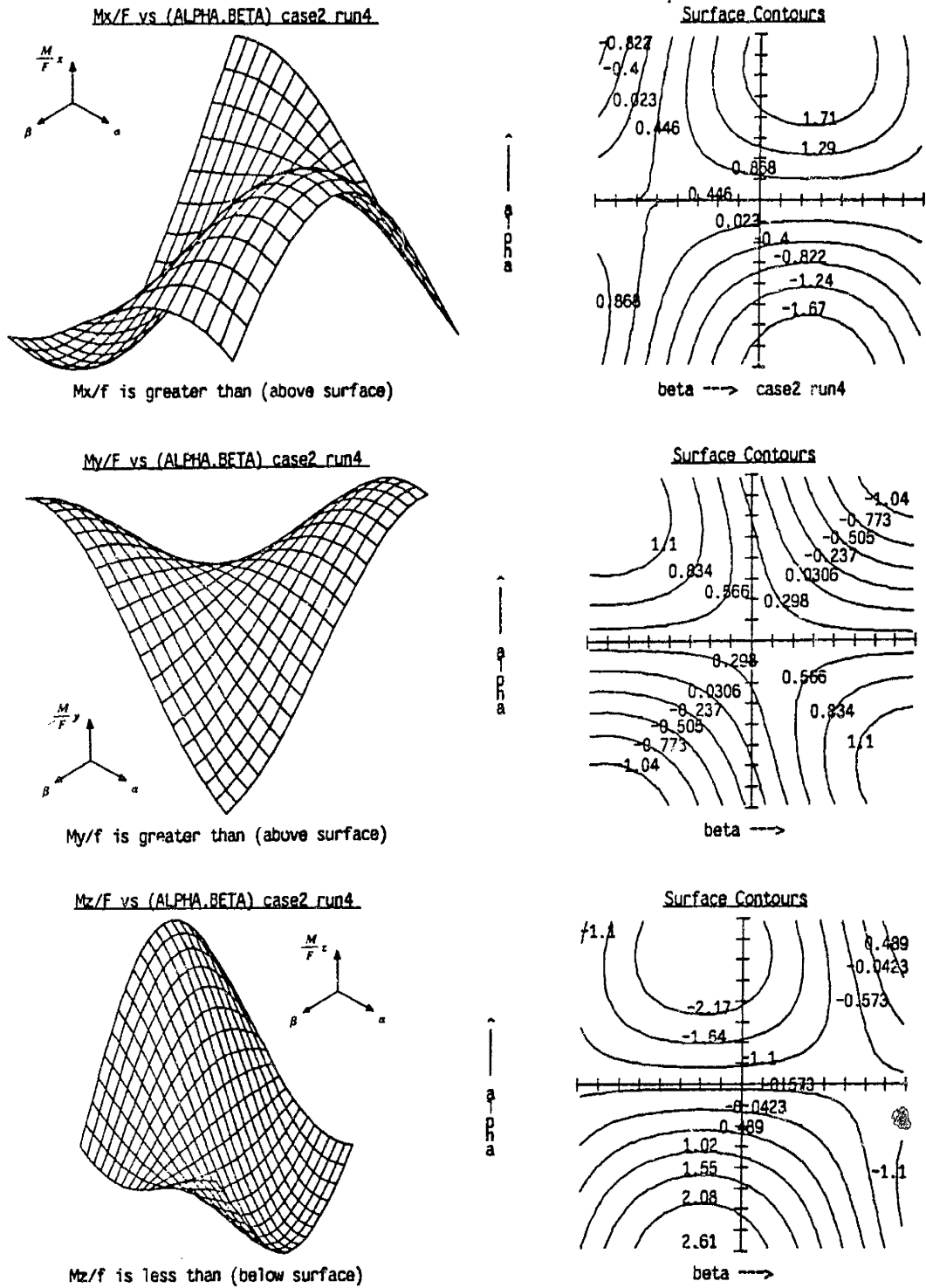


Figure 3.26: Solution Regions for Case Two

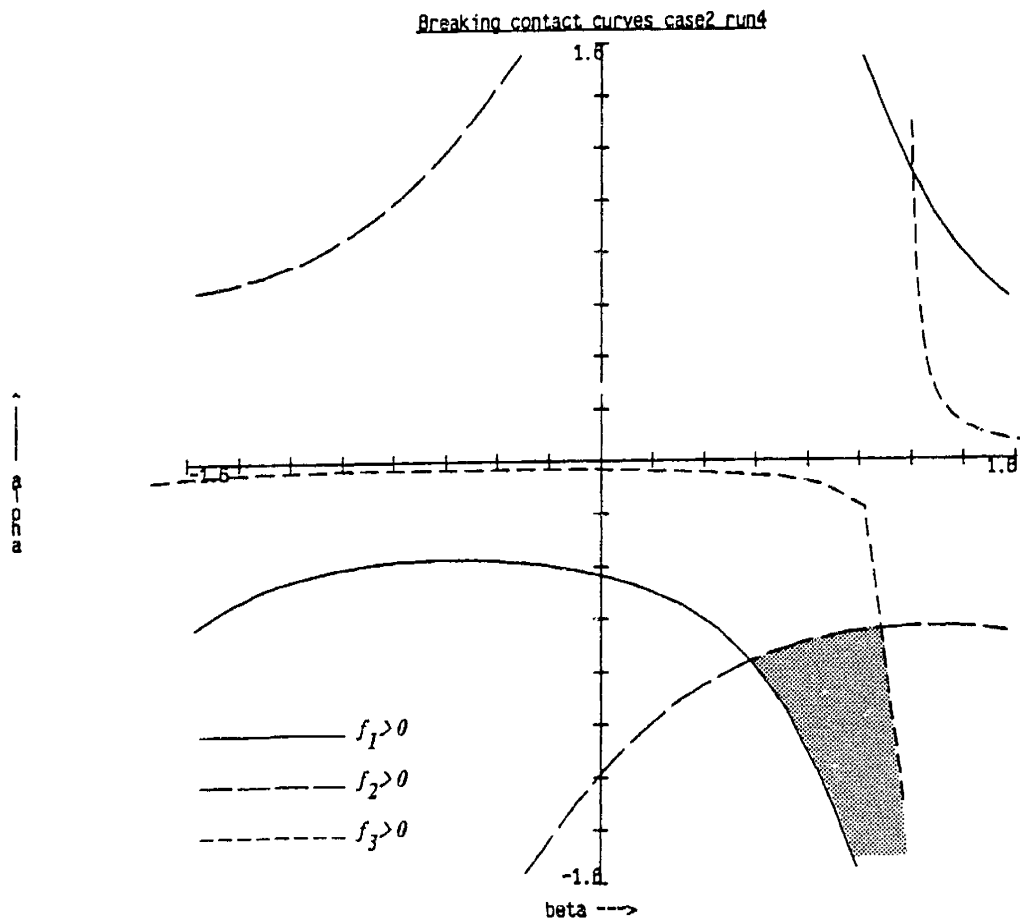


Figure 3.27: Solution Regions for Case Two, (Cont.)

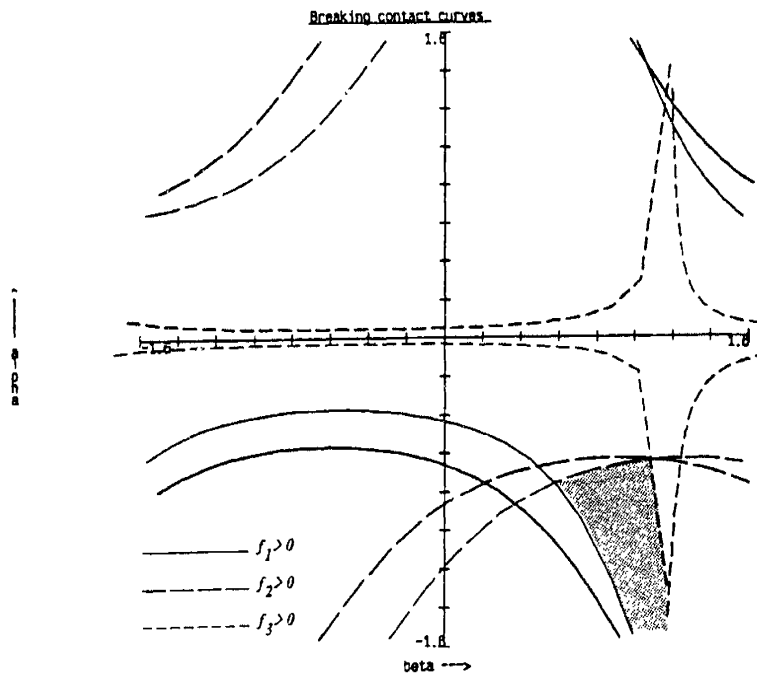


Figure 3.28: Intersection of Breaking Contact Constraint Curves

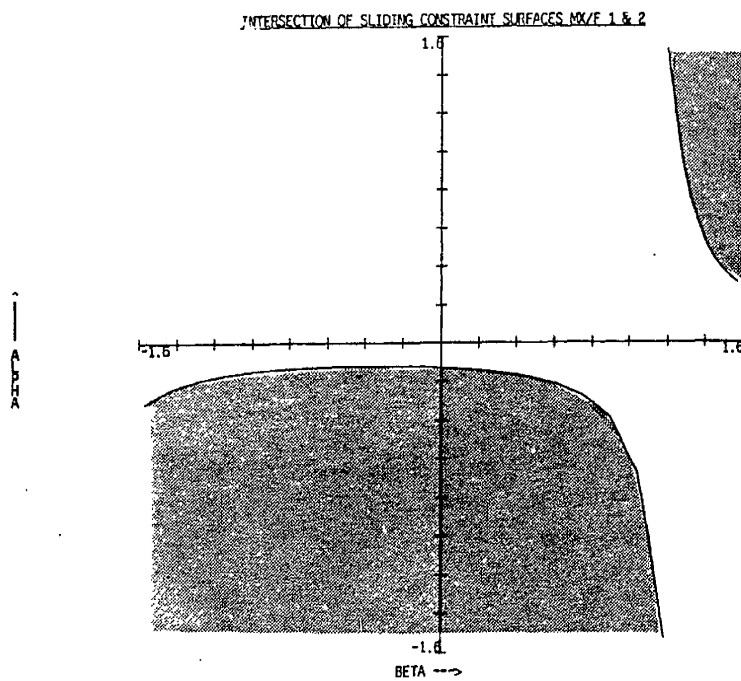


Figure 3.29: Intersection of Sliding Boundaries Projected into Two Dimensions

$(\alpha, \beta)$  plane, we will end up with a region defining the values of  $\alpha$  and  $\beta$  that will neither jam nor break any contacts with the proper associated moments. To determine what the resulting ranges allowed for the  $\frac{M_i}{F}$  terms are, we can pick values for  $\alpha$  and  $\beta$  within the solution region and back solve the constraint equations for them explicitly.

To find a force-moment solution region valid for the entire assembly path, we can extend this procedure to include the solution regions for all of the contact states under consideration. Our search for an overall solution region, then, reduces to a two-dimensional search for intersection regions in the plane defined by the applied force angles  $\alpha$  and  $\beta$ . In general, a single solution valid over the entire assembly path will not be found on the first iteration, and a number of iterations on the set of state vectors will be required, equivalent to re-specifying the nominal assembly path. This iteration and re-specification may also have to include new configurations as well as states.<sup>32</sup>

### 3.5.4 Multi-Step Strategies

It is possible that even after a number of iterations have been made, a single solution region may not always be found. There may be cases where the conservative assumptions regarding jamming and breaking contacts will prove to be too restrictive and need to be modified to find a solution. It is quite possible, for example, that an assembly may require more than one set of applied force-moment vectors to guide it along a given assembly path. We will now examine how these multiple applied force-moment vectors may be determined when necessary.

If we are able to find a single force-moment solution region which is valid over the entire assembly path from initial state to fully assembled end state, then all that needs to be done is to select a point within this region to define a single applied force-moment vector to control the assembly. The only question that remains is how robust this solution region is in the presence of uncertainty.

---

<sup>32</sup>As an example of this, the cases one through four listed in Section 3.3.2 were the result of a number of iterations, with the final set being different from the first.

The issue of robustness shall be dealt with in the next section. In general, it is worthwhile to make at least a few iterations on the assembly path to determine if such a single solution is likely. After these iterations, the designer should be able to determine if his solutions seem to be converging towards a single region. The decision to give up searching for a single step strategy and adopt a multi-step approach is a decision that must be made after evaluating these initial trials.

If we are not able to find a single force-moment solution region for a complete assembly path, we may be forced to choose a set of solution regions which, taken sequentially, will cover the entire assembly path. The termination state of one solution region would then become the beginning state for the next. The question here becomes one of how reliable is a given termination state in terms of establishing the initial conditions for the next segment of the assembly path, with a minimum amount of associated uncertainty. To divide our overall assembly path into a set of intermediate paths with associated intermediate solution regions, we must decide what restrictions we wish to apply during these transitions. We stated earlier in the chapter that our primary goal was to avoid jamming and breaking contacts in the search for our solution regions. In the cases where we are unable to find a single solution region we must reconsider these restrictions. For example, if we determine a solution region which slides the peg into the hole in the first three of four states in an assembly path but leaves the peg jammed in the fourth, it will be necessary to apply a different force-moment vector which will carry the peg from this jammed state to the desired end state. One possible strategy, then, may be to allow the peg to jam in this fourth state, and then continue with a new force-moment vector, forming a two step assembly strategy. Although a two step strategy (or more generally a *multi-step* strategy) is less desirable than a single step strategy since it would tend to preclude a simple or passive implementation, it still represents a valid assembly strategy. The termination state then becomes the immediate goal state as far as the previous path segments are concerned. All restrictions concerning the selection of position and unit velocity state variables required to reach this state reliably will still hold.

As an example of when a multi-step strategy would be required, consider the



transition between the solution regions of two cases,  $\mathcal{A}$  and  $\mathcal{B}$ , of the peg and hole assembly. Figure 3.30 shows the intersection regions for the  $\frac{M_y}{F}$  subspace of cases  $\mathcal{A}$  and  $\mathcal{B}$  in the  $(\alpha, \beta)$  plane. The intersection between sliding constraint regions does not overlap the intersection of the breaking contact regions shown, so a separate force-moment vector must be used to satisfy the sliding solution regions for each case separately. The resulting motion will end with the peg jammed in the beginning state of case  $\mathcal{B}$ . To ensure a stable initial state for the next motion, the new force-moment vector will have the same  $\alpha$  and  $\beta$  components, ensuring that no contacts shall be broken in the transition process.

In considering multi-step strategies it will be necessary to rank the various constraints in terms of their importance in defining a robust intermediate termination state. Of the two major constraints considered for the single step strategies, jamming and breaking contact, the breaking contact constraint represented the most important restriction since breaking contacts resulted in a higher degree of uncertainty. A jammed state on the other hand maintained the number of contacts, but represented a termination of motion along the assembly path. Therefore, for multi-step assembly strategies, breaking contact constraints represent the primary restriction relating the separate solution regions, with jamming becoming a secondary consideration. In terms of our three force-moment subspaces, then, the intersection of contact maintaining regions in the  $(\alpha, \beta)$  plane will be the first priority, with the intersection of non-jamming regions wherever possible as the second priority.

Developing a multi-step strategy, then, is analogous to developing a single step strategy except that the *goal* state from a previous step will determine the initial conditions for the next.<sup>33</sup> The resulting applied force-moment vectors can be represented by a chain of discrete points in the three force-moment subspaces. Each point will lie within a solution region valid over some segment of the total assembly path. We will say that the chosen assembly path (discrete set of state vectors) and applied force-moment vector(s) together form an assembly *trajectory*. Figure 3.31 shows an assembly path in the space of position parameters (configuration space), as well as the associated *path* in force-moment

<sup>33</sup>See [Erdmann 84].

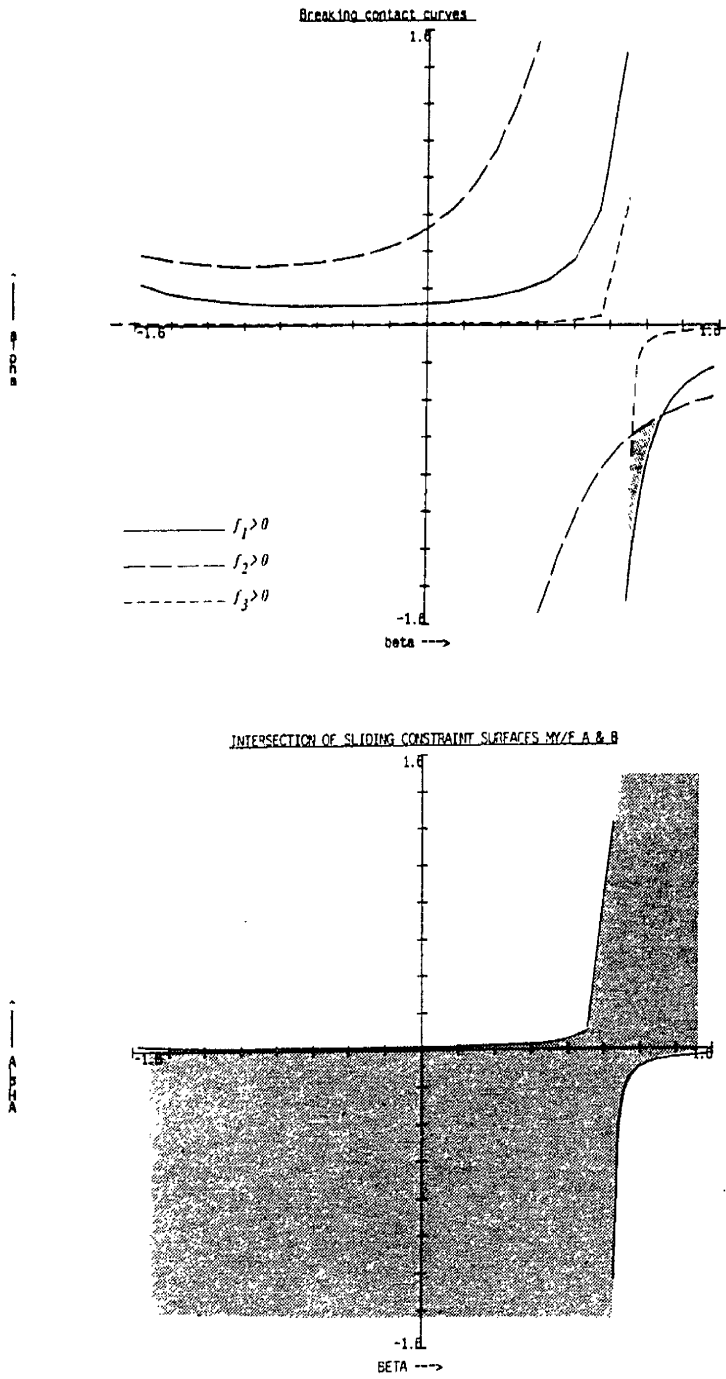


Figure 3.30: Non-Intersecting Solution Regions

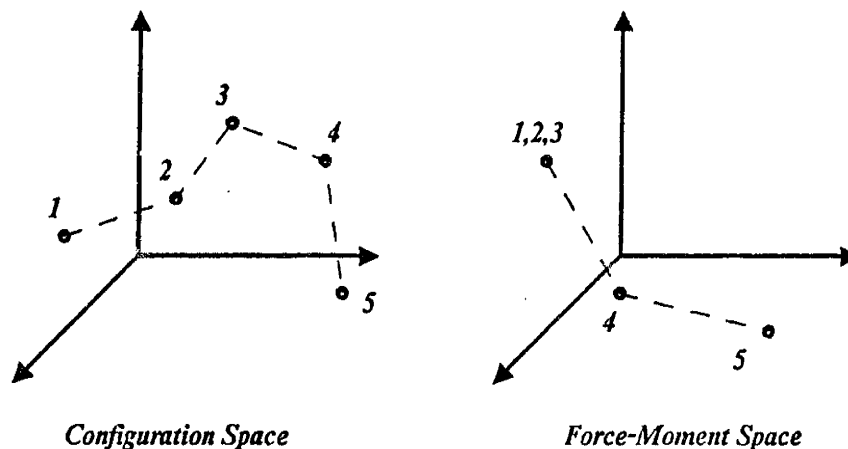


Figure 3.31: The assembly Trajectory

space. Each point chosen in force-moment space will have associated with it one or more points in configuration space. The points in force-moment space represent the applied force-moment vectors valid for the corresponding states. These combined sets of (force-moment/position state variable) points represent points along the overall assembly trajectory. This trajectory, then, combines the projected assembly path in configuration space with the controlling applied force-moment path in force-moment space, and is the final product of the strategy development process.

Another consideration in allowing our system to become jammed during a multi-step strategy is that of wedging. We recall from Section 2.2.2 that if a system is jammed in a state where some components of the reaction forces point towards each other, reaction forces could arise that were uncontrollable by means of the applied force and moments. In order to reduce the risks of wedging, then, it will be necessary to limit the magnitude of the applied force such that the reaction forces due to the elastic deformation of the parts are insignificant by comparison. In the fourth chapter we shall see how the issue of wedging will be

taken into account when considering implementations of strategies.

The next step towards using the resulting strategy to control an actual assembly lies in the implementation of the assembly trajectory utilizing some passive device, if possible, or an active control law relating the various parameters of the trajectory.<sup>34</sup> No assumption is made concerning the time domain in the development of these strategies. In other words, the models used to generate a multi-step strategy, as well as for a single step strategy, will not determine the amount of time required before switching to the next applied force-moment vector in the assembly trajectory. The determination of this switching parameter, either in terms of time or position, will depend on the particular form of implementation selected for the strategy.

## 3.6 Evaluation

Having now developed a strategy and with at least one iteration of the design loop completed, it is time to compare and evaluate the results to determine if another iteration is in order. To perform this evaluation adequately we will have to consider issues such as sensitivity of the resulting strategies to uncertainty, their simplicity in terms of implementation, and the generality of the strategies as applied to other assemblies. We will begin by again considering the issue of uncertainty.

### 3.6.1 Determining the Effects of Uncertainty

One of the reasons we considered controlling an assembly with the applied forces and moments in the first place is the belief that the resulting behavior of the system would be relatively insensitive to uncertainty in position. Assuming that we have eliminated the possibility of encountering unanticipated configurations by properly choosing our state variables, we will consider the behavior of our force-moment solution regions within the neighborhood of a given state. A solution region which varies markedly with a small change in position will be

---

<sup>34</sup>See Section 1.3.3

considered to be sensitive to positional uncertainty and therefore less desirable than a more insensitive region.

In terms of an assembly trajectory, we can say that a solution region in force-moment space that is strongly coupled with its associated state vector in configuration space is less robust than an equivalent solution space and force-moment pair that are less strongly coupled. Since the components of this state vector represent both position and unit velocity, this coupling will be a function of both the position and unit velocity of that state. In addition, since the limits on the unit velocity components can be expressed in terms of the position components of that state and neighboring states (see Section 3.5.1), the resulting mapping of sensitivity from force-moment space to configuration space is solely a function of the position states of the chosen assembly path.

To establish a quantitative means of comparing the sensitivity of a given assembly trajectory to that of another trajectory, we can write the following relation:

$$S_{\mathcal{T}} = \left| \frac{\partial \mathcal{V}_{\mathcal{T}}}{\partial P} \right| \quad (3.82)$$

where  $S_{\mathcal{T}}$  is defined as the *sensitivity factor* of a given trajectory  $\mathcal{T}$ ,  $\mathcal{V}_{\mathcal{T}}$  is the *volume* of the associated solution region in force-moment space, and  $P$  is the nominal assembly path. Since the assembly path of our trajectory  $\mathcal{T}$  would be made up of discrete states, we can discretize the evaluation of  $S_{\mathcal{T}}$  for each state along the path. In terms of our three force-moment subspaces and the positional uncertainty term  $\epsilon$  we can rewrite Equation 3.82 for a given state  $A$  of trajectory  $\mathcal{T}$  as:

$$S_A \approx \left| \frac{\mathcal{V}_A - \mathcal{V}_{A+\epsilon}}{\epsilon} \right| \quad (3.83)$$

where the volume of the solution region for state  $A$  has the units of *length*<sup>3</sup> and the positional uncertainty  $\epsilon$  has unit of *length*. The sensitivity factor of state  $A$ , then, has the units of *length*<sup>2</sup>.

For a given state the bounding surfaces of our force-moment solution regions are functions of  $\alpha$  and  $\beta$  only, so we can explicitly write the sensitivity factor as:

$$S_A \approx \frac{1}{\epsilon} \sum_i \left[ \int_{\beta=-\frac{\pi}{2}}^{\beta=\frac{\pi}{2}} \int_{\alpha_{\text{bound}}} \int_{\frac{M_i}{F} \text{ bound}} d \frac{M_i}{F} d\alpha d\beta \right] \quad (3.84)$$

for  $i = x, y, z$ , and where

$$\frac{M_i}{F_{bound}} = \left| \frac{M_i^A}{F} - \frac{M_i^{A+\epsilon}}{F} \right|$$

$$\alpha_{bound} = [ \alpha_1^A \cap \alpha_2^A \cap \dots \alpha_n^A ] \cap [ \alpha_1^{A+\epsilon} \cap \dots \alpha_n^{A+\epsilon} ]$$

where  $\frac{M_i}{F_{bound}}$  are the bounding surfaces that vary in three dimensions,  $\alpha_{bound}$  are the bounding surfaces that vary in two or less dimensions ( $\alpha$  and  $\beta$ ), and  $n$  is the number of contacts in case  $A$ . To judge which of two selected trajectories is more sensitive in terms of the states along those trajectories, then, we can compare the average sensitivity factors of each. The trajectory with the lower average sensitivity factor can be considered to be less sensitive to positional uncertainty and hence more robust. In addition, since we are able to evaluate each discrete state along a given assembly trajectory, we can optimize that trajectory to make the total sensitivity factor a minimum.

The above sensitivity factor provides an approximate quantitative measure of the sensitivity of solution regions to positional uncertainty in a given state. As such, the sensitivity factor is a useful addition to the previously presented visual techniques to evaluate sensitivity of a given strategy. We note that a similar procedure may be used to determine the sensitivity of the force-moment solution regions to variations in other parameters as well. For example, if we use  $\epsilon$  to represent variations in, say, the coefficient of friction  $\mu$ , or the tolerances of our parts ( $l' - l$ ), etc., then the above expression for  $S_A$  becomes a general sensitivity factor for state  $A$  for any parameter of interest.

### 3.6.2 Propagation of Errors Between Discrete States

Recall the linearized error cone of Section 3.5.1. With it we were able to represent graphically the bounds imposed of the unit velocity components of our state vector. In addition, we were able to illustrate the relationship between the maximum uncertainty at the beginning state of a given contact case and the maximum uncertainty at the end state of that case.<sup>35</sup> Since the states that comprise our assembly path are discrete in nature, there is no way to represent

<sup>35</sup>See Figure 3.25

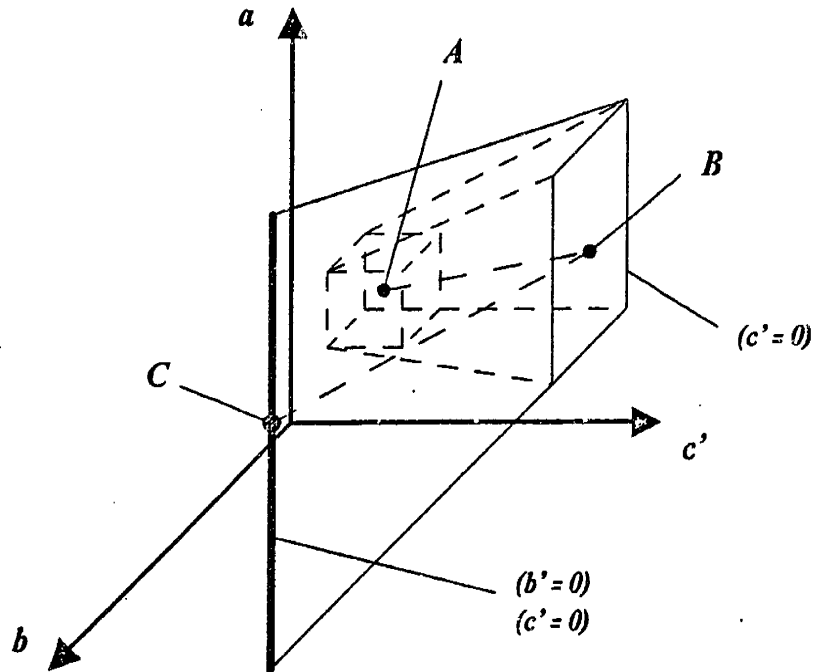


Figure 3.32: Linking Discrete States Via the Velocity Error Cone

the effects of uncertainty acting continuously along that path. However, given the discrete error cone connecting two states, we are able to approximate the propagation of errors along our assembly path.

Figure 3.32 shows a series of states linked by error cones in a three-dimensional linear configuration space. State  $A$  is the initial state of case I, which has three degrees of freedom and therefore has a three-dimensional error 'cube' of length  $2\epsilon$  on a side. State  $B$  represents the end state of case I and the beginning state of case II. Case II has only two degrees of freedom so the uncertainty region is now a two-dimensional rectangle. The dimensions of this rectangle are determined by the extreme values of the remaining independent position state variables that intersect other, possibly undesirable, configurations. In other words, if state variables  $a$  or  $b$  lie outside the rectangle in state  $B$ , one of the dependent contact position variables would become less than or equal to zero, indicating an additional contact had been made or an existing contact had been broken. The

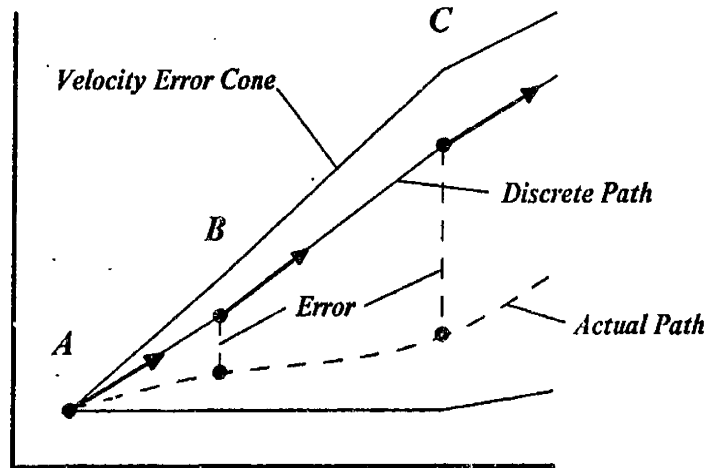


Figure 3.33: Propagation of Linearization Errors

plane containing this new uncertainty rectangle is the plane defined by  $c' = 0$ . State  $C$  represents the end state of case II and the beginning state of case III. Case III has only one degree of freedom so its uncertainty region is a line.<sup>36</sup> The endpoints of the line represent extreme values of  $a$  that represent two new configurations. State  $C$  could be the goal state of a strategy, an intermediate state in a strategy, or a sub-goal of a sub-strategy. These connected error cones serve as a visual as well as analytical means to bound our assembly path in terms of the maximum anticipated positional uncertainty of our system.<sup>37</sup>

In Section 3.5.1 we indicated that the velocities between states would be considered constant. One consequence of this assumption was the requirement that the distance between discrete states be small. In general this linearization assumption, as well as other assumptions and simplifications will introduce errors that will also be propagated along the assembly path. Figure 3.33 illustrates this error propagation. Assuming no uncertainty for the moment, the position at state  $A$  is known precisely. A linear velocity is assumed to move the system

<sup>36</sup>Note the analogy to the uncertainty regions in Figure 3.3.

<sup>37</sup>See [Brooks 82].



from state  $A$  to state  $B$ . The actual path of the system is shown as a dotted curve, and the actual state  $B$  is at a different position than projected. This process is repeated in moving the system to state  $C$ , where the error continues to grow. As we stated before, the inherent system uncertainty here is assumed to be zero. The propagated errors we are seeing, then, are an example of model based uncertainty introduced by our simplifications. Unlike inherent system based uncertainty, we have some control over this kind of uncertainty. To avoid the risks associated with model based uncertainty we have to be careful to examine our assumptions continually and be as conservative as possible with our restrictions. In the case of the errors shown in the figure, for example, we have chosen a set of velocity error cones that will contain these propagated errors.<sup>38</sup>

### 3.7 Summary

In this chapter we have modeled the insertion of a three-dimensional rectangular peg into a rectangular hole. By using a set of modeling elements that describe the interactions between simple straight edges and flat surfaces, we have tried to extend the generality of these models to include a wide variety of rectangular parts.

We have outlined a set of heuristics to aid in the development of assembly strategies and identified a set of contact configurations that would be sufficient to describe an assembly path. In addition, we have determined a set of initial conditions that reduces the positional precision required of the device performing the assembly.

The representation of the state of an assembly in terms of a convenient set of linear contact parameters was introduced. In order to represent the limits that uncertainty places on the specification of an assembly path, the use of a linearized position uncertainty region and velocity error cone were also introduced.

We have again chosen the applied force and moment as the control variables of the insertion. We have extended the concept of the applied force-moment

---

<sup>38</sup>Note that the range of errors propagated in this fashion will in general be bounded by the geometric constraints of the assembly.

solution space developed in the last chapter and represented within this space the constraints imposed by friction on the range of forces and moments that would allow an assembly to proceed. Due to the higher dimensional nature of this force-moment space for three dimensions, methods of simplifying the representation of constraint surfaces into lower dimensional subspaces were developed. These subspaces were then used as a convenient domain in which to intersect and compare constraints for different configurations of the peg and hole.

In cases where intersection regions might not be found that span an entire assembly path, the concept of the multi-step strategy was introduced. Here a strategy was considered that allowed jamming to occur between commanded motions under controlled circumstances. An overall assembly strategy, then, would consist of a set, or chain, of compliant motions that would connect an initial configuration to a desired goal configuration. This additional strategy was presented as an extension of the earlier planning techniques to allow them to handle cases that otherwise might have not appear to possess solutions.

Finally, the means of evaluating and comparing the relative sensitivity of strategies to uncertainty was presented in order to guide the iteration towards the development of a robust assembly strategy.

# Chapter 4

## Implementation

Up to this point we have focused our attention on the development of assembly strategies and have not primarily concerned ourselves with the issues involved in their implementation. In particular, while developing our assembly strategies in the last two chapters we have avoided the explicit specification of position and velocity terms wherever possible and chose the applied force and moment as the primary control variables of our system. By considering only the applied forces in controlling the assembly, we have implicitly assumed that we could implement a strategy in pure force control.<sup>1</sup>

In this chapter, we shall provide two examples of strategy implementation. In the case of the two-dimensional peg and hole of Chapter 2, we shall present a passive device which successfully performs chamferless insertion in the presence of significant positional uncertainty and tight part tolerances. For the case of the three-dimensional rectangular peg and hole, we shall present an implementation based on active force control using a robot.

Implementation of compliant motion strategies, both in terms of analysis and actual hardware, involves many interesting and difficult issues. Our aim in this chapter will be to provide a sampling of some of these issues as they relate to our understanding of how strategies may be better designed with their implementation in mind. As we shall see, a passive implementation will in general

---

<sup>1</sup>See Section 1.3.3.

be faster, simpler, and more robust for a given set of part parameters; while an active (computer controlled) implementation will often have lower performance, but may be more easily modified and applied to a wider range of parts. Indeed the decision to pursue a passive versus active strategy implementation will invariably depend upon the associated tradeoffs between performance and flexibility for a given case.

The second major goal of this chapter will be to evaluate experimentally the results of our previous derivations and calculations. In particular, by providing a proof of concept, we hope to verify that the models developed and assumptions made throughout the strategy development process are valid representations of the actual physical systems.

## 4.1 2D Example: A Passive Implementation

Figure 4.1 shows a simple planar mechanism implementing a passive insertion strategy. The material of the peg, hole, and linkage is aluminum, and the white background is Teflon sheet used to reduce external friction. The Clearance between the peg and the hole shown is 0.002 inch and the measured static coefficient of friction is  $\mu_s \approx 0.6$ . The total time required for a typical insertion using this device is on the order of one second. If the linkage is made to move much faster, inertial effects become evident and the peg tends to jam.

The mechanism consists of four links interconnected by four joints, three rotary and one sliding, as shown. Neglecting rigid body motion of the entire mechanism in the plane, the linkage has one degree of freedom. There are no energy storage elements incorporated into the mechanism (i.e. springs), and the only non-conservative forces and moments acting between the links are those due to friction. In order to analyze the equivalent force and moment exerted on the peg, we shall examine the mechanism in quasi-static equilibrium. Figure 4.2 presents a schematic illustration of the linkage of Figure 4.1. The force and

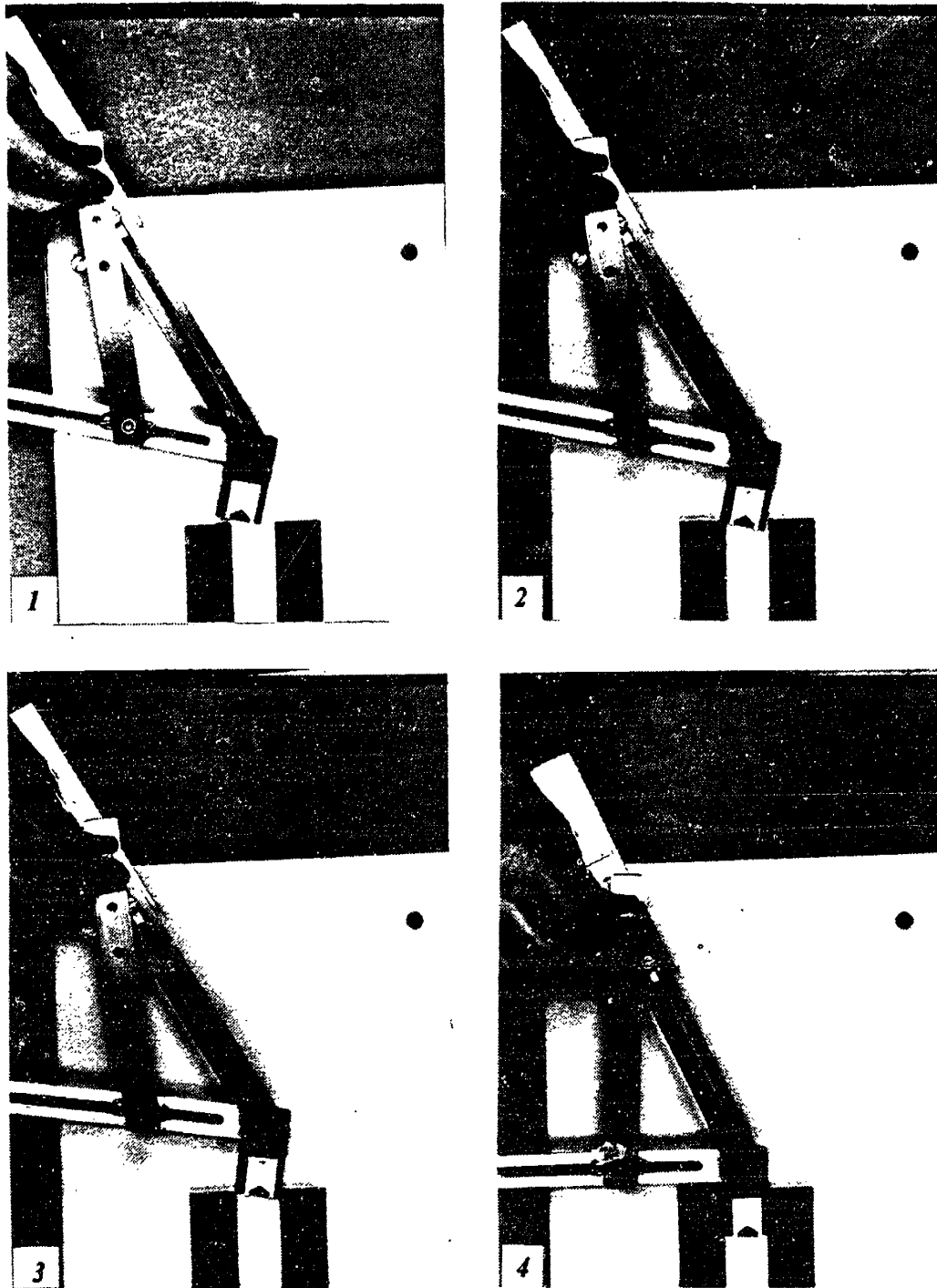


Figure 4.1: Mechanism Performing Insertion

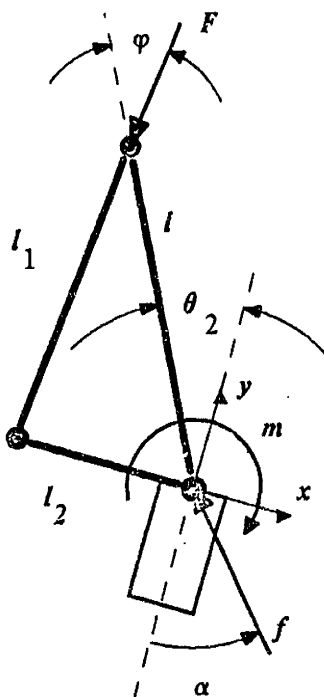


Figure 4.2: Chamferless Insertion Mechanism

moment balance equations are:

$$\begin{aligned}\sum F_x &= -f \sin \alpha + F \sin (\theta_2 - \phi) = 0 \\ \sum F_y &= f \cos \alpha - F \cos (\theta_2 - \phi) = 0 \\ \sum M_0 &= m - l F \sin \phi = 0\end{aligned}\quad (4.85)$$

From the kinematic constraints of the linkage, using the law of cosines, we can write:

$$l = l_2 \sin \theta_2 + \sqrt{l_1^2 - l_2^2 \cos^2 \theta_2} \quad (4.86)$$

Combining these expressions, we derive the following applied force and moment equations:

$$\begin{aligned}\frac{m}{f} &= [l_2 \sin \theta_2 + \sqrt{l_1^2 - l_2^2 \cos^2 \theta_2}] \sin \phi \\ \alpha &= \theta_2 - \phi\end{aligned}\quad (4.87)$$

where  $\frac{m}{f}$  and  $\alpha$  are opposite in sense to the equivalent terms applied directly to the peg in the analysis of chapter two.

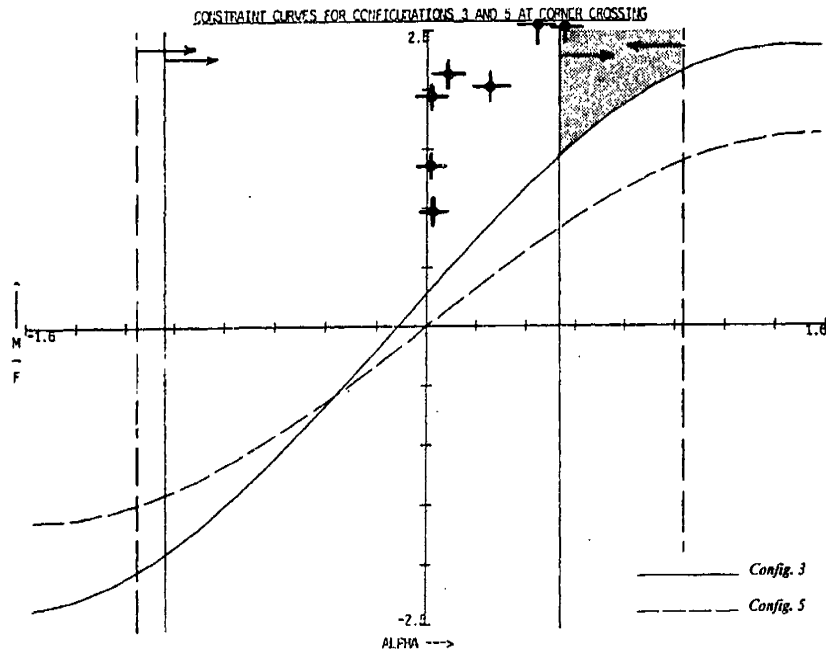


Figure 4.3: Experimental Results from the Planar Insertion Mechanism

Using the quasi-static model of the planar linkage, experiments were performed and the equivalent applied forces and moments acting on the peg were determined. The mechanism parameters used in the experiment were

$$l_1 = 5.50 \text{ inches}$$

$$l_2 = 2.80 \text{ inches}$$

The results of these experiments were plotted in the corner crossing ( $\frac{M}{F}$  vs  $\alpha$ ) plane for the given parameters of the peg and the hole and are shown in Figure 4.3. As we can see from the distribution of data points, the mechanism succeeded in performing insertion over a wider range of applied forces and moments than predicted by our model. We notice that all of the points recorded correspond to values that should allow the peg to slide (i.e., are above the appropriate  $\frac{M}{F}$  curves), but that many would theoretically correspond to values of  $\alpha$  that would cause contacts to be broken. We recall from the plots of Figures 2.24 and 2.25 that the breaking contact limits on  $\alpha$  were strongly influenced by the

value of  $\mu$  whereas the sliding constraint curves were less so. This would indicate that the value of  $\mu$  chosen for our model, which we measured experimentally, could be a significant source of error in our model.

Another possible reason for the discrepancies observed is the presence of unmodeled external friction in the system, provided by the surface of the plane. Since friction is acting on the insertion mechanism as well as the peg, the magnitude and direction of the apparent force and moment acting on the peg would differ from that modeled.

## 4.2 3D Example: Implementing a Strategy on a Robot

For three-dimensional assembly, the development of a passive assembly mechanism is not, unfortunately, so straightforward. In order to demonstrate the validity of our results from Chapter 3, we shall choose to implement our assembly strategy on a robot operating under force control. The parameters from our three-dimensional analysis that we shall be using for our implementation are the initial conditions of the peg and hole, and the applied forces and moments derived from the force-moment subspace regions for cases 1 through 4. The plots from which these force and moment were derived are contained in Appendix 3.

### 4.2.1 The MIT Compliant Motion System

Figure 4.4 shows the M.I.T. Artificial Intelligence Lab's PUMA 600 robot which has been extensively modified to run both in position and compliance control modes under the high level command of a Symbolics 3600 lisp machine and the lower level command of two PDP 11/23 minicomputers. Attached between the wrist and end effector of the robot is a 6 axis strain gauge force sensor which allows the PUMA to sense forces along and moments about a set of cartesian axes located at the wrist. The compliance mode of the PUMA used in the experiments conducted for this thesis was implemented in the form of the



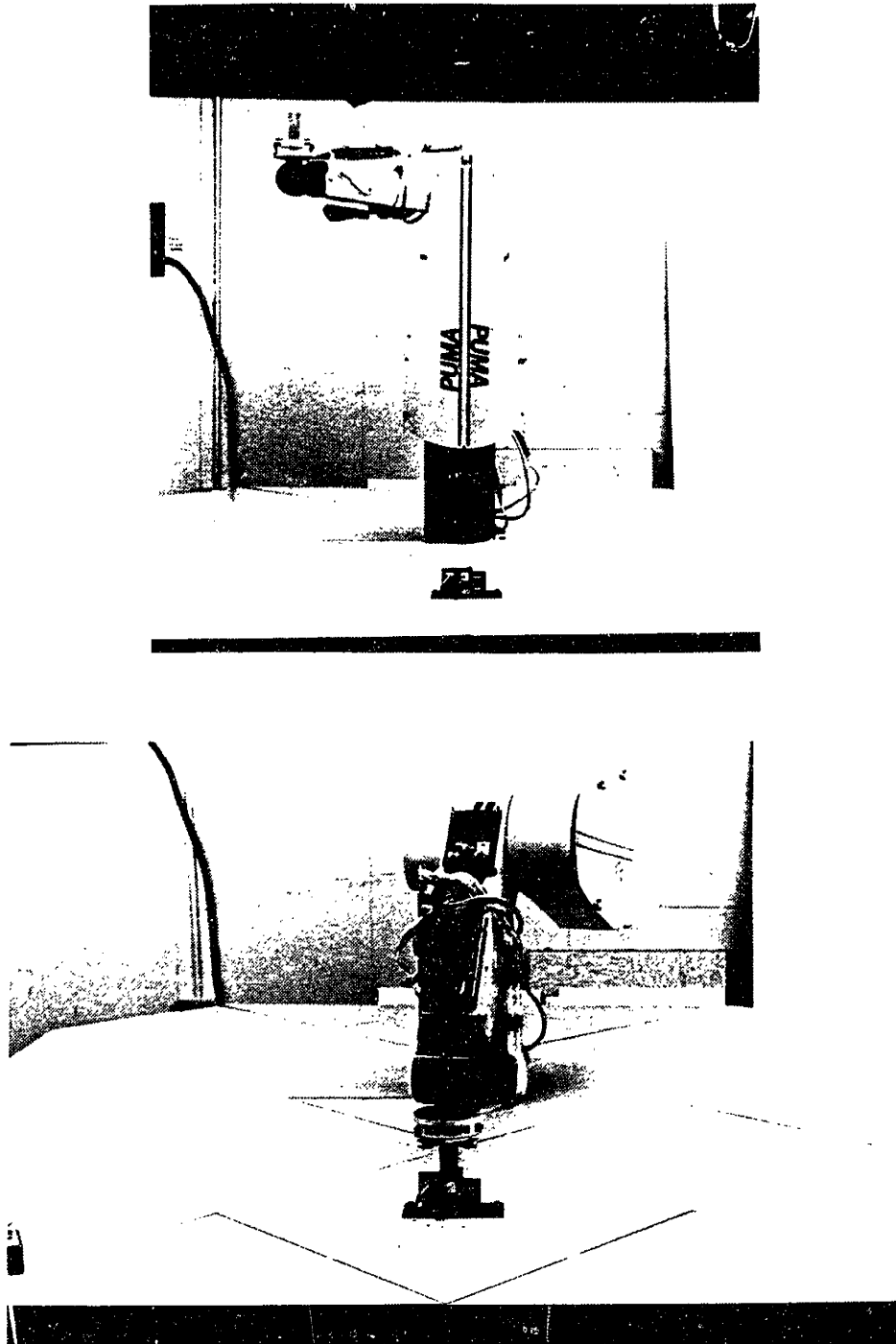


Figure 4.4: PUMA 600 Robot, Modified for Active Compliance

compliance equation:

$$\Delta\dot{\theta} = B_j^{-1}J^T[(F_C - F_S)] \quad (4.88)$$

where:

$\Delta\dot{\theta}$  The velocities to be added to the joint velocities of a non – compliant trajectory, a 6 vector.

$B_j^{-1}$  Inverse of the joint damping matrix, a 6 element diagonal matrix.

$J^T$  Jacobian transpose in the frame of the compliance center, a  $6 \times 6$  matrix.

$F_C$  The commanded cartesian force – torque vector, a 6 vector in the compliance center.

$F_S$  The transformed force – torque vector as measured by the wrist sensor.

For the purposes of our implementation we shall be specifying the commanded forces and moments to be exerted by the robot. Although we have commanded no velocities the implementation of compliance on the PUMA incorporates a significant amount of joint damping  $B_j^{-1}$  in order to maintain the stability of the system while in contact with the environment.<sup>2</sup> As a result, the speed of the resulting implementation will be reduced considerably in the presence of this joint damping.

## 4.2.2 Experimental Setup

Figure 4.5 shows the experimental setup used for inserting a rectangular peg into a rectangular hole. The rectangular peg, shown in Figure 4.6 is made of machined steel and is fixed to the force gauge at the end of the PUMA's wrist. In order to improve the stability of the force control algorithm, 0.125 inch rubber spacers are placed between the wrist and peg mounting plates to act as a low pass filter on the sensed force signals.

The hole into which the peg will be inserted, shown in Figure 4.7, is also made of machined steel. The sides of the hole are bolted to each other through horizontal slots, as shown. These slots allow the dimensions of the hole to be varied. The hole assembly is then rigidly bolted to the work surface. The

<sup>2</sup>See [Jones and O'Donnell 85].

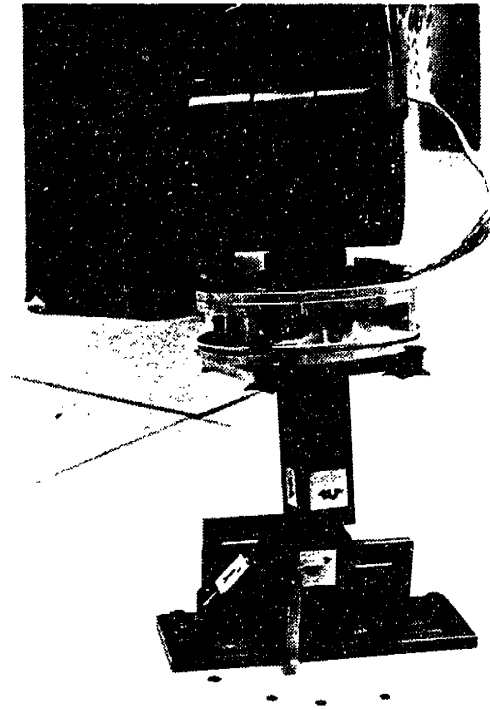


Figure 4.5: Setup for the Three-Dimensional Peg and Hole

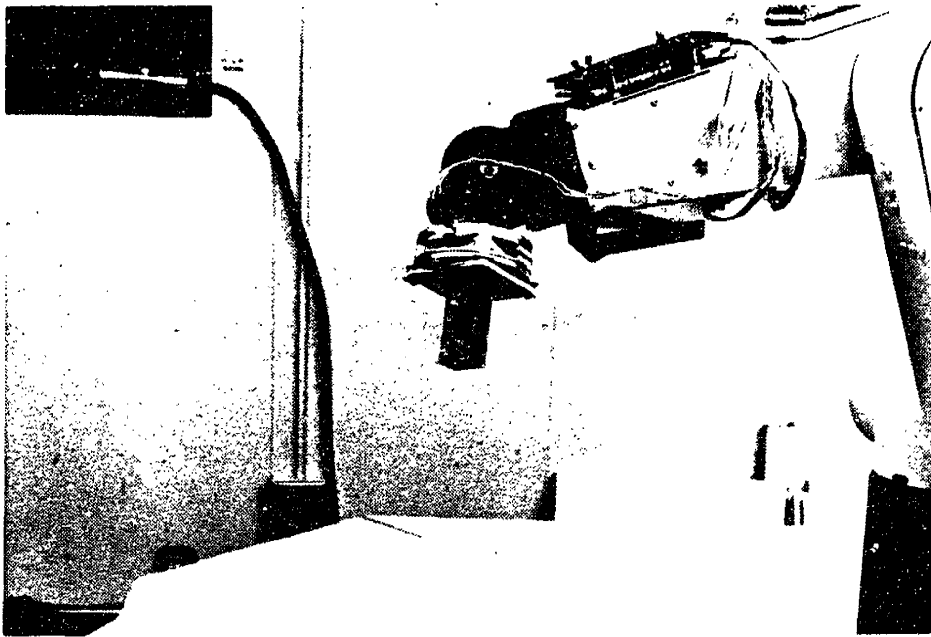


Figure 4.6: Rectangular Peg

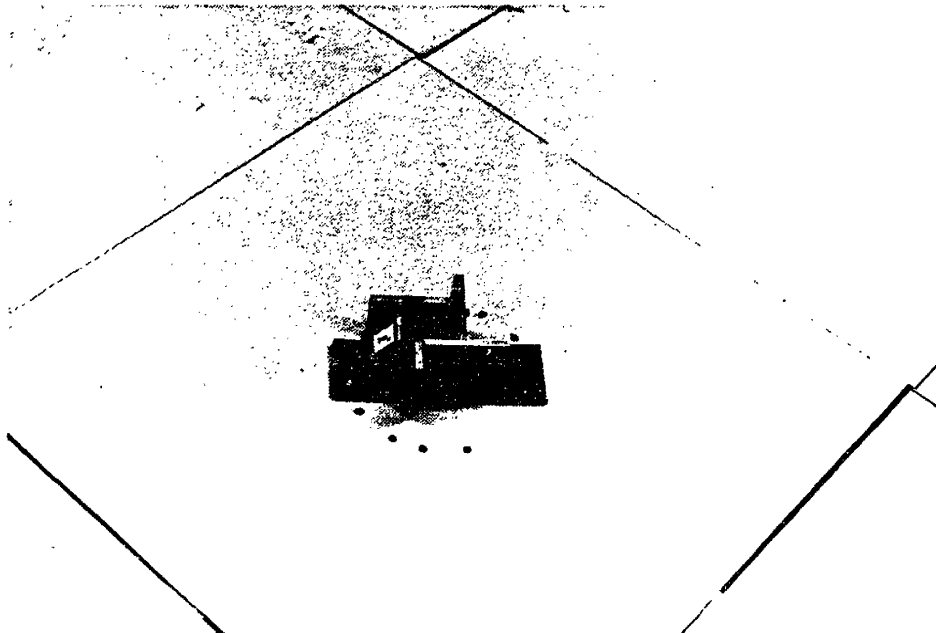


Figure 4.7: Adjustable Rectangular Hole

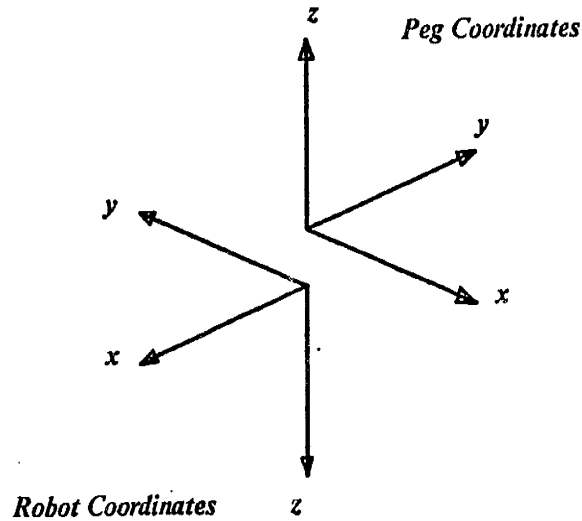


Figure 4.8: Wrist Coordinates of the PUMA

parameters of the peg and hole used in the experiment are:

$$\begin{aligned}
 l &= 1.250 \text{ in} \\
 w &= 1.000 \text{ in} \\
 l' &= 1.255 \text{ in} \\
 w' &= 1.005 \text{ in} \\
 L &= 2.000 \text{ in} \\
 \mu &= 0.5 \quad \text{steel on steel}
 \end{aligned}$$

The coordinate frame of the wrist of the robot, shown in Figure 4.8, is different from the peg coordinate frame in which we specified our commanded forces and moments. In order to specify the proper forces and moments, then, we shall have to transform them into the wrist coordinates of the robot. The transformation for this is simply

$$\begin{aligned}
 x_{peg} &= -y_{robot} \\
 y_{peg} &= -x_{robot} \\
 z_{peg} &= -z_{robot}
 \end{aligned}$$

In addition, the origin of the wrist coordinate frame of the PUMA is located 6.5 inches from the tip of the peg. Since our strategy was developed for a tip to coordinate frame distance of only  $L = 2.0$  inches, we shall have to scale our moments about the  $x$  and  $y$  axes of the robot by a factor of 3.25.

We recall from Section 3.3.5.1 that the orientation of the peg was determined in roll-pitch-yaw coordinates.<sup>3</sup> We also recall that another way to represent a given position of the peg in a given contact case was to determine a set of linear edge contact parameters ( $a', b$ , etc.) and then iterate to determine the corresponding roll-pitch-yaw angles. To set up the peg in initial contact with the hole, then, we simply place the robot in compliance mode and move the peg by hand to the desired position.

### 4.3 Results

The commanded forces and moments in the wrist coordinates of the PUMA were:

$$\begin{aligned} F_x &= 26.8 \text{ ounces} \\ F_y &= 19.6 \text{ ounces} \\ F_z &= 22.4 \text{ ounces} \\ M_x &= -139.2 \text{ in - ounces} \\ M_y &= 160.0 \text{ in - ounces} \\ M_z &= 8.8 \text{ in - ounces} \end{aligned}$$

These values correspond, after scaling, to the following values derived from the force-moment intersection regions in Appendix 3.

$$\begin{aligned} \alpha &= -0.98 \text{ radians} \\ \beta &= 0.94 \text{ radians} \\ \frac{M_x}{F} &= -1.23 \text{ inches} \\ \frac{M_y}{F} &= 1.07 \text{ inches} \\ \frac{M_z}{F} &= -0.07 \text{ inches} \end{aligned}$$

---

<sup>3</sup>Actually, we specified the position of the hole relative to the peg.

The Specific values chosen for the applied forces represented a tradeoff between system resolution and the stiffness of the peg and hole hardware. Specifically, the force control algorithm of the PUMA is capable of exerting forces and moments within a resolution of approximately  $\pm 10$  ounces (in-ounces) about a nominal selected value. This resolution limitation places a lower bound on the commanded forces and moments that will meet the requirements of a given strategy. On the other end of the scale, the problem of part deformation and wedging places an upper bound on the allowable commanded forces.<sup>4</sup> The values chosen, then, represent a reasonable compromise between these two limitations.

Once the peg was positioned relative to the hole and the desired forces and moments entered into the controller, the PUMA was commanded to begin complying. Figures 4.9 and 4.10 show the progression of the assembly.

During the assembly, marks were made on the peg along the edges of the hole. Figure 4.11 shows this process and the resulting visual record of the assembly. The total time required to complete the insertion shown was approximately 1 hour and 10 minutes. As we can see, this performance does not compare favorably with that of the passive implementation of Section 4.1. One reason for the low speed of the implementation was the large amount of robot joint damping required to maintain stability during contact between the peg and the hole.

Despite the rather low performance of the robot implementation, the experimental results do appear to indicate that the nominal forces and moments derived from our strategy development procedure are valid given the limitations of the robot's force resolution capability and variations in the established initial conditions. Although operating very slowly, the robot did succeed in carrying out the insertion in the presence of a significant amount of positional uncertainty. The portion of the assembly path that consisted of the peg entering the hole (edge crossing, cases one through three) took a little over a minute to complete while the remaining 69 minutes consisted of the peg slowly sliding deeper into the hole (case four).

---

<sup>4</sup>See Section 2.2.2.

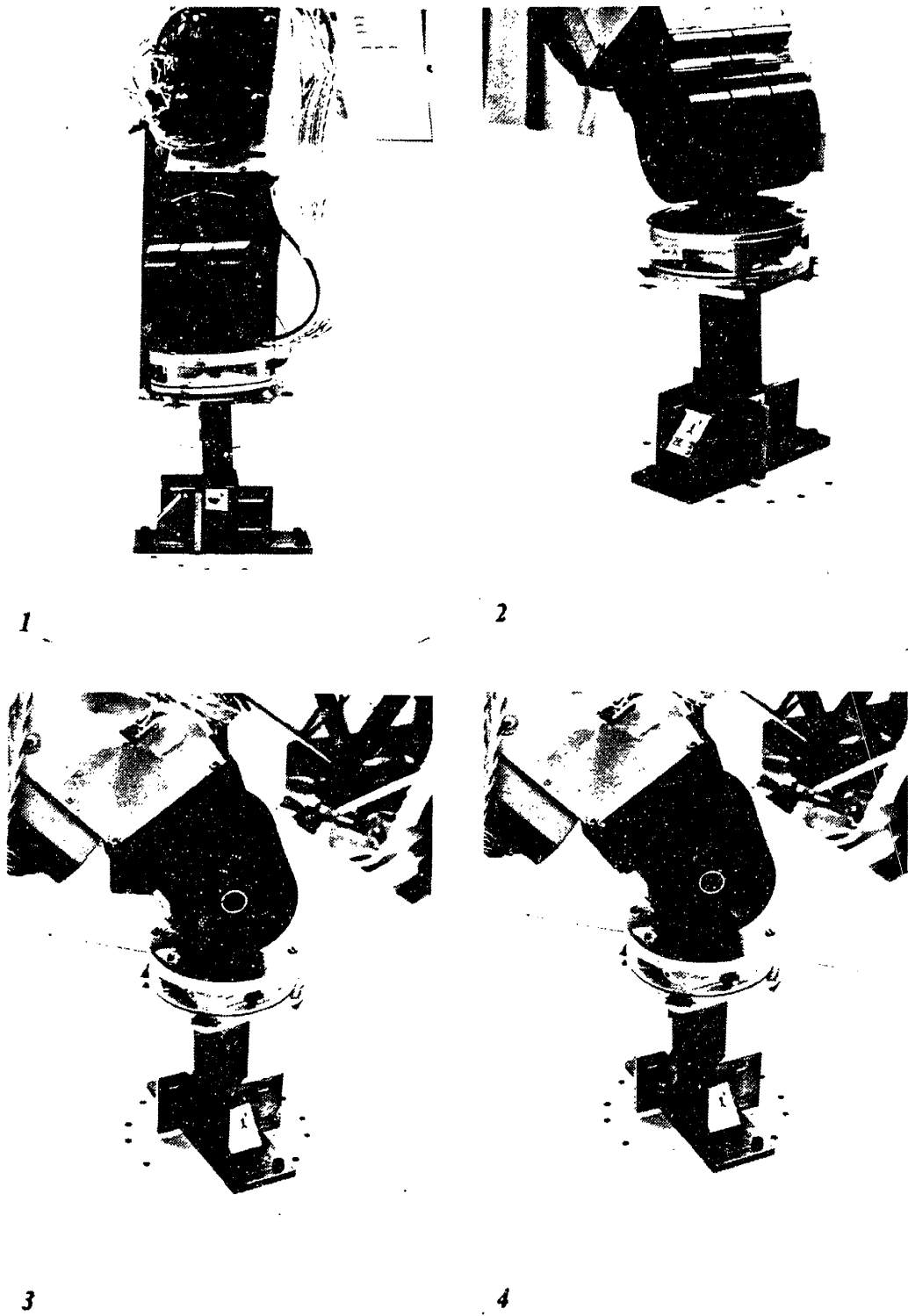


Figure 4.9: PUMA Performing Peg in Hole Insertion



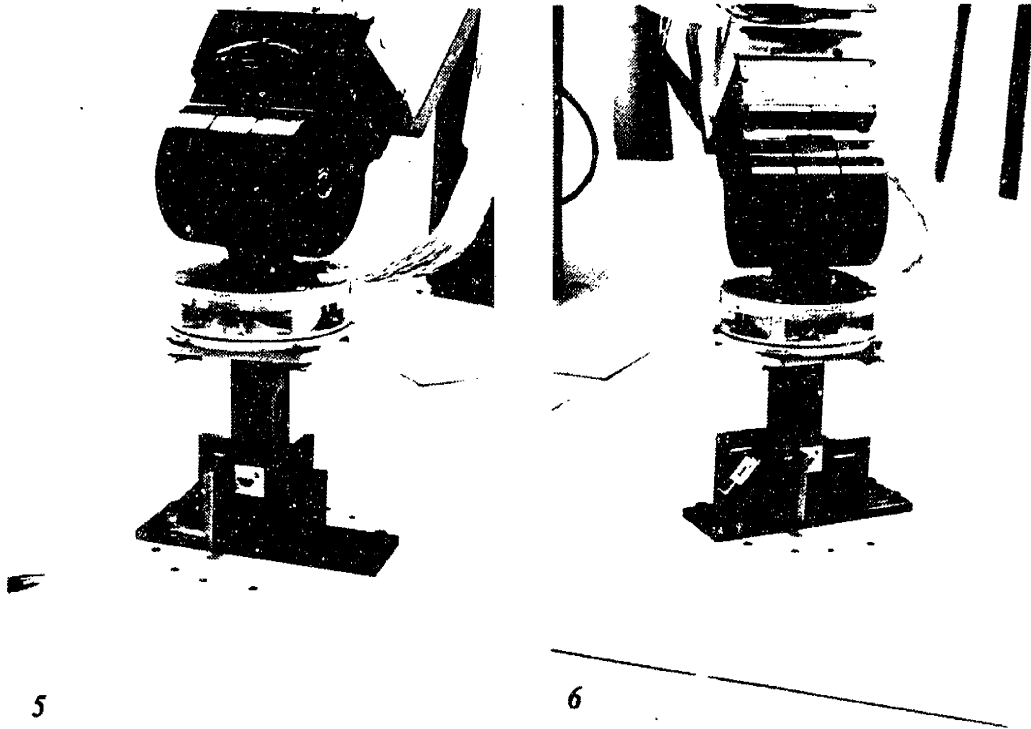


Figure 4.10: PUMA Performing Peg in Hole Insertion, (Cont.)

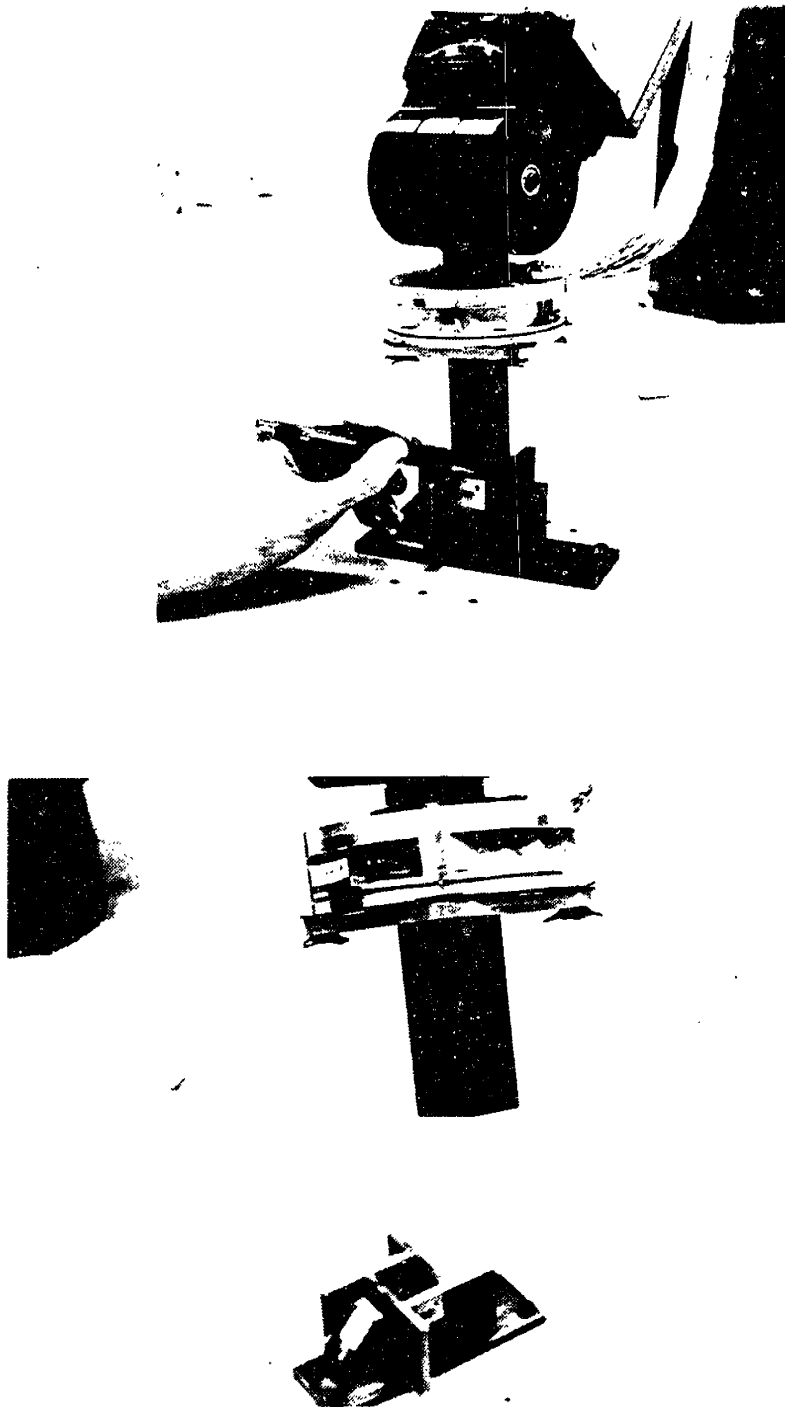


Figure 4.11: Visually Recording the Assembly

# Chapter 5

## Conclusion

### 5.1 Summary

In this thesis we have developed compliant motion strategies that successfully insert a rectangular peg into a rectangular hole without the aid of chamfers. The modeling and analytical techniques that have been developed along the way have given us the means with which to plan assembly in three dimensions. By controlling our system with a set of applied forces and moments, and representing motion constraints in terms of limits on those forces and moments, we have reduced the sensitivity of our solutions to positional uncertainty. We have determined a set of initial conditions that are insensitive to small positional variations, and in so doing, have reduced the overall positioning requirements and increased the robustness of the assembly system.

The applied force-moment constraint space represents a convenient domain in which many of the factors that affect assembly can be combined and studied. The response of a system to changes in various parameters, such as position or tolerancing, can be represented visually as changes in the corresponding constraint surfaces. The search for forces and moments that would guide an assembly towards completion can then be reduced to searching for intersecting regions, bounded by constraint curves, in a two-dimensional plane.

By limiting ourselves to the quasi-static domain we were attempting to

reduce the complexity of modeling and analyzing three dimensional systems. When it was determined that the combined jamming and breaking contact constraints necessary to guarantee a reliable strategy could be too restrictive under some circumstances, we extended our strategies to allow multiple force-moment vectors to be applied sequentially. Transition states between the application of new force-moment vectors would allow the system to be jammed under pre-determined conditions.

Finally, by separating the applied force and moment terms from the position and velocity terms as much as possible in our analysis, we have not tied our resulting strategies to any one particular means of implementation. In fact, we were able to devise different implementation strategies from the same form of analysis for the two and three-dimensional rectangular peg and hole examples, and demonstrated the validity of the resulting strategies by means of their implementation.

## 5.2 Generality of the Planning Process

The strategy development process outlined in this thesis is an intermediate strategy planning technique in that many of the decisions and parameter specifications are left up to a designer instead of being determined automatically. In Figure 3.22 of Section 3.5 we presented a flow chart of the strategy development process outlining the iterative nature of the design process of assembly planning. Much of this iteration results from the limitations of our models and the assumptions necessary to make the problem manageable. A natural question to ask, then, is how general are our strategy development techniques and the assembly strategies that result from their use. Putting it another way, have our assumptions and simplifications limited the generality of the strategy development process to any great extent and, if so, how may we evaluate and possibly overcome some of these limitations? In order to address these questions, we shall compare our planning procedure to some more general (and as of yet unimplemented) high level procedures developed for fine motion planning.

### 5.2.1 Higher Dimensional Constraint Regions

Recall from Section 3.4.2.2 that the solution regions in the 5 dimensional applied force-moment space represented the set applied forces and moments that would cause an assembly to slide at a *specific position* and in a *given direction*. We chose to model only discrete states of the assembly in order to reduce the tremendous complexity involved in developing models valid for all states. Each state, by our definition, consisted of the independent set of position and unit velocity parameters that allowed the equilibrium condition of the assembly to be uniquely defined. In order to specify the force-moment constraints over the entire assembly path, then, it was necessary to discretely pick points along that path. Solution regions for each point, or state, were then intersected to determine if any general solution regions existed over the path or some segment of it. In order to validate this *discretization* of the assembly trajectory, we had to assume that the force-moment constraint surfaces were well behaved functions of position.<sup>1</sup> The only conditions under which we could not make this assumption were at configuration transition points, where we knew from the results of Chapter 2 that these transitions often accompanied large and discontinuous changes in the associated constraint regions.

Figure 5.1 illustrates conceptually the relationship of these *single state* force-moment constraint regions to the constraint regions valid for all positions and velocities. A state defining a set of regions in our 5 dimensional force-moment space would map to a single discrete point in the state space of the assembly, and would represent a cross-section, or subspace, of this higher dimensional constraint region. An assembly path would be a set of points, or in the limit a continuous curve, projected onto the surface of this higher dimensional region. Each point on the curve would represent a state (position and corresponding instantaneous unit velocity) of the system along that path. The assembly trajectory, then, would be the resulting curve represented in the entire force-moment-position-velocity space.

We recall that the number of independent parameters necessary to describe

---

<sup>1</sup>See Section 3.2.1.5.

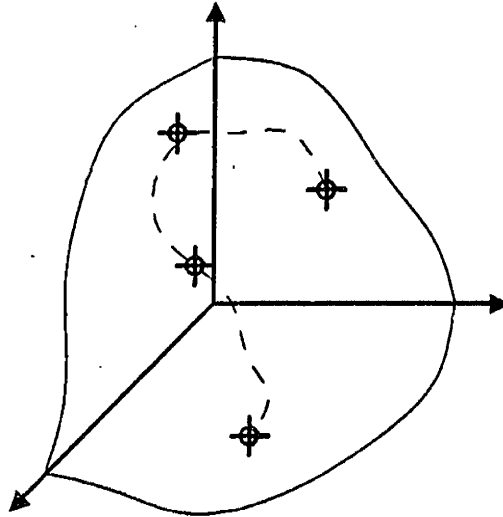


Figure 5.1: Discrete Points on a General Constraint Surface

a state of a system was  $2 \times dof - 1$ . From this we can conclude that, for the quasi-static domain, the *full state* constraint space would have 16 dimensions, consisting of the five applied force-moment parameters  $(\alpha, \beta, \frac{M_x}{F}, \frac{M_y}{F}, \frac{M_z}{F})$ , six position parameters, and five velocity parameters. In the case of the three-dimensional peg and hole assembly, where we started out in a configuration with only three degrees of freedom, we could represent the constraints in a 13 dimensional full state constraint space as long the peg never exceeded three degrees of freedom.

We mention the full state constraint space here only for completeness. To make the actual assembly planning process tractable, given the complexity of searching a 16 dimensional solution space, it will be necessary to choose an assembly path ahead of time, as we have done. By using planning tools such as the velocity error cone developed in Section 3.5.1 and heuristics similar to those of Section 3.2.1, we have tried to circumvent some of these complexities wherever possible.

### 5.2.2 Relationship to More Formal Techniques

If we examine the full state constraint space more closely, we see that the six position state variables we have added to the five unit velocity state variables and five force-moment parameters represent the dimensions of the configuration space of the assembly. This is not surprising since we have already seen that the configuration space is a natural domain within which to represent the geometric constraints specifying an assembly path.

A more formal technique for synthesizing fine motion strategies automatically is that outlined by Lozano-Pérez, Mason, and Taylor [83], which we mentioned briefly in Section 1.3.4. In this planning approach (also known as the LMT approach) everything, including friction, is represented in the configuration space domain. First order damping is assumed to govern the motion of the system, so the applied forces and moments are directly related to the difference between the commanded and actual velocity via the damping matrix. The jamming constraints in this approach are represented as constraints on the commanded velocity of the system.<sup>2</sup>

Because applied forces and moments are linked by the specification of damper dynamics to the difference between the commanded and actual velocities, the LMT approach maps the force and moment constraints directly into the state space of the assembly. In particular, by specifying commanded velocities of the system, the applied forces and moments are determined automatically in terms of the difference between these velocities and their components projected onto the configuration space constraint surfaces. The backprojections described by Erdmann [84] automatically determined a chain of connected regions in configuration space stretching from the final goal configuration to the specified initial configuration of the system.

The chain of backprojected regions is analogous to the assembly path we specified in state space. Backprojections, however, determine a continuous path (or volume) that explicitly represents the limits on the path imposed by uncertainty. In order to determine the bounds on the discrete points along our

---

<sup>2</sup>Here the term velocity is assumed to represent general, i.e. both linear and rotational, velocity.

path, we employed linearized approximations of unit velocity uncertainty cones. In addition, our discrete assembly path was specified starting with an initial configuration of the assembly and ending at an acceptable goal state, instead of vice versa, with the determination of the intermediate points often requiring a few iterations. The task of determining this path, rather than being automatic, had to be performed almost entirely by the designer.

A major advantage of the LMT approach is that it does not constrain parts to only slide over one another to determine a successful strategy. In fact, any motion that does not jam an assembly and terminates in a state lying within a desired goal (or pre-goal) region constitutes a legal move, whether contacts are broken or not. The greater freedom in selecting command variables provides a greater chance for a strategy being found which will guarantee a successful assembly. This generality, however, comes at the expense of greater complexity both in terms of the models used to describe the system as well as the number of configurations that must be explicitly examined for surfaces that will cause jamming. As we saw in Section 3.2.2, the number of possible contact cases grows rapidly with the geometric complexity of the system being modeled. Consequently, there is as yet no full scale implementation of the LMT approach in three dimensions.

### 5.3 Suggestions for Future Work

As we stated earlier, by simplifying our system models wherever possible, and incorporating heuristics in the place of formal constraints in many instances, we have attempted to make the problem of planning three-dimensional assemblies more tractable while minimizing the workload on the designer. While the resulting technique is neither as formal or general as the LMT approach, it does provide a useful *intermediate level* approach to three-dimensional assembly planning. As with any area of research, however, much more work remains to be done before assembly planning techniques such as these can be put to use in industrial assembly.



- **Full Dynamic Modeling of Part Interactions**

In this thesis we assumed a quasi-static equilibrium model for our system. A more complete and accurate model would be a full scale second order model capable of determining the behavior of parts under full inertial effects. In using these models we would again have to come to terms with the issue of complexity in modeling and analysis. Specifically, by including accelerations in our specification of the state of a system, we would have increased the dimension of our state space accordingly. While it might be possible to derive constraints on these extra values similar in concept to the unit velocity error cone, it is not clear they would provide any more insight into the assembly process. It might then become necessary to use numerical simulations to examine the behavior of the system in a set of phase planes (subspaces of the full state space).

- **Modeling of Non-Rigid Compliant Parts**

In addition to modeling the inertia of parts, it would be useful to examine the behavior of non-rigid parts in assembly, such as during inelastic collisions. During our discussion of wedging in Section 2.2.2, we considered the peg to have some stiffness  $K$ . Even though we were allowing the peg to deform by means of this stiffness we assumed the deformation was small enough to be considered negligible. In assemblies involving plastic or rubber parts, for example, this rigid model assumption would be invalid. One possible approach to handle the modeling of complex flexible parts would be to utilize finite element analysis techniques. Here again, the issue model complexity will be significant.

- **Incorporating Better Models of Friction**

Throughout this thesis we have used the dry Coulomb model of friction. For a process like assembly where friction is such a significant factor, however, a more complete and accurate model would be desirable. Since friction is extremely nonlinear in nature, these extended models must usually be implemented by means of computer simulation. If compliant part models are used, the orientations of the reaction forces will also depend on

the corresponding deformation of the parts. In addition, if the parts are allowed to deform such that area contacts are possible, a new model of friction incorporating reaction torques about the normal of contact will be required.

- **Extending Models to Include General Polyhedra**

As we mentioned in Section 1.4, one of the reasons for considering rectangular parts is the fact that many relatively complex parts can be represented, in terms of their assembly, as collections of polyhedra, among the most common of which is the right parallelepiped. It would be useful, therefore, to extend the rectangular modeling techniques to include other types of polyhedra, thus building up a 'library' of modeling components from which to construct (automatically if possible) and evaluate various part representations. This process of extending our existing models would be relatively straightforward except for the added algebra associated with representing non-orthogonal edges in a set of orthogonal coordinates. In addition, the set of possible contact configurations would become somewhat more complicated due to the odd angles between edges.

- **Developing Strategies for Collections of Polyhedra**

Once general polyhedral models have been developed, it will be desirable to integrate these models into the strategy development procedure. Specifically, by representing the various interacting edges and surfaces of a part as a collection of the appropriate polyhedra, the resulting force-moment solution spaces may be used to find the proper applied forces and moments to control the system, just as for the rectangular peg and hole. The difficult part comes about in determining a complete and consistent set of contact configurations that sufficiently span an assembly path and determining the corresponding set of state vectors describing that path.

- **Improved Strategy Implementation Procedures**

The last issue we dealt with in this thesis dealt with and examined in the least detail was implementation. The process of developing the hardware

or software necessary to implement an assembly strategy was treated as a separate issue from developing the strategy itself. Because it is such an integral part of automated assembly, the issue of implementation deserves, and is receiving, much more focused attention. Specific examples of this work may be found in the areas of improved force control techniques as well as the development of devices with adjustable passive compliance. Further work is needed to unify and integrate these areas into assembly planning.

# Appendix A

## Generating the Constraint Surfaces

In this appendix we present the batch files used by MACSYMA<sup>TM</sup> to generate the kinematic constraint equations, the unit angular velocity of the hole in terms of edge contact velocities, the contact normals, and the force and moment balance equations used to generate the constraint curves and surfaces of Section 3.4.2.1 and Appendix C. For each case, the batch files are loaded and executed in the order presented, with PLOT MEC and INTSCT MEC being run on the results.<sup>1</sup>

---

<sup>1</sup>The files presented here were written for Version 304 of MACSYMA running on a DEC KL-10.

## To Plot the Constraint Surfaces

```

/* plot mec, plots solution spaces for cases 1 thru 4 */
/* Make sure that casevalue file has been loaded for the given case */
/* plot checks for the presence of f4 */

dynamalloc:print$

dateplot:false$
noprint:true$

plotnumprec:3$
n:10$
equalscale:true$

xmin:-1.57$
xmax:1.57$
ymin:-1.57$
ymax:1.57$

phis:""$

/* plot breaking contact constraint curves in ALPHA vs BETA plane */

plotmode:'[g,gr]$

if flag1=out then plot2(alpha1,beta,-1.57,1.57,concat("beta ---> (",
part(sign1,1),part(sign1,2),")"),"^||| alpha",
concat("Force Constraints for F1>0 ",caseno," ",runname," ",phis)) else
(plot2(alpha1,beta,-1.57,(root1-.01),
concat("beta ---> (",part(sign1,1),part(sign1,2),")"),"^||| alpha","",first),
plot2(alpha1,beta,(root1+.01),1.57,false,false,
concat("Force Constraints for F1>0 ",caseno," ",runname," ",phis),same,last))$
(doward_file(),dover_file())$

if flag2=out then plot2(alpha2,beta,-1.57,1.57,concat("beta ---> (",
part(sign2,1),part(sign2,2),")"),"^||| alpha",
concat("Force Constraints for F2>0 ",caseno," ",runname," ",phis)) else
(plot2(alpha2,beta,-1.57,(root2-.01),
concat("beta ---> (",part(sign2,1),part(sign2,2),")"),"^||| alpha",
"",first),
plot2(alpha2,beta,(root2+.01),1.57,false,false,
concat("Force Constraints for F2>0 ",caseno," ",runname," ",phis),same,last))$
(doward_file(),dover_file())$

if flag3=out then plot2(alpha3,beta,-1.57,1.57,concat("beta ---> (",
part(sign3,1),part(sign3,2),")"),"^||| alpha",
concat("Force Constraints for F3>0 ",caseno," ",runname," ",phis)) else
(plot2(alpha3,beta,-1.57,(root3-.01),
concat("beta ---> (",part(sign3,1),part(sign3,2),")"),"^||| alpha",
"",first),
plot2(alpha3,beta,(root3+.01),1.57,false,false,
concat("Force Constraints for F3>0 ",caseno," ",runname," ",phis),same,last))$
(doward_file(),dover_file())$

/* superimpose plots */
plotmode:'[g,gr]$

```

```

if flag1=out then plot2([alpha1],beta,-1.57,1.57,[0],"beta --->",
"~||| alpha",concat("Breaking contact curves
",caseno," ",runname),first) else
(plot2(alpha1,beta,-1.57,(root1-.01),
"beta --->","~||| alpha",
concat("Breaking contact curves ",caseno," ",runname),first),
plot2(alpha1,beta,(root1+.01),1.57,[0],false,false,"",same))$

xaxis:false$
yaxis:false$

if flag2=out then plot2([alpha2],beta,-1.57,1.57,[1],same) else
(plot2([alpha2],beta,-1.57,(root2-.01),[1],same),
plot2([alpha2],beta,(root2+.01),1.57,[1],same))$

if flag3=out then plot2([alpha3],beta,-1.57,1.57,[2],false,false,
concat("Breaking contact curves ",caseno," ",runname),same,last) else
(plot2([alpha3],beta,-1.57,(root3-.01),[2]),nameplot(plt3a),
plot2([alpha3],beta,(root3+.01),1.57,[2],same,last))$
(doward_file(),dover_file())$

xaxis:true$
yaxis:true$

/* Plot jamming constraint surfaces in force/moment subspaces and their
contours in the ALPHA vs BETA plane */

kill(xmin,xmax,ymin,ymax)$
equalscale:false$

plotmode:'[g.gr]$
n:15$
title:concat("Mx/F vs (ALPHA,BETA) ",caseno," ",runname," ",phis)$
plot3d(expmx,beta,-1.57,1.57,alpha,-1.57,1.57,signx,false,title)$
nameplot(pltx1)$
equalscale:true$
contx(zmin,zmax,n):=(lst:[],for count:0 step 1 thru n do(
lst:append(lst,[zmin+abs(zmax-zmin)/n*count]),lst)$
zmin:zmin3d$
zmax:zmax3d$
title:"Mx/F="+expmx$
replot(true,contour,"beta --->","~||| alpha",title),contours:contx(zmin3d,
zmax3d,n)$
(doward_file(),dover_file())$
zmin:zmin3d$
zmax:zmax3d$
n:10$
title:"Surface Contours"$
replot(true,contour,concat("beta ---> ",caseno," ",runname),"~||| alpha",
title),contours:contx(zmin3d,zmax3d,n)$
nameplot(pltx2)$
replot4(pltx1,pltx2),contours:contx(zmin3d,zmax3d,n)$
(doward_file(),dover_file())$
equalscale:false$

n:15$
title:concat("My/F vs (ALPHA,BETA) ",caseno," ",runname," ",phis)$
plot3d(expmx,beta,-1.57,1.57,alpha,-1.57,1.57,signy,false,title)$

```

```

nameplot(plt1)$
equalscale:true$
conty(zmin,zmax,n):=(lst:[],for count:0 step 1 thru n do(
lst:append(lst,[zmin+abs(zmax-zmin)/n*count]),lst)$
zmin:zmin3d$
zmax:zmax3d$
titley:"My/F="+expmy$
replot(true,contour,concat("beta ---> ",caseno," ",runname),"||| alpha",
titley),contours:conty(zmin3d,zmax3d,n)$
(doward_file(),dover_file())$
zmin:zmin3d$
zmax:zmax3d$
n:10$
titley:"Surface Contours"$
replot(true,contour,"beta --->","||| alpha",titley),contours:conty(zmin3d,zmax3d,n)$
nameplot(plt2)$
replot4(plt1,plt2),contours:conty(zmin3d,zmax3d,n)$
(doward_file(),dover_file())$
equalscale:false$

n:15$
titlez:concat("Mz/F vs (ALPHA,BETA) ",caseno," ",runname," ",phis)$
plot3d(expmz,beta,-1.57,1.57,alpha,-1.57,1.57,signz,false,titlez)$
nameplot(pltz1)$
equalscale:true$
contz(zmin,zmax,n):=(lst:[],for count:0 step 1 thru n do(
lst:append(lst,[zmin+abs(zmax-zmin)/n*count]),lst)$
zmin:zmin3d$
zmax:zmax3d$
titlez:"Mz/F="+expmz$
replot(true,contour,concat("beta ---> ",caseno," ",runname),"||| alpha",
titlez),contours:contz(zmin3d,zmax3d,n)$
(doward_file(),dover_file())$
zmin:zmin3d$
zmax:zmax3d$
n:10$
titlez:"Surface Contours"$
replot(true,contour,"beta --->","||| alpha",titlez),contours:contz(zmin3d,zmax3d,n)$
nameplot(pltz2)$
replot4(pltz1,pltz2),contours:contz(zmin3d,zmax3d,n)$
(doward_file(),dover_file())$
equalscale:false$

```

## To Plot the Intersection of the Sliding Constraint Surfaces

```

/*  intsect mec, intersects two surfaces and projects intersection curve into
    the (alpha.beta) plane  */

dynamalloc:print$

/*  data for surfaces to be intersected  */

mxyz:"MY/F"; /* which subspace is being modeled */
cases:"A & B"; /* which cases are being intersected */

aa1:4.77672637; /* coefficients */
bb1:-10.9350841; /* for surface */
cc1:0.49162517; /* one */

aa2:-0.486644814; /* coefficients */
bb2:-1.75717762; /* for surface */
cc2:0.342795227; /* two */

/*  forming two equations and intersecting  */

mf1:aa1*sin(alpha)*sin(beta)+bb1*sin(alpha)*cos(beta)+cc1*cos(alpha);
mf2:aa2*sin(alpha)*sin(beta)+bb2*sin(alpha)*cos(beta)+cc2*cos(alpha);
intersect:ev((mf1-mf2),infeval);

/*  parting equation to plot */

assume(sin(alpha)>0,cos(alpha)>0,sin(beta)>0,cos(beta)>0);

aa:if sign(part(intersect,1))=neg then -part(intersect,1,1) else part(intersect,1,1);
bb:if sign(part(intersect,2))=neg then -part(intersect,2,1,1) else part(intersect,2,1);
cc:if sign(part(intersect,3))=neg then -part(intersect,3,1,1) else part(intersect,3,1);

alpha:atan(-cc/(aa*sin(beta)+bb*cos(beta)));
root:atan(-bb/aa);
if root<-1.57 or root>1.57 then flag:out else flag:in;

dateplot:false$
noprint:true$
plotnumprec:3$
equal*scale:true$

xmin:-1.57$
xmax:1.57$
ymin:-1.57$
ymax:1.57$

/*  plot intersection curves in ALPHA vs BETA plane  */

plotmode:'[g,gr]$

if flag=out then (plot2(alpha,beta,-1.57,1.57,"BETA ---> ", ""||| ALPHA",
concat("INTERSECTION OF SLIDING CONSTRAINT SURFACES ",mxyz," ",cases))) else
(plot2(alpha,beta,-1.57,(root-.006),
"BETA ---> ", ""||| ALPHA", "",first),
plot2(alpha,beta,(root+.006),1.57,false,false,
concat("INTERSECTION OF SLIDING CONSTRAINT SURFACES ",mxyz," ",cases),same,last))$

```



(dovard\_file(),dover\_file())\$

## For Case One

```

/* kini mec, Kinematic equations and solutions for case1 (post multiplying)*/

dynamalloc:print$

batch(vector,oper)$

/* generating kinematic constraint equations */

cmat:transpose(matrix([cc11,cc12,cc13],[cc21,cc22,cc23],[cc31,cc32,cc33]));
r0:matrix([rx,ry,rz]);
i:[1,0,0];
j:[0,1,0];
k:[0,0,1];

rip:matrix([lp,ap,0]);
r2p:matrix([bp,0,0]);
r3p:matrix([0,wp-cp,0]);

p1:r1p.cmat+r0;
p2:r2p.cmat+r0;
p3:r3p.cmat+r0;
pt1:part(p1,1);
pt2:part(p2,1);
pt3:part(p3,1);

e1:dot(j,pt1)+w/2;
e2:dot(k,pt1)+l1;
e3:dot(i,pt2)+l/2;
e4:dot(j,pt2)+w/2;
e5:dot(i,pt3)+l/2;
e6:dot(k,pt3)+l1;

eq1:ev(e1);
eq2:ev(e2);
eq3:ev(e3);
eq4:ev(e4);
eq5:ev(e5);
eq6:ev(e6);

/* put equations in matrix form and solve */

solmat:matrix([cc22,0,0,0,1,0],[cc32,0,0,0,0,1],[0,cc11,0,1,0,0],
[0,cc21,0,0,1,0],[0,0,cc12,1,0,0],[0,0,cc32,0,0,1]);

rsmat:matrix([-lp*cc21-w/2],[-lp*cc31-l1],[-1/2],[-w/2],[-1/2],
[-l1]);

result:invert(solmat).rsmat;

/* evaluate expressions with rotation matrix elements */

batch(rpyrot);
kill(c);

ev(result[1,1])$
ap:trigsimp(%);

```

```

ev(result[2,1])$
bp:trigsimp(%);
ev(wp-result[3,1])$
cp:trigsimp(%);
ev(result[4,1])$
rx:trigsimp(%);
ev(result[5,1])$
ry:trigsimp(%);
ev(result[6,1])$
rz:trigsimp(%);

/* Solve for a,b,c */

r1:matrix([1/2-a],[-w/2],[-11]);
r2:matrix([-1/2],[-w/2],[b-11]);
r3:matrix([-1/2],[w/2-c],[-11]);

r0:matrix([rx],[ry],[rz]);

rip:matrix([1p],[ap],[0]);
r2p:matrix([bp],[0],[0]);
r3p:matrix([0],[wp-cp],[0]);

cmat:transpose(cmat);

ev(r1-(cmat.rip+r0));
%[1,1];
rhs(part(solve(% ,a),1));
a:trigsimp(%);

ev(r2-(cmat.r2p+r0));
%[3,1];
rhs(part(solve(% ,b),1));
b:trigsimp(%);

ev(r3-(cmat.r3p+r0));
%[2,1];
rhs(part(solve(% ,c),1));
c:trigsimp(%);

save([result1.mec,dsk,dy],ap,bp,cp,rx,ry,rz,a,b,c);

```

```

/* vell mec calculates the angular velocities in terms of linear
   velocities */

dynamalloc:print;

load(reslt1);
values;

declare(l,constant);
declare(w,constant);
declare(lp,constant);
declare(wp,constant);
declare(ll,constant);

diff(bp)$
dbp:trigsimp(%);
diff(b)$
db:trigsimp(%);
diff(cp)$
dcp:trigsimp(%);

load(facexp);

ratexpand(dbp)$
dbp:collectterms(% ,del(tx),del(ty),del(tz));

ratexpand(db)$
db:collectterms(% ,del(tx),del(ty),del(tz));

ratexpand(dcp)$
dcp:collectterms(% ,del(tx),del(ty),del(tz));

/* separate equations as coefficients of del(tx),del(ty),del(tz) */

amat:matrix([a11,a12,a13],[a21,a22,a23],[a31,a32,a33]);
ainv:invert(amat)$
linvel:matrix([del_bp],[del_b],[del_cp])$
delang:ainv.linvel$
del_tx:ratsimp(delang[1,1]);
del_ty:ratsimp(delang[2,1]);
del_tz:ratsimp(delang[3,1]);

ev(part(dbp,3)/del(tx))$
a11:trigsimp(%);
ev(part(dbp,2)/del(ty))$
a12:trigsimp(%);
ev(part(dbp,1)/del(tz))$
a13:trigsimp(%);
ev(part(db,3)/del(tx))$
a21:trigsimp(%);
ev(part(db,2)/del(ty))$
a22:trigsimp(%);
ev(part(db,1)/del(tz))$
a23:trigsimp(%);
ev(part(dcp,3)/del(tx))$
a31:trigsimp(%);
ev(part(dcp,2)/del(ty))$
a32:trigsimp(%);

```

```
ev(part(dcp,1)/del(tz))$
a33:trigsimp(%);

ev(del_tx)$
del_tx:trigsimp(%);
ev(del_ty)$
del_ty:trigsimp(%);
ev(del_tz)$
del_tz:trigsimp(%);

save([dtx1,mec,dsk,nsing],del_tx);
save([dty1,mec,dsk,nsing],del_ty);
save([dtz1,mec,dsk,nsing],del_tz);
```

```

/* nandvi mec, determines normal and contact velocity vectors for case1 */

dynamalloc:print;

batch(vector,oper);

batch(rpyrot);
cmat:c;
kill(c);

/* normals */

i:[1,0,0];
j:[0,1,0];
k:[0,0,1];
ip:transpose(matrix(i));
jp:transpose(matrix(j));
kp:transpose(matrix(k));
ip:cmat.ip;
jp:cmat.jp;
kp:cmat.kp;
ip:[ip[1,1],ip[2,1],ip[3,1]];
jp:[jp[1,1],jp[2,1],jp[3,1]];
kp:[kp[1,1],kp[2,1],kp[3,1]];

n1:cross(i,jp);
sqrt(n1[1]^2+n1[2]^2+n1[3]^2);
n1/%;
n1:transpose(matrix(%));

n2:cross(k,ip);
sqrt(n2[1]^2+n2[2]^2+n2[3]^2);
n2/%;
n2:transpose(matrix(%));

n3:cross(jp,j);
sqrt(n3[1]^2+n3[2]^2+n3[3]^2);
n3/%;
n3:transpose(matrix(%));

/* velocities */

trigsimp(diff(cmat))$
subst(deltx,del(tx),%)$
subst(delty,del(ty),%)$
del_cmat:subst(deltz,del(tz),%);

rip:matrix([ip],[ap],[0]);
r2p:matrix([bp],[0],[0]);
r3p:matrix([0],[vp-cp],[0]);

del_r0:matrix([del_rx],[del_ry],[del_rz]);

del_r1:del_cmat.rip+del_r0;
sqrt(del_r1[1,1]^2+del_r1[2,1]^2+del_r1[3,1]^2);
v1:(del_r1/%);

del_r2:del_cmat.r2p+del_r0;
sqrt(del_r2[1,1]^2+del_r2[2,1]^2+del_r2[3,1]^2);

```

```

v2:(del_r2/%);

del_r3:del_cmat.r3p+del_r0;
sqrt(del_r3[1,1]^2+del_r3[3,1]^2+del_r3[3,1]^2);
v3:(del_r3/%);

load(result1);

declare(l,constant);
declare(lp,constant);
declare(x,constant);
declare(wp,constant);
declare(ll,constant);

del_rx:trigsimp(diff(rx))$
subst(deltx,del(tx),%)$
subst(deltz,del(tz),%)$
del_rx:subst(deltz,del(tz),%);

del_ry:trigsimp(diff(ry))$
subst(deltx,del(tx),%)$
subst(deltz,del(tz),%)$
del_ry:subst(deltz,del(tz),%);

del_rz:trigsimp(diff(rz))$
subst(deltx,del(tx),%)$
subst(deltz,del(tz),%)$
del_rz:subst(deltz,del(tz),%);

kill(allbut(m1,n2,n3,v1,v2,v3,rx,ry,rz,del_rx,del_ry,del_rz,
a,b,c,ap,bp,cp));

rix:ev(1/2-a,infoval);
riy:-w/2;
riz:-ll;
r2x:-1/2;
r2y:-w/2;
r2z:ev(b-ll,infoval);
r3x:-1/2;
r3y:ev(w/2-c,infoval);
r3z:-ll;
n1x:n1[1,1];
n1y:n1[2,1];
n1z:n1[3,1];
n2x:n2[1,1];
n2y:n2[2,1];
n2z:n2[3,1];
n3x:n3[1,1];
n3y:n3[2,1];
n3z:n3[3,1];
v1x:v1[1,1];
v1y:v1[2,1];
v1z:v1[3,1];
v2x:v2[1,1];
v2y:v2[2,1];
v2z:v2[3,1];
v3x:v3[1,1];
v3y:v3[2,1];
v3z:v3[3,1];

```

```
save([result1, mec, dsk, dy], r1x, r1y, r1z, r2x, r2y, r2z, r3x, r3y, r3z, n1x, n1y, n1z, n2x,  
n2y, n2z, n3x, n3y, n3z, v1x, v1y, v1z, v2x, v2y, v2z, v3x, v3y, v3z, del_rx, del_ry, del_rz,  
ap, bp, cp, s, b, c, rx, ry, rz);
```



```

/* vals1 mec. creates and evaluates parameters for case one */

/* writefile(); */

load(result1);

caseno:"case1";

runname:"run2";

/* peg and hole parameters */

tol:0.005; /* tolerance */
l:1.25;
w:1.00;
lp:l+tol;
wp:w+tol;
ll:2.0;
u:0.5; /* friction */

/* roll-pitch-yaw angles*/

tx:-0.20915;
ty:0.14324;
tz:-0.11218;

/* the linear parameters are: */

check():=ev([a,b,c,ap,bp,cp,rx,ry,rz],infeval)$
check();

/* edge contact velocities */

del_cp:-1.0; /* defined */

del_bp:-0.5;
del_b:-0.3;

/* linear limits on edge velocities (assuming del_cp=-1) */

del_bp_min:ev(-bp/cp,infeval); /* del_bp must be more positive */
del_b_min:ev(-b/cp,infeval); /* del_b must be more positive */

/* end configuration from given values (linear assumptions) */

cp_end:0.0;
bp_end:ev(bp+del_bp*cp,infeval);
b_end:ev(b+del_b*cp,infeval);

/* closefile(); */

/* batch case1 mec */

/* batch(case1); */

```

```

/* frchbl1 mec. determines the force and moment balance equations, case1 */
dynamalloc:print;

batch(vector,oper);

r1:[r1x,r1y,r1z];
r2:[r2x,r2y,r2z];
r3:[r3x,r3y,r3z];
n1:[n1x,n1y,n1z];
n2:[n2x,n2y,n2z];
n3:[n3x,n3y,n3z];
v1:[v1x,v1y,v1z];
v2:[v2x,v2y,v2z];
v3:[v3x,v3y,v3z];

rixn1:matrix(cross(r1,n1));
r2xn2:matrix(cross(r2,n2));
r3xn3:matrix(cross(r3,n3));

rixv1:matrix(cross(r1,v1));
r2xv2:matrix(cross(r2,v2));
r3xv3:matrix(cross(r3,v3));

n1:matrix(n1);
n2:matrix(n2);
n3:matrix(n3);
v1:matrix(v1);
v2:matrix(v2);
v3:matrix(v3);

fext:matrix([fx,fy,fz]);
next:matrix([mx,my,mz]);

fbal:fext+f1*n1+f2*n2+f3*n3+u*f1*v1+u*f2*v2+u*f3*v3+del*(v1+v2+v3);

mbal:next+f1*rixn1+u*f1*rixv1+f2*r2xn2+u*f2*r2xv2+f3*r3xn3+u*f3*r3xv3
+del*(rixv1+r2xv2+r3xv3);

fbalx:fbal[1,1];
fbaly:fbal[1,2];
fbalz:fbal[1,3];

mbalx:mbal[1,1];
mbaly:mbal[1,2];
mbalz:mbal[1,3];

eliminate([fbalx,fbaly,fbalz],[f2,f3])$
solve(%f1)$
nf1:rhs(%[1]);

eliminate([fbalx,fbaly,fbalz],[f1,f3])$
solve(%f2)$
nf2:rhs(%[1]);

eliminate([fbalx,fbaly,fbalz],[f1,f2])$
solve(%f3)$
nf3:rhs(%[1]);

```

```
f1:nf1$  
f2:nf2$  
f3:nf3$  
f4:""$
```

```
save([mandf1, mec, dsk, dy], mbalx, mbaly, mbalz, f1, f2, f3, f4);
```

```

/* Case1 mec, combines and evaluates expressions for computation of jamming
and breaking contact regions */

/* Make sure that vals1 has been batched */

dynamalloc:print$

loadfile(dtx1.mec,dsk,nsing);
deltx:ev(del_tx,infeval)$

loadfile(dty1.mec,dsk,nsing);
deltty:ev(del_ty,infeval)$

loadfile(dtz1.mec,dsk,nsing);
deltz:ev(del_tz,infeval)$

loadfile(mandf1.mec,dsk,dy);

del_rx:ev(del_rx,infeval)$
del_ry:ev(del_ry,infeval)$
del_rz:ev(del_rz,infeval)$
rix:ev(rix)$
riy:ev(riy)$
riz:ev(riz)$
r2x:ev(r2x)$
r2y:ev(r2y)$
r2z:ev(r2z)$
r3x:ev(r3x)$
r3y:ev(r3y)$
r3z:ev(r3z)$
nix:ev(nix)$
niy:ev(niy)$
niz:ev(niz)$
n2x:ev(n2x)$
n2y:ev(n2y)$
n2z:ev(n2z)$
n3x:ev(n3x)$
n3y:ev(n3y)$
n3z:ev(n3z)$
vix:ev(vix,infeval)$
viy:ev(viy,infeval)$
viz:ev(viz,infeval)$
v2x:ev(v2x,infeval)$
v2y:ev(v2y,infeval)$
v2z:ev(v2z,infeval)$
v3x:ev(v3x,infeval)$
v3y:ev(v3y,infeval)$
v3z:ev(v3z,infeval)$

ratprint:false$
keepfloat:true$
expop:9$
expon:9$
ratexpand:false$
numer:true$

f1:ratexpand(ev(f1,infeval))$
f2:ratexpand(ev(f2,infeval))$
f3:ratexpand(ev(f3,infeval))$

```

```

mbalx:ratexpand(ev(mbalx,ineval))$
mbaly:ratexpand(ev(mbaly,ineval))$
mbalz:ratexpand(ev(mbalz,ineval))$

solve(mbalx,del)$
expand(rhs(%[1]))$
expax:ev(%,float);

solve(mbaly,del)$
expand(rhs(%[1]))$
expay:ev(%,float);

solve(mbalz,del)$
expand(rhs(%[1]))$
expaz:ev(%,float);

/* Determining jamming and non-jamming sides of surfaces */

assume(mx>0,my>0,mz>0)$ /* this assumption is not necessary */

signx:sign(part(expax,1))$
termx:if signx=neg then -part(expax,1,1,1) else part(expax,1,1,1)$
signx:if signx=neg then "Mx/f is less than (below surface)" else "Mx/f is greater than
(above surface)";
expmx:expand(-(expax-part(expax,1))/termx);

signy:sign(part(expay,1))$
termy:if signy=neg then -part(expay,1,1,1) else part(expay,1,1,1)$
signy:if signy=neg then "My/f is less than (below surface)" else "My/f is greater than
(above surface)";
expy:expand(-(expay-part(expay,1))/termy);

signz:sign(part(expaz,1))$
termz:if signz=neg then -part(expaz,1,1,1) else part(expaz,1,1,1)$
signz:if signz=neg then "Mz/f is less than (below surface)" else "Mz/f is greater than
(above surface)";
expmz:expand(-(expaz-part(expaz,1))/termz);

del:0.0$

ev(f1)$
f1:expand(%)$
f1+"0=";

ev(f2)$
f2:expand(%)$
f2+"0=";

ev(f3)$
f3:expand(%)$
f3+"0=";

/* Determining breaking contact and non-breaking contact sides of curves */

assume(fx>0,fy>0,fz>0)$ /* this assumption is not necessary */

```

```

c1:if sign(part(f1,1))=neg then -part(f1,1,1,1) else part(f1,1,1)$
b1:if sign(part(f1,2))=neg then -part(f1,2,1,1) else part(f1,2,1)$
a1:if sign(part(f1,3))=neg then -part(f1,3,1,1) else part(f1,3,1)$
alpha1:atan(c1/(a1*cos(beta)+b1*sin(beta)));
root1:atan(-a1/b1)$
if root1<-1.57 or root1>1.57 then flag1:out else flag1:in;
a1*cos(root1+.01)+b1*sin(root1+.01)$
sign1:if sign(%)=neg then "alpha < for Beta > "-root1 else "alpha > for Beta > "+root1;

c2:if sign(part(f2,1))=neg then -part(f2,1,1,1) else part(f2,1,1)$
b2:if sign(part(f2,2))=neg then -part(f2,2,1,1) else part(f2,2,1)$
a2:if sign(part(f2,3))=neg then -part(f2,3,1,1) else part(f2,3,1)$
alpha2:atan(c2/(a2*cos(beta)+b2*sin(beta)));
root2:atan(-a2/b2)$
if root2<-1.57 or root2>1.57 then flag2:out else flag2:in;
a2*cos(root2+.01)+b2*sin(root2+.01)$
sign2:if sign(%)=neg then "alpha < for Beta > "+root2 else "alpha > for Beta > "+root2;

c3:if sign(part(f3,1))=neg then -part(f3,1,1,1) else part(f3,1,1)$
b3:if sign(part(f3,2))=neg then -part(f3,2,1,1) else part(f3,2,1)$
a3:if sign(part(f3,3))=neg then -part(f3,3,1,1) else part(f3,3,1)$
alpha3:atan(c3/(a3*cos(beta)+b3*sin(beta)));
root3:atan(-a3/b3)$
if root3<-1.57 or root3>1.57 then flag3:out else flag3:in;
a3*cos(root3+.01)+b3*sin(root3+.01)$
sign3:if sign(%)=neg then "alpha < for Beta > "+root3 else "alpha > for Beta > "+root3;

fx:sin(alpha)*cos(beta)$
fy:sin(alpha)*sin(beta)$
fz:-cos(alpha)$

expmx:expand(ev(expmx))$
"Mx/F="+expmx;
expmy:expand(ev(expmy))$
"My/F="+expmy;
expmz:expand(ev(expmz))$
"Mz/F="+expmz;

save([csival,nec,dsk,dy],caseno,runname,signx,signy,signz,expmx,expmy,expmz,
alpha1,root1,flag1,sign1,alpha2,root2,flag2,sign2,alpha3,root3,flag3,sign3,f4);

```

## For Case Two

```

/* kin2 mec, kinematic equations and solutions for case2 (post multiplying).
   bottom edge in corner */

dynamalloc:print:

batch(vector,oper):

cmat:transpose(matrix([cc11,cc12,cc13],[cc21,cc22,cc23],[cc31,cc32,cc33]));
r0:matrix([rx,ry,rz]);
i:[1,0,0];
j:[0,1,0];
k:[0,0,1];

rip:matrix([lp,ap,0]);
r2p:matrix([bp,0,0]);
r3p:matrix([0,wp,0]); /* note cp is zero */

p1:rip.cmat+r0;
p2:r2p.cmat+r0;
p3:r3p.cmat+r0;
pt1:part(p1,1);
pt2:part(p2,1);
pt3:part(p3,1);

e1:dot(j,pt1)+w/2;
e2:dot(k,pt1)+l1;
e3:dot(i,pt2)+l/2; /* redundant equation */
e4:dot(j,pt2)+w/2;
e5:dot(i,pt3)+l/2;
e6:dot(k,pt3)+l1;

eq1:ev(e1);
eq2:ev(e2);
eq3:ev(e3);
eq4:ev(e4);
eq5:ev(e5);
eq6:ev(e6);

constraint:ev(eq3-eq6); /* specifies redundancy of eq3 */

/* inverting equations, without third equation containing constraint on
   RP7 angles */

solmat:matrix([cc22,0,0,1,0],[cc32,0,0,0,1],[0,cc21,0,1,0],[0,0,1,0,0],
[0,0,0,0,1]);
rhsmat:matrix([-lp*cc21-w/2],[-l1-lp*cc31],[-w/2],[-l/2-wp*cc12],
[-l1-wp*cc32]);

result:ratsimp(invert(solmat).rhsmat);

/* evaluate expressions with rotation matrix elements */

batch(rpyrot);
kill(c);

ev(result[1,1])$

```

```

ap:trigsimp(%);
ev(result[2,1])$
bp:trigsimp(%);
cp:0;
ev(result[3,1])$
rx:trigsimp(%);
ev(result[4,1])$
ry:trigsimp(%);
ev(result[5,1])$
rz:trigsimp(%);

/* Solve for a,b,c */

r1:matrix([1/2-a],[-w/2],[-11]);
r2:matrix([-1/2],[-w/2],[b-11]);
r3:matrix([-1/2],[w/2-c],[-11]);

r0:matrix([rx],[ry],[rz]);

rip:matrix([1p],[ap],[0]);
r2p:matrix([bp],[0],[0]);
r3p:matrix([0],[wp],[0]);

cmat:transpose(cmat);

ev(r1-(cmat.r1p+r0));
%[1,1];
rhs(part(solve(% ,a),1));
a:trigsimp(%);

ev(r2-(cmat.r2p+r0));
%[3,1];
rhs(part(solve(% ,b),1));
b:trigsimp(%);

ev(r3-(cmat.r3p+r0));
%[2,1];
rhs(part(solve(% ,c),1));
c:trigsimp(%);

constraint:trigsimp(ev(constraint,infeval));

save([reslt2,mec,dsk,dy],constraint,ap,bp,cp,rx,ry,rz,a,b,c);

```



```

/* vel2 mec calculates the angular velocities in terms of linear
   velocities for case2 */

dynamalloc:print;

load(result2);
values;

declare(l,constant);
declare(w,constant);
declare(lp,constant);
declare(wp,constant);
declare(ll,constant);

diff(b)$
db:trigsimp($);
diff(bp)$
dbp:trigsimp($);

del_const:diff(constraint)$

load(facexp);

ratexpand(del_const)$
del_tz:rhs(part(solve(% ,del(tz)),1));

subst(del_tz,del(tz),db)$
ratexpand($)$
db:collectterms(% ,del(tx),del(ty));

subst(del_tz,del(tz),dbp)$
ratexpand($)$
dbp:collectterms(% ,del(tx),del(ty));

/* separate equations as coefficients of del(tx),del(ty) */

amat:matrix([a11,a12],[a21,a22])$
ainv:invert(amat)$
linvel:matrix([del_b],[del_bp])$
delang:ainv.linvel$
del_tx:ratsimp(delang[1,1])$
del_ty:ratsimp(delang[2,1])$
ev(part(db,2)/del(tx))$
a11:trigsimp($)$
ev(part(db,1)/del(ty))$
a12:trigsimp($)$
ev(part(dbp,2)/del(tx))$
a21:trigsimp($)$
ev(part(dbp,1)/del(ty))$
a22:trigsimp($)$

del_tx:ev(del_tx,infeval)$

del_ty:ev(del_ty,infeval)$

subst(del_tx,del(tx),del_tz)$
subst(del_ty,del(ty),%)$
del_tz:ev(% ,infeval)$

```

```
save([dtx2,mec,dsk,nsing],del_tx);  
save([dty2,mec,dsk,nsing],del_ty);  
save([dtz2,mec,dsk,nsing],del_tz);
```

```

/* nandv2 mec. determines normal and contact velocity vectors for case2 */

dynamalloc:print;

batch(vector,oper);

batch(rpyrot);
cmat:c;
kill(c);

/* normals */

i:[1,0,0];
j:[0,1,0];
k:[0,0,1];
ip:transpose(matrix(i));
jp:transpose(matrix(j));
kp:transpose(matrix(k));
ip:cmat.ip;
jp:cmat.jp;
kp:cmat.kp;
ip:[ip[1,1],ip[2,1],ip[3,1]];
jp:[jp[1,1],jp[2,1],jp[3,1]];
kp:[kp[1,1],kp[2,1],kp[3,1]];

n1:cross(i,jp);
sqrt(n1[1]^2+n1[2]^2+n1[3]^2);
n1/%;
n1:transpose(matrix(%));

n2:cross(k,ip);
sqrt(n2[1]^2+n2[2]^2+n2[3]^2);
n2/%;
n2:transpose(matrix(%));

n3a:cross(ip,j);
sqrt(n3a[1]^2+n3a[2]^2+n3a[3]^2);
n3a/%;
n3a:transpose(matrix(%));

n3c:cross(jp,j);
sqrt(n3c[1]^2+n3c[2]^2+n3c[3]^2);
n3c/%;
n3c:transpose(matrix(%));

/* velocities */

trigsimp(diff(cmat))$
subst(deltx,del(tx),%)$
subst(delty,del(ty),%)$
del_cmat:subst(del_tz,del(tz),%);

rip:matrix([ip],[ap],[0]);
r2p:matrix([bp],[0],[0]);
r3p:matrix([0],[wp],[0]);

del_r0:matrix([del_rx],[del_ry],[del_rz]);

del_r1:del_cmat.rip+del_r0;

```

```

sqrt(del_r1[1,1]^2+del_r1[2,1]^2+del_r1[3,1]^2);
v1:(del_r1/%)

del_r2:del_cmat.r2p+del_r0;
sqrt(del_r2[1,1]^2+del_r2[2,1]^2+del_r2[3,1]^2);
v2:(del_r2/%)

del_r3:del_cmat.r3p+del_r0;
sqrt(del_r3[1,1]^2+del_r3[2,1]^2+del_r3[3,1]^2);
v3:(del_r3/%)

load(:solt2);

declare(l,constant);
declare(lp,constant);
declare(w,constant);
declare(wp,constant);
declare(ll,constant);

del_rx:trigsimp(diff(rx))$
subst(deltx,del(tx),%)$
subst(delty,del(ty),%)$
del_rx:subst(del_tz,del(tz),%);

del_ry:trigsimp(diff(ry))$
subst(deltx,del(tx),%)$
subst(delty,del(ty),%)$
del_ry:subst(del_tz,del(tz),%);

del_rz:trigsimp(diff(rz))$
subst(deltx,del(tx),%)$
subst(delty,del(ty),%)$
del_rz:subst(del_tz,del(tz),%);

kill(allbut(n1,n2,n3s,n3c,v1,v2,v3,rx,ry,rz,del_rx,del_ry,del_rz,
a,b,c,ap,bp,cp,constraint));

rix:ev(1/2-a,infeval);
riy:-w/2;
riz:-ll;
r2x:-1/2;
r2y:-w/2;
r2z:ev(b-ll,infeval);
r3x:-1/2;
r3y:ev(w/2-c,infeval);
r3z:-ll;
nix:n1[1,1];
niy:n1[2,1];
niz:n1[3,1];
n2x:n2[1,1];
n2y:n2[2,1];
n2z:n2[3,1];
n3ax:n3s[1,1];
n3ay:n3s[2,1];
n3sz:n3s[3,1];
n3cx:n3c[1,1];
n3cy:n3c[2,1];
n3cz:n3c[3,1];
vix:v1[1,1];
viy:v1[2,1];

```

```
v1z:v1[3,1];
v2x:v2[1,1];
v2y:v2[2,1];
v2z:v2[3,1];
v3x:v3[1,1];
v3y:v3[2,1];
v3z:v3[3,1];
```

```
save([result2,mec,dsk,dy],rix,riy,r1z,r2x,r2y,r2z,r3x,r3y,r3z,n1x,n1y,n1z,n2x,
n2y,n2z,n3sx,n3sy,n3sz,n3cx,n3cy,n3cz,v1x,v1y,v1z,v2x,v2y,v2z,v3x,v3y,v3z,
del_rx,del_ry,del_rz,ap,bp,cp,e,b,c,rx,ry,rz,constraint);
```

```

/* vals2 mec, creates and checks values for case two */

/* writefile(); */

load(result2);

caseno:"case2";

runname:"run4";

/* peg and hole parameters */

tol:0.005; /* tolerance */
l:1.25;
w:1.00;
lp:l+tol;
wp:w+tol;
ll:2.0;
u:0.5; /* friction */

/* roll-pitch-yaw angles*/

tx:-0.20915;
tz:-0.11218;

/* orientation of normal for f3 */
phi3:%pi/4; /* 0 <= phi3 <= %PI/2 */

/* ty is determined from tx,tz */

load(ins1);

/* the linear parameters are */

i(ty):='constraint;

zsolveeps:1.0e-4;
zsolvensig:4;

check():=(sol:zsolve([f],[-1.0]),
ty:part(sol,2,1),ev([a,b,c,ap,bp,cp,rx,ry,rz],infeval));
check();
tx;
ty;
tz;

/* edge contact velocities */

del_b:-1.0;
del_bp:-1.0; /* defined */

/* linear limits on edge velocities (assuming del_bp=-1) */
del_b_min:ev(-b/bp,infeval); /* del_b must be more positive */

```

```
/* end state based on given values (linear assumptions) */  
  
bp_end:0.0;  
b_end:ev(b+del_b*bp,ineval);  
  
/* closefile(); */  
  
/* batch case2 */  
  
/* batch(case2); */
```

```

/* frch12 mec, determines the force and moment balance equations, case2
   the fourth contact parameter is the angle phi3, bottom edge in corner */

dynamalloc:print;

batch(vector,oper);

r1:[r1x,r1y,r1z];
r2:[r2x,r2y,r2z];
r3:[r3x,r3y,r3z];
n1:[n1x,n1y,n1z];
n2:[n2x,n2y,n2z];
n3s:[n3sx,n3sy,n3sz]; /* two components */
n3c:[n3cx,n3cy,n3cz]; /* of normal 3 */
v1:[v1x,v1y,v1z];
v2:[v2x,v2y,v2z];
v3:[v3x,v3y,v3z];

r1xn1:matrix(cross(r1,n1));
r2xn2:matrix(cross(r2,n2));
r3xn3s:matrix(cross(r3,n3s));
r3xn3c:matrix(cross(r3,n3c));

r1xv1:matrix(cross(r1,v1));
r2xv2:matrix(cross(r2,v2));
r3xv3:matrix(cross(r3,v3));

n1:matrix(n1);
n2:matrix(n2);
n3s:matrix(n3s);
n3c:matrix(n3c);
v1:matrix(v1);
v2:matrix(v2);
v3:matrix(v3);

fext:matrix([fx,fy,fz]);
mext:matrix([mx,my,mz]);

fbal:fext+f1*n1+f2*n2+f3*(n3s*sin(phi3)+n3c*cos(phi3))+u*f1*v1+u*f2*v2+u*f3*v3
+del*(v1+v2+v3);

mbal:mext+f1*r1xn1+u*f1*r1xv1+f2*r2xn2+u*f2*r2xv2
+f3*(r3xn3s*sin(phi3)+r3xn3c*cos(phi3))+u*f3*r3xv3+del*(r1xv1+r2xv2+r3xv3);

fbalx:fbal[1,1];
fbaly:fbal[1,2];
fbalz:fbal[1,3];

mbalx:mbal[1,1];
mbaly:mbal[1,2];
mbalz:mbal[1,3];

eliminate([fbalx,fbaly,fbalz],[f2,f3])$
solve(%,f1)$
nf1:rhs(%[1]);

eliminate([fbalx,fbaly,fbalz],[f1,f3])$
solve(%,f2)$
nf2:rhs(%[1]);

```



```
eliminate([fbalx,fbaly,fbalz],[f1,f2])$  
solve(%,f3)$  
nf3:rhs(%[1]);  
  
f1:nf1$  
f2:nf2$  
f3:nf3$  
f4:""$  
  
save([mandf2,mec,dsk,dy],mbalx,mbaly,mbalz,f1,f2,f3,f4);
```

```

/* Case2 mec, combines and evaluates parameters for the jamming and
   breaking contact equations */

/* Make sure that vals2 has been batched and values chosen for tx,ty,tz */

dynamalloc:print$

loadfile(dtx2,mec,dsk,nsing);
del_tx:ev(del_tx,ineval)$

loadfile(dty2,mec,dsk,nsing);
del_ty:ev(del_ty,ineval)$

loadfile(dtz2,mec,dsk,nsing);
del_tz:ev(del_tz,ineval)$

loadfile(mandf2,mec,dsk,dy);

del_rx:ev(del_rx,ineval)$
del_ry:ev(del_ry,ineval)$
del_rz:ev(del_rz,ineval)$
rix:ev(rix)$
riy:ev(riy)$
riz:ev(riz)$
r2x:ev(r2x)$
r2y:ev(r2y)$
r2z:ev(r2z)$
r3x:ev(r3x)$
r3y:ev(r3y)$
r3z:ev(r3z)$
nix:ev(nix)$
niy:ev(niy)$
niz:ev(niz)$
n2x:ev(n2x)$
n2y:ev(n2y)$
n2z:ev(n2z)$
n3sx:ev(n3sx)$
n3sy:ev(n3sy)$
n3sz:ev(n3sz)$
n3cx:ev(n3cx)$
n3cy:ev(n3cy)$
n3cz:ev(n3cz)$
vix:ev(vix,ineval)$
viy:ev(viy,ineval)$
viz:ev(viz,ineval)$
v2x:ev(v2x,ineval)$
v2y:ev(v2y,ineval)$
v2z:ev(v2z,ineval)$
v3x:ev(v3x,ineval); /* should be 0 */
v3y:ev(v3y,ineval); /* should be +-1 */
v3z:ev(v3z,ineval); /* should be 0 */

ratprint:false$
keepfloat:true$
expop:9$
expon:9$
ratexpand:false$
numer:true$

```

```

f1:ratexpand(ev(f1,infeval))$
f2:ratexpand(ev(f2,infeval))$
f3:ratexpand(ev(f3,infeval))$

mbalx:ratexpand(ev(mbalx,infeval))$
mbaly:ratexpand(ev(mbaly,infeval))$
mbalz:ratexpand(ev(mbalz,infeval))$

solve(mbalx,del)$
expand(rhs(%[1]))$
expmx:ev(%,float);

solve(mbaly,del)$
expand(rhs(%[1]))$
expmy:ev(%,float);

solve(mbalz,del)$
expand(rhs(%[1]))$
expmz:ev(%,float);

assume(mx>0,my>0,mz>0)$

signx:sign(part(expmx,1))$
termx:if signx=neg then -part(expmx,1,1,1) else part(expmx,1,1)$
signx:if signx=neg then "Mx/f is less than (below surface)" else "Mx/f is greater than
(above surface)";
expmx:expand(-(expmx-part(expmx,1))/termx);

signy:sign(part(expmy,1))$
termy:if signy=neg then -part(expmy,1,1,1) else part(expmy,1,1)$
signy:if signy=neg then "My/f is less than (below surface)" else "My/f is greater than
(above surface)";
expmy:expand(-(expmy-part(expmy,1))/termy);

signz:sign(part(expmz,1))$
termz:if signz=neg then -part(expmz,1,1,1) else part(expmz,1,1)$
signz:if signz=neg then "Mz/f is less than (below surface)" else "Mz/f is greater than
(above surface)";
expmz:expand(-(expmz-part(expmz,1))/termz);

del:0.0$

ev(f1)$
f1:expand(%)$
f1+"0=";

ev(f2)$
f2:expand(%)$
f2+"0=";

ev(f3)$
f3:expand(%)$
f3+"0=";

assume(fx>0,fy>0,fz>0)$

c1:if sign(part(f1,1))=neg then -part(f1,1,1,1) else part(f1,1,1)$
b1:if sign(part(f1,2))=neg then -part(f1,2,1,1) else part(f1,2,1)$
a1:if sign(part(f1,3))=neg then -part(f1,3,1,1) else part(f1,3,1)$
alpha1:atan(c1/(a1*cos(beta)+b1*sin(beta)));

```

```

root1:atan(-a1/b1)$
if root1<-1.57 or root1>1.57 then flag1:out else flag1:in;
a1*cos(root1+.01)+b1*sin(root1+.01)$
sign1:if sign(%)=neg then "alpha < for Beta > "+root1 else "alpha > for Beta > "+root1;

c2:if sign(part(f2,1))=neg then -part(f2,1,1,1) else part(f2,1,1)$
b2:if sign(part(f2,2))=neg then -part(f2,2,1,1) else part(f2,2,1)$
a2:if sign(part(f2,3))=neg then -part(f2,3,1,1) else part(f2,3,1)$
alpha2:atan(c2/(a2*cos(beta)+b2*sin(beta)));
root2:atan(-a2/b2)$
if root2<-1.57 or root2>1.57 then flag2:out else flag2:in;
a2*cos(root2+.01)+b2*sin(root2+.01)$
sign2:if sign(%)=neg then "alpha < for Beta > "+root2 else "alpha > for Beta > "+root2;

c3:if sign(part(f3,1))=neg then -part(f3,1,1,1) else part(f3,1,1)$
b3:if sign(part(f3,2))=neg then -part(f3,2,1,1) else part(f3,2,1)$
a3:if sign(part(f3,3))=neg then -part(f3,3,1,1) else part(f3,3,1)$
alpha3:atan(c3/(a3*cos(beta)+b3*sin(beta)));
root3:atan(-a3/b3)$
if root3<-1.57 or root3>1.57 then flag3:out else flag3:in;
a3*cos(root3+.01)+b3*sin(root3+.01)$
sign3:if sign(%)=neg then "alpha < for Beta > "+root3 else "alpha > for Beta > "+root3;

fx:sin(alpha)*cos(beta)$
fy:sin(alpha)*sin(beta)$
fz:-cos(alpha)$

expmx:expand(ev(expmx))$
"Mx/F="+expmx;
expmy:expand(ev(expmy))$
"My/F="+expmy;
expmz:expand(ev(expmz))$
"Mz/F="+expmz;

save([cs2val,mec,dk,dy],caseno,runname,phi3,signx,signy,signz,expmx,expmy,
expaz,alpha1,root1,flag1,sign1,alpha2,root2,flag2,sign2,alpha3,root3,flag3,
sign3,f4);

```

## For Case Three

```

/* kin3a mec, kinematic equations and solutions for case3a (post multiplying)*/
dynamalloc:print;

batch(vector,oper);

cmat:transpose(matrix([cc11,cc12,cc13],[cc21,cc22,cc23],[cc31,cc32,cc33]));
r0:matrix([rx,ry,rz]);
i:[1,0,0];
j:[0,1,0];
k:[0,0,1];

rip:matrix([lp,ap,0]);
r2p:matrix([0,0,0]); /* note bp is zero */
r3p:matrix([0,wp,0]); /* note cp is zero */

p1:r1p.cmat+r0;
p2:r2p.cmat+r0;
p3:r3p.cmat+r0;
pt1:part(p1,1);
pt2:part(p2,1);
pt3:part(p3,1);

e1:dot(j,pt1)+w/2; /* redundant with e4 */
e2:dot(k,pt1)+l1;
e3:dot(i,pt2)+l/2;
e4:dot(j,pt2)+w/2;
e5:dot(i,pt3)+l/2; /* redundant with e3 */
e6:dot(k,pt3)+l1;

eq1:ev(e1);
eq2:ev(e2);
eq3:ev(e3);
eq4:ev(e4);
eq5:ev(e5);
eq6:ev(e6);

constraint1:ev(eq3-eq5);

constraint2:ev(eq1-eq4);

/* inverting equations, without first and fifth equations containing
constraints on RPY angles */

solmat:matrix([cc32,0,0,1],[0,1,0,0],[0,0,1,0],[0,0,0,1]);
rsmat:matrix([-lp*cc31-l1],[-l/2],[-w/2],[-l1-wp*cc32]);

result:ratsimp(invert(solmat).rsmat);

/* evaluate expressions with rotation matrix elements */

batch(rpyrot);
kill(c);

ev(result[1,1])$
ap:trigsimp(%);

```

```

bp:0;
cp:0;
ev(result[2,1])$
rx:trigsimp(%);
ev(result[3,1])$
ry:trigsimp(%);
ev(result[4,1])$
rz:trigsimp(%);

/* Solve for a,b,c */

r1:matrix([1/2-a],[-w/2],[-11]);
r2:matrix([-1/2],[-w/2],[b-11]);
r3:matrix([-1/2],[w/2-c],[-11]);

r0:matrix([rx],[ry],[rz]);

rip:matrix([1p],[ap],[0]);
r2p:matrix([0],[0],[0]);
r3p:matrix([0],[wp],[0]);

cmat:transpose(cmat);

ev(r1-(cmat.rip+r0));
%[1,1];
rhs(part(solve(%a),1));
a:trigsimp(%);

ev(r2-(cmat.r2p+r0));
%[3,1];
rhs(part(solve(%b),1));
b:trigsimp(%);

ev(r3-(cmat.r3p+r0));
%[2,1];
rhs(part(solve(%c),1));
c:trigsimp(%);

constraint1:trigsimp(ev(constraint1,ifeval));
constraint2:trigsimp(ev(constraint2,ifeval));

save([res13a,mec,dsk,dy],constraint1,constraint2,ap,bp,cp,rx,ry,rz,a,b,c);

```

```

/* vel3a.mec calculates the angular velocity in terms of the linear
   velocity for case3a (one d.o.f.) */

dynamalloc:print;

load(res13a);
values;

declare(l,constant);
declare(w,constant);
declare(lp,constant);
declare(wp,constant);
declare(ll,constant);

??(b)$
db:trigsimp(%);

del_const1:diff(constraint1)$
del_const2:diff(constraint2)$

load(facexp);

ratexpand(del_const1)$
del_tz:rhs(part(solve(%,del(tz)),1));

subst(del_tz,del(tz),del_const2)$
ratexpand(%)$
del_ty:rhs(part(solve(%,del(ty)),1));

subst(del_ty,del(ty),del_const1)$
ratexpand(%)$
del_tz:rhs(part(solve(%,del(tz)),1));

subst(del_tz,del(tz),db)$
subst(del_ty,del(ty),%)$
ratexpand(%)$
db:collectterms(%,del(tx));

/* separate equation into a coefficient of del(tx) */

amat:a11;
ainv:1/a11;
linvel:del_b;
delang:ainv*linvel$
del_tx:ratsimp(delang);

ev(db/del(tx))$
a11:trigsimp(%);

ev(del_tx,infeval)$
del_tx:trigsimp(%);

ev(subst(del_tx,del(tx),del_ty),infeval);
del_ty:trigsimp(%);

ev(subst(del_tx,del(tx),del_tz),infeval);
del_tz:trigsimp(%);

save([dtx3a,mec,dsk,nsing],del_tx);

```

```
save([dty3a,mec,dsk,nsing],del_ty);  
save([dtz3a,mec,dsk,nsing],del_tz);
```



```

/* nanv3a mec, determines normal and contact velocity vectors for case3a */

dynamalloc:print;

batch(vector,oper);

batch(rpyrot);
cmat:c;
kill(c);

/* normals */

i:[1,0,0];
j:[0,1,0];
k:[0,0,1];
ip:transpose(matrix(i));
jp:transpose(matrix(j));
kp:transpose(matrix(k));
ip:cmat.ip;
jp:cmat.jp;
kp:cmat.kp;
ip:[ip[1,1],ip[2,1],ip[3,1]];
jp:[jp[1,1],jp[2,1],jp[3,1]];
kp:[kp[1,1],kp[2,1],kp[3,1]];

n1:cross(i,jp);
sqrt(n1[1]^2+n1[2]^2+n1[3]^2);
n1/%;
n1:transpose(matrix(%));

n2s:cross(k,ip);
sqrt(n2s[1]^2+n2s[2]^2+n2s[3]^2);
n2s/%;
n2s:transpose(matrix(%));

n2c:cross(jp,k);
sqrt(n2c[1]^2+n2c[2]^2+n2c[3]^2);
n2c/%;
n2c:transpose(matrix(%));

n3s:cross(ip,j);
sqrt(n3s[1]^2+n3s[2]^2+n3s[3]^2);
n3s/%;
n3s:transpose(matrix(%));

n3c:cross(jp,j);
sqrt(n3c[1]^2+n3c[2]^2+n3c[3]^2);
n3c/%;
n3c:transpose(matrix(%));

/* velocities */

trigsimp(diff(cmat))$
subst(deltx,del(tx),%)$
subst(del_ty,del(ty),%)$
del_cmat:subst(del_tz,del(tz),%);

rip:matrix([1p],[ap],[0]);
r2p:matrix([0],[0],[0]);

```

```

r3p:matrix([0],[wp],[0]);

del_r0:matrix([del_rx],[del_ry],[del_rz]);

del_r1:del_cmat.r1p+del_r0;
sqrt(del_r1[1,1]^2+del_r1[2,1]^2+del_r1[3,1]^2);
v1:(del_r1/%)

del_r2:del_cmat.r2p+del_r0;
sqrt(del_r2[1,1]^2+del_r2[2,1]^2+del_r2[3,1]^2);
v2:(del_r2/%)

del_r3:del_cmat.r3p+del_r0;
sqrt(del_r3[1,1]^2+del_r3[2,1]^2+del_r3[3,1]^2);
v3:(del_r3/%)

load(res13a);

declare(l,constant);
declare(lp,constant);
declare(w,constant);
declare(wp,constant);
declare(ll,constant);

del_rx:trigsimp(diff(rx))$
subst(deltx,del(tx),%)$
subst(del_ty,del(ty),%)$
del_rx:subst(del_tz,del(tz),%);

del_ry:trigsimp(diff(ry))$
subst(deltx,del(tx),%)$
subst(del_ty,del(ty),%)$
del_ry:subst(del_tz,del(tz),%);

del_rz:trigsimp(diff(rz))$
subst(deltx,del(tx),%)$
subst(del_ty,del(ty),%)$
del_rz:subst(del_tz,del(tz),%);

kill(allbut(n1,n2s,n2c,n3s,n3c,v1,v2,v3,rx,ry,rz,del_rx,del_ry,del_rz,
a,b,c,ap,bp,cp,constraint1,constraint2));

rix:ev(1/2-a,ineval);
riy:-w/2;
riz:-ll;
r2x:-1/2;
r2y:-w/2;
r2z:ev(b-ll,ineval);
r3x:-1/2;
r3y:ev(w/2-c,ineval);
r3z:-ll;
nix:n1[1,1];
niy:n1[2,1];
niz:n1[3,1];
n2sx:n2s[1,1];
n2sy:n2s[2,1];
n2sz:n2s[3,1];
n2cx:n2c[1,1];
n2cy:n2c[2,1];
n2cz:n2c[3,1];

```

```
n3sx:n3s[1,1];
n3sy:n3s[2,1];
n3sz:n3s[3,1];
n3cx:n3c[1,1];
n3cy:n3c[2,1];
n3cz:n3c[3,1];
v1x:v1[1,1];
v1y:v1[2,1];
v1z:v1[3,1];
v2x:v2[1,1];
v2y:v2[2,1];
v2z:v2[3,1];
v3x:v3[1,1];
v3y:v3[2,1];
v3z:v3[3,1];

save([res13a,mec,dsk,dy],rix,r1y,r1z,r2x,r2y,r2z,r3x,r3y,r3z,n1x,n1y,n1z,n2sx,
n2sy,n2sz,n2cx,n2cy,n2cz,n3sx,n3sy,n3sz,n3cx,n3cy,n3cz,v1x,v1y,v1z,v2x,v2y,v2z,
v3x,v3y,v3z,del_rx,del_ry,del_rz,ap,bp,cp,a,b,c,rx,ry,rz,
constraint1,constraint2);
```

```

/* vals3a mec, creates and checks values for case three A */

/* writefile(): */

load(res13a);

caseno:"case3A";

runname:"run5";

/* solve for ty */

solve(ev(constraint1),ty);
evty:rhs(part(ev(%,ineval),1));
ty:evty;
constraint2:ev(constraint2,ineval);
kill(ty);

/* peg and hole parameters */

tol:0.005; /* tolerance */
l:1.25;
w:1.00;
lp:l+tol;
wp:w+tol;
ll:2.0;
u:0.5; /* friction */

/* roll-pitch-yaw angles*/

tx:-0.1549i;

/* orientation of normals for f2 and f3 */

phi2:%pi/2; /* 0 <= phi2 <= %pi/2 */
phi3:0.0; /* 0 <= phi3 <= %pi/2 */

/* tz is determined from tx and ty */

load(ims1);

f(tz):='constraint2;

zsolveeps:1.0e-4;
zsolvensig:4;

/* the linear parameters are: */

check():=(kill(tz,ty),ty:evty,soltz:zsolve([f],[-0.05]),tz:part(soltz,2,1),
ty:ev(ty),ev([a,b,c,ap,bp,cp,rx,ry,rz],ineval));
check();
tx;
ty;
tz;

/* edge contact velocity */

del_b:-1.0; /* binary choice (1 d.o.f.) */

```

```
/* end state based on given values (linear assumptions) */  
c_end:0.0; /* trivial */  
/* closefile(); */  
/* batch case3a mec */  
/* batch(case3a); */
```

```

/* fcbl3a mec, determines the force and moment balance equations, case3a
   the fourth parameter is phi2, the fifth parameter is phi3, side edge
   in corner, bottom edge in corner */

dynamalloc:print;

batch(vector,oper);

r1:[r1x,r1y,r1z];
r2:[r2x,r2y,r2z];
r3:[r3x,r3y,r3z];
r1:[n1x,n1y,n1z];
n2s:[n2sx,n2sy,n2sz]; /* two components */
n2c:[n2cx,n2cy,n2cz]; /* of normal 2 */
n3s:[n3sx,n3sy,n3sz]; /* two components */
n3c:[n3cx,n3cy,n3cz]; /* of normal 3 */
v1:[v1x,v1y,v1z];
v2:[v2x,v2y,v2z];
v3:[v3x,v3y,v3z];

r1xn1:matrix(cross(r1,n1));
r2xn2s:matrix(cross(r2,n2s));
r2xn2c:matrix(cross(r2,n2c));
r3xn3s:matrix(cross(r3,n3s));
r3xn3c:matrix(cross(r3,n3c));

r1xv1:matrix(cross(r1,v1));
r2xv2:matrix(cross(r2,v2));
r3xv3:matrix(cross(r3,v3));

n1:matrix(n1);
n2s:matrix(n2s);
n2c:matrix(n2c);
n3s:matrix(n3s);
n3c:matrix(n3c);
v1:matrix(v1);
v2:matrix(v2);
v3:matrix(v3);

fext:matrix([fx,fy,fz]);
mext:matrix([mx,my,mz]);

fbal:fext+f1*n1+f2*(n2s*sin(phi2)+n2c*cos(phi2))+f3*(n3s*sin(phi3)
+n3c*cos(phi3))+u*f1*v1+u*f2*v2+u*f3*v3+del*(v1+v2+v3);

mbal:mext+f1*r1xn1+u*f1*r1xv1+f2*(r2xn2s*sin(phi2)+r2xn2c*cos(phi2))+u*f2*r2xv2
+f3*(r3xn3s*sin(phi3)+r3xn3c*cos(phi3))+u*f3*r3xv3+del*(r1xv1+r2xv2+r3xv3);

fbalx:fbal[1,1];
fbaly:fbal[1,2];
fbalz:fbal[1,3];

mbalx:mbal[1,1];
mbaly:mbal[1,2];
mbalz:mbal[1,3];

eliminate([fbalx,fbaly,fbalz],[f2,f3])$
solve(%,f1)$
nf1:rhs(%[1]);

```

```
eliminate([fbalx,fbaly,fbalz],[f1,f3])$  
solve(%,f2)$  
nf2:rhs(%[1]);  
  
eliminate([fbalx,fbaly,fbalz],[f1,f2])$  
solve(%,f3)$  
nf3:rhs(%[1]);  
  
f1:nf1$  
f2:nf2$  
f3:nf3$  
f4:""$  
  
save([mndf3a,mec,dsk,dy],mbalx,mbaly,mbalz,f1,f2,f3,f4);
```

```

/* Case3a mec, combines and evaluates parameters for the jamming and
   breaking contact equations */

/* make sure that vals3a has been batched and values chosen for tx,ty,tz */

dynamalloc:print$

loadfile(dtx3a,mec,dsk,nsing);
deltx:ev(del_tx,infeval)$

loadfile(dty3a,mec,dsk,nsing);
del_ty:ev(del_ty)$

loadfile(dtz3a,mec,dsk,nsing);
del_tz:ev(del_tz)$

loadfile(mndf3a,mec,dsk,dy);

del_rx:ev(del_rx,infeval)$
del_ry:ev(del_ry,infeval)$
del_rz:ev(del_rz,infeval)$
rix:ev(rix)$
riy:ev(riy)$
riz:ev(riz)$
r2x:ev(r2x)$
r2y:ev(r2y)$
r2z:ev(r2z)$
r3x:ev(r3x)$
r3y:ev(r3y)$
r3z:ev(r3z)$
n1x:ev(n1x)$
n1y:ev(n1y)$
n1z:ev(n1z)$
n2sx:ev(n2sx)$
n2sy:ev(n2sy)$
n2sz:ev(n2sz)$
n2cx:ev(n2cx)$
n2cy:ev(n2cy)$
n2cz:ev(n2cz)$
n3sx:ev(n3sx)$
n3sy:ev(n3sy)$
n3sz:ev(n3sz)$
n3cx:ev(n3cx)$
n3cy:ev(n3cy)$
n3cz:ev(n3cz)$
v1x:ev(v1x,infeval)$
v1y:ev(v1y,infeval)$
v1z:ev(v1z,infeval)$
v2x:ev(v2x,infeval); /* should be 0 */
v2y:ev(v2y,infeval); /* should be 0 */
v2z:ev(v2z,infeval); /* should be -1 */
v3x:ev(v3x,infeval); /* should be 0 */
v3y:ev(v3y,infeval); /* should be 1 */
v3z:ev(v3z,infeval); /* should be 0 */

ratprint:false$
keepfloat:true$
expop:9$
expon:9$

```



```

ratexpand:false$
numer:true$

f1:ratexpand(ev(f1,ineval))$
f2:ratexpand(ev(f2,ineval))$
f3:ratexpand(ev(f3,ineval))$

mbalx:ratexpand(ev(mbalx,ineval))$
mbaly:ratexpand(ev(mbaly,ineval))$
mbalz:ratexpand(ev(mbalz,ineval))$

solve(mbalx,del)$
expand(rhs(%[1]))$
expmx:ev(%,float);

solve(mbaly,del)$
expand(rhs(%[1]))$
expmy:ev(%,float);

solve(mbalz,del)$
expand(rhs(%[1]))$
expmz:ev(%,float);

assume(mx>0,my>0,mz>0)$

signx:sign(part(expmx,1))$
termx:if signx=neg then -part(expmx,1.1,1) else part(expmx,1,1)$
signx:if signx=neg then "Mx/f is less than (below surface)" else "Mx/f is greater than
(above surface)";
expmx:expand(-(expmx-part(expmx,1))/termx);

signy:sign(part(expmy,1))$
termy:if signy=neg then -part(expmy,1.1,1) else part(expmy,1,1)$
signy:if signy=neg then "My/f is less than (below surface)" else "My/f is greater than
(above surface)";
expmy:expand(-(expmy-part(expmy,1))/termy);

signz:sign(part(expmz,1))$
termz:if signz=neg then -part(expmz,1.1,1) else part(expmz,1,1)$
signz:if signz=neg then "Mz/f is less than (below surface)" else "Mz/f is greater than
(above surface)";
expmz:expand(-(expmz-part(expmz,1))/termz);

del:0.0$

ev(f1)$
f1:expand(%)$
f1+"0=";

ev(f2)$
f2:expand(%)$
f2+"0=";

ev(f3)$
f3:expand(%)$
f3+"0=";

assume(fx>0,fy>0,fz>0)$

c1:if sign(part(f1,1))=neg then -part(f1,1.1,1) else part(f1,1,1)$

```

```

b1:if sign(part(f1,2))=neg then -part(f1,2,1,1) else part(f1,2,1)§
a1:if sign(part(f1,3))=neg then -part(f1,3,1,1) else part(f1,3,1)§
alpha:atan(c1/(a1*cos(beta)+b1*sin(beta)));
root1:atan(-a1/b1)§
if root1<-1.57 or root1>1.57 then flag1:out else flag1'in;
a1*cos(root1+.01)+b1*sin(root1+.01)§
sign1:if sign(%)=neg then "alpha < for Beta > "+root1 else "alpha > for Beta > "+root1;

c2:if sign(part(f2,1))=neg then -part(f2,1,1,1) else part(f2,1,1)§
b2:if sign(part(f2,2))=neg then -part(f2,2,1,1) else part(f2,2,1)§
a2:if sign(part(f2,3))=neg then -part(f2,3,1,1) else part(f2,3,1)§
alpha2:atan(c2/(a2*cos(beta)+b2*sin(beta)));
root2:atan(-a2/b2)§
if root2<-1.57 or root2>1.57 then flag2:out else flag2'in;
a2*cos(root2+.01)+b2*sin(root2+.01)§
sign2:if sign(%)=neg then "alpha < for Beta > "+root2 else "alpha > for Beta > "+root2;

c3:if sign(part(f3,1))=neg then -part(f3,1,1,1) else part(f3,1,1)§
b3:if sign(part(f3,2))=neg then -part(f3,2,1,1) else part(f3,2,1)§
a3:if sign(part(f3,3))=neg then -part(f3,3,1,1) else part(f3,3,1)§
alpha3:atan(c3/(a3*cos(beta)+b3*sin(beta)));
root3:atan(-a3/b3)§
if root3<-1.57 or root3>1.57 then flag3:out else flag3'in;
a3*cos(root3+.01)+b3*sin(root3+.01)§
sign3:if sign(%)=neg then "alpha < for Beta > "+root3 else "alpha > for Beta > "+root3;

fx:sin(alpha)*cos(beta)§
fy:sin(alpha)*sin(beta)§
fz:-cos(alpha)§

expmx:expand(ev(expmx))§
"Mx/F="+expmx;
expmy:expand(ev(expmy))§
"My/F="+expmy;
expmz:expand(ev(expmz))§
"Mz/F="+expmz;

save([cs3avl,mec,dsk,dy],caseno,runname,phi2,phi3,signx,signy,signz,expmx,
expmy,expmz,alpha1,root1,flag1,sign1,alpha2,root2,flag2,sign2,alpha3,root3,
flag3,sign3,f4);

```

## For Case Four

```

/* kin4 mec, kinematic equations and solutions for case4 (post multiplying)*/

dynamalloc:print;

batch(vector,oper);

cmat:transpose(matrix([cc11,cc12,cc13],[cc21,cc22,cc23],[cc31,cc32,cc33]));
r0:matrix([rx,ry,rz]);
i:[1,0,0];
j:[0,1,0];
k:[0,0,1];

/* note lack of contact one */
r2p:matrix([0,0,0]);
r3p:matrix([0,wp-cp,0]);
r4p:matrix([(lp-dp),wp,-ep]); /* note new contact vector */

p2:r2p.cmat+r0;
p3:r3p.cmat+r0;
p4:r4p.cmat+r0;

pt2:part(p2,1);
pt3:part(p3,1);
pt4:part(p4,1);

e1:dot(i,pt2)+1/2;
e2:dot(j,pt2)+w/2;
e3:dot(i,pt3)+1/2; /* redundant/inconsistent equation */
e4:dot(j,pt3)-w/2;
e5:dot(i,pt4)-1/2;
e6:dot(j,pt4)-w/2;
e7:dot(k,pt4)+11;

eq1:ev(e1);
eq2:ev(e2);
eq3:ev(e3); /* redundant/inconsistent equation */
eq4:ev(e4);
eq5:ev(e5);
eq6:ev(e6);
eq7:ev(e7);

constraint:ev(eq3-eq1);

/* inverting equations, without equation containing RPY constraint */

solmat:matrix([0,0,0,1,0,0],[0,0,0,0,1,0],[cc22,0,0,0,1,0],
[0,cc11,-cc13,1,0,0],[0,cc21,-cc23,0,1,0],[0,cc31,-cc33,0,0,1]);

rhmamatrix:matrix([-1/2],[-w/2],[w/2],
[-wp*cc12+1/2],[w/2-wp*cc22],[-11-wp*cc32]);

result:ratsimp(invert(solmat).rhmamatrix);

/* evaluate expressions with rotation matrix elements */

batch(rpyrot);
kill(c);

```

```
ev(wp-result[1,1])$
cp:trigsimp(%);
ev(lp-result[2,1])$
dp:trigsimp(%);
ev(result[3,1])$
ep:trigsimp(%);
ev(result[4,1])$
rx:trigsimp(%);
ev(result[5,1])$
ry:trigsimp(%);
ev(result[6,1])$
rz:trigsimp(%);

/* note lack of r1 */
r2:matrix([-1/2],[-w/2],[b-11]);
r3:matrix([-1/2],[w/2],[nc-11]);
r4:matrix([1/2],[w/2],[-11]);

r0:matrix([rx],[ry],[rz]);

/* note lack of rip */
r2p:matrix([0],[0],[0]);
r3p:matrix([0],[wp-cp],[0]);
r4p:matrix([lp-dp],[wp],[-ep]);

cmat:transpose(cmat);

ev(r2-(cmat.r2p+r0));
%[3,1];
rhs(part(solve(% ,b),1));
b:trigsimp(%);

ev(r3-(cmat.r3p+r0));
%[3,1];
rhs(part(solve(% ,nc),1));
nc:trigsimp(%);

constraint:trigsimp(ev(constraint,infeval));

save([reslt4,nec,dsk,dy],constraint,cp,dp,ep,b,nc,rx,ry,rz);
```

```

/* vel4 mec calculates the angular velocity in terms of the linear
velocity for case4 */

dynamalloc:print;

load(result4);
values;

declare(l,constant);
declare(w,constant);
declare(lp,constant);
declare(wp,constant);
declare(ll,constant);

diff(dp)$
ddp:trigsimp(%);

diff(ep)$
dep:trigsimp(%);

del_const:diff(constraint)$

load(facexp);

ratexpand(del_const)$
solve(% ,del(ty));
del_ty:rhs(part(% ,1));

subst(del_ty,del(ty),ddp)$
ratexpand(%)$
ddp:collectterms(% ,del(tx),del(tz));

subst(del_ty,del(ty),dep)$
ratexpand(%)$
dep:collectterms(% ,del(tx),del(tz));

/* separate equation and represent as a coefficient of del(ty) */

amat:matrix([a11,a12],[a21,a22])$
ainv:invert(amat)$
linvel:matrix([del_dp],[del_ep])$
delang:ainv.linvel$
del_tx:ratsimp(delang[1,1])$
del_tz:ratsimp(delang[2,1])$
ev(part(ddp,2)/del(tx));
a11:trigsimp(%);
ev(part(ddp,1)/del(tz));
a12:trigsimp(%);
ev(part(dep,2)/del(tx));
a21:trigsimp(%);
ev(part(dep,1)/del(tz));
a22:trigsimp(%);

ev(del_tx)$
del_tx:trigsimp(%);
ev(del_tz)$
del_tz:trigsimp(%);

```

```
subst(del_tx,del(tx),del_ty);
subst(del_tz,del(tz),%);
ev(%)$
del_ty:trigsimp(%);

save([dtx4,mec,dsk,nsing],del_tx);
save([dty4,mec,dsk,nsing],del_ty);
save([dtz4,mec,dsk,nsing],del_tz);
```

```

/* nandv4 mec, determines normal and contact velocity vectors for case4 */

dynamalloc:print;

batch(vector,oper);

batch(rpyrot);
cmat:c;
kill(c);

/* normals */

i:[1,0,0];
j:[0,1,0];
k:[0,0,1];
ip:transpose(matrix(i));
jp:transpose(matrix(j));
kp:transpose(matrix(k));
ip:cmat.ip;
jp:cmat.jp;
kp:cmat.kp;
ip:[ip[1,1],ip[2,1],ip[3,1]];
jp:[jp[1,1],jp[2,1],jp[3,1]];
kp:[kp[1,1],kp[2,1],kp[3,1]];

n2s:cross(jp,k);
sqrt(n2s[1]^2+n2s[2]^2+n2s[3]^2);
n2s/%;
n2s:transpose(matrix(%));

n2c:cross(k,ip);
sqrt(n2c[1]^2+n2c[2]^2+n2c[3]^2);
n2c/%;
n2c:transpose(matrix(%));

n3:cross(jp,k); /* note new normal at point 3 */
sqrt(n3[1]^2+n3[2]^2+n3[3]^2);
n3/%;
n3:transpose(matrix(%));

n4:-jp; /* note new normal at point 4 */
n4:transpose(matrix(%));

/* velocities */

trigsimp(diff(cmat))$
subst(deltx,del(tx),%)$
subst(deltz,del(tz),%)$
del_cmat:subst(deltz,del(tz),%);

r2p:matrix([0],[0],[0]);
r3p:matrix([0],[wp-cp],[0]);
r4p:matrix([lp-dp],[wp],[-ep]);

del_r0:matrix([del_rx],[del_ry],[del_rz]);

del_r2:del_cmat.r2p+del_r0;
sqrt(del_r2[1,1]^2+del_r2[2,1]^2+del_r2[3,1]^2);
v2:(del_r2/%);

```

```

del_r3:del_cmat.r3p+del_r0;
sqrt(del_r3[1,1]^2+del_r3[2,1]^2+del_r3[3,1]^2);
v3:(del_r3/);

del_r4:del_cmat.r4p+del_r0;
sqrt(del_r4[1,1]^2+del_r4[2,1]^2+del_r4[3,1]^2);
v4:(del_r4/);

load(result4);

declare(l,constant);
declare(lp,constant);
declare(w,constant);
declare(wp,constant);
declare(ll,constant);

del_rx:trigsimp(diff(rx))$
subst(deltx,del(tx),%)$
subst(delty,del(ty),%)$
del_rx:subst(deltz,del(tz),%);

del_ry:trigsimp(diff(ry))$
subst(deltx,del(tx),%)$
subst(delty,del(ty),%)$
del_ry:subst(deltz,del(tz),%);

del_rz:trigsimp(diff(rz))$
subst(deltx,del(tx),%)$
subst(delty,del(ty),%)$
del_rz:subst(deltz,del(tz),%);

kill(allbut(n2s,n2c,n3,n4,v2,v3,v4,rx,ry,rz,del_rx,del_ry,del_rz,
b,nc,cp,dp,ep,constraint));

r2x:-1/3;
r2y:-w/2;
r2z:ev(b-ll,ineval);
r3x:-1/2;
r3y:w/2;
r3z:ev(nc-ll,ineval);
r4x:1/2;
r4y:w/2;
r4z:-ll;

n2sx:n2s[1,1];
n2sy:n2s[2,1];
n2sz:n2s[3,1];
n2cx:n2c[1,1];
n2cy:n2c[2,1];
n2cz:n2c[3,1];
n3x:n3[1,1];
n3y:n3[2,1];
n3z:n3[3,1];
n4x:n4[1,1];
n4y:n4[2,1];
n4z:n4[3,1];

v2x:v2[1,1];
v2y:v2[2,1];

```



```
v2z:v2[3,1];  
v3x:v3[1,1];  
v3y:v3[2,1];  
v3z:v3[3,1];  
v4x:v4[1,1];  
v4y:v4[2,1];  
v4z:v4[3,1];
```

```
save([result4,mec,dsk,dy],r2x,r2y,r2z,r3x,r3y,r3z,r4x,r4y,r4z,  
n2sx,n2sy,n2sz,n2cx,n2cy,n2cz,n3x,n3y,n3z,n4x,n4y,r4z,  
v2x,v2y,v2z,v3x,v3y,v3z,v4x,v4y,v4z,del_rx,del_ry,del_rz,  
cp,dp,ep,b,nc,rx,ry,rz,constraint);
```

```

/* case4 mec. creates and evaluates parameters for case four */

/* writefile(); */

load(result4);

caseno:"case4";

runname:"run4";

/* peg and hole parameters */

tol:0.006; /* tolerance */
l:1.25;
w:1.00;
lp:l+tol;
wp:w+tol;
ll:2.0;
u:0.5; /* friction */

/* roll-pitch-yaw angles*/

tx:-0.07573; /* -0.08414;*/
tz:-0.00108; /* -0.00114;*/

/* orientation of normals for f1 and f2 */

phi2:%pi/8; /* 0 <= phi2 <= %pi/2 */

/* ty is determined from tx,tz */

load(ims1);
ratprint:false$

/* the linear parameters are */

f(ty):='constraint;

zsolveeps:1.0e-4;
zsolvensig:4;

check():=(sol:zsolve([f],[-1.0]).
ty:part(sol,2,1),ev([b,nc,cp,dp,ep,rx,ry,rz],infeval));
check();
tx;
ty;
tz;

/* edge contact velocities */

del_ep:1.0; /* defined */
del_dp:0.0; /* stay away from corner in hole ? */

/* closefile(); */

/* choose a value for ty and batch case4 mec */

```

```
/* batch(case4); */
```

```

/* frcb14 mec, determines the force and moment balance equations, case4 */

dynamalloc:print;

batch(vector,oper);

r2:[r2x,r2y,r2z];
r3:[r3x,r3y,r3z];
r4:[r4x,r4y,r4z];

n2s:[n2sx,n2sy,n2sz]; /* two components */
n2c:[n2cx,n2cy,n2cz]; /* of normal 2 */
n3:[n3x,n3y,n3z];
n4:[n4x,n4y,n4z];

v2:[v2x,v2y,v2z];
v3:[v3x,v3y,v3z];
v4:[v4x,v4y,v4z];

r2xn2s:matrix(cross(r2,n2s));
r2xn2c:matrix(cross(r2,n2c));
r3xn3:matrix(cross(r3,n3));
r4xn4:matrix(cross(r4,n4));

r2xv2:matrix(cross(r2,v2));
r3xv3:matrix(cross(r3,v3));
r4xv4:matrix(cross(r4,v4));

n2s:matrix(n2s);
n2c:matrix(n2c);
n3:matrix(n3);
n4:matrix(n4);

v2:matrix(v2);
v3:matrix(v3);
v4:matrix(v4);

fext:matrix([fx,fy,fz]);
mext:matrix([mx,my,mz]);

fbal:fext+f2*(n2s*sin(phi2)+n2c*cos(phi2))
+f3*n3+f4*n4+u*f2*v2+u*f3*v3+u*f4*v4+del*(v2+v3+v4);

mbal:mext+f2*(r2xn2s*sin(phi2)+
r2xn2c*cos(phi2))+f3*r3xn3+f4*r4xn4+u*f2*r2xv2+u*f3*r3xv3+u*f4*r4xv4
+del*(r2xv2+r3xv3+r4xv4);

fbalx:ratexpand(fbal[1,1]);
fbaly:ratexpand(fbal[1,2]);
fbalz:ratexpand(fbal[1,3]);

mbalx:ratexpand(mbal[1,1]);
mbaly:ratexpand(mbal[1,2]);
mbalz:ratexpand(mbal[1,3]);

eliminate([fbalx,fbaly,fbalz],[f3,f4]);
solve(%,f2);
nf2:rhs(part(%,1));

```

```
eliminate([fbalx,fbaly,fbalz],[f2,f4]);
solve(%,f3);
nf3:rhs(part(%,1));

eliminate([fbalx,fbaly,fbalz],[f2,f3]);
solve(%,f4);
nf4:rhs(part(%,1));

/* f1:???; */
f2:ratsimp(nf2);
f3:ratsimp(nf3);
f4:ratsimp(nf4);

save([mandf4,nec,dsk,dy],mbalx,mbaly,mbalz,f1,f2,f3,f4);
```

```
/* Case4 mec, collects and evaluates parameters for jamming and breaking
   contact curves for case4 */

/* make sure that vals4 has been batched and values chosen for tx,ty,tz */

dynamalloc:print;

loadfile(dtx4,mec,dsk,nsing);
deltx:ev(del_tx,ineval);

loadfile(dty4,mec,dsk,nsing);
deltty:ev(del_ty,ineval);

loadfile(dtz4,mec,dsk,nsing);
deltz:ev(del_tz,ineval);

loadfile(mandf4,mec,dsk,dy);

del_rx:ev(del_rx,ineval);
del_ry:ev(del_ry,ineval);
del_rz:ev(del_rz,ineval);

r2x:ev(r2x);
r2y:ev(r2y);
r2z:ev(r2z);
r3x:ev(r3x);
r3y:ev(r3y);
r3z:ev(r3z);
r4x:ev(r4x);
r4y:ev(r4y);
r4z:ev(r4z);

n2sx:ev(n2sx);
n2sy:ev(n2sy);
n2sz:ev(n2sz);
n2cx:ev(n2cx);
n2cy:ev(n2cy);
n2cz:ev(n2cz);
n3x:ev(n3x);
n3y:ev(n3y);
n3z:ev(n3z);
n4x:ev(n4x);
n4y:ev(n4y);
n4z:ev(n4z);

v2x:ev(v2x,ineval); /* should be 0 */
v2y:ev(v2y,ineval); /* should be 0 */
v2z:ev(v2z,ineval); /* should be +-1 */
v3x:ev(v3x,ineval);
v3y:ev(v3y,ineval);
v3z:ev(v3z,ineval);
v4x:ev(v4x,ineval);
v4y:ev(v4y,ineval);
v4z:ev(v4z,ineval);

ratprint:false;
keepfloat:true;
expop:9;
expon:9;
```

```

ratexpand:false;
numer:true;

f2:ratexpand(ev(f2,ineval));
f3:ratexpand(ev(f3,ineval));
f4:ratexpand(ev(f4,ineval));

mbalx:ratexpand(ev(mbalx,ineval));
mbaly:ratexpand(ev(mbaly,ineval));
mbalz:ratexpand(ev(mbalz,ineval));

solve(mbalx,del);
expand(rhs(%[1]))$
expmx:ev(%,float);

solve(mbaly,del);
expand(rhs(%[1]))$
expmx:ev(%,float);

solve(mbalz,del);
expand(rhs(%[1]))$
expmz:ev(%,float);

assume(mx>0,my>0,mz>0)$

signx:sign(part(expmx,1));
termx:if signx=neg then -part(expmx,1.1.1) else part(expmx,1.1);
signx:if signx=neg then "Mx/f is less than (below surface)" else "Mx/f is greater than
(above surface)";
expmx:expand(-(expmx-part(expmx,1))/termx);

signy:sign(part(expmx,1));
termy:if signy=neg then -part(expmx,1.1.1) else part(expmx,1.1);
signy:if signy=neg then "My/f is less than (below surface)" else "My/f is greater than
(above surface)";
expmx:expand(-(expmx-part(expmx,1))/termy);

signz:sign(part(expmz,1));
termz:if signz=neg then -part(expmz,1.1.1) else part(expmz,1.1);
signz:if signz=neg then "Mz/f is less than (below surface)" else "Mz/f is greater than
(above surface)";
expmz:expand(-(expmz-part(expmz,1))/termz);

del:0.0$

ev(f2)$
f2:expand(%)$
f2+"0=";

ev(f3)$
f3:expand(%)$
f3+"0=";

ev(f4)$
f4:expand(%)$
f4+"0=";

assume(fx>0,fy>0,fz>0)$

c2:if sign(part(f2,1))=neg then -part(f2,1.1.1) else part(f2,1.1)$

```

```

b2:if sign(part(f2,2))=neg then -part(f2,2,1,1) else part(f2,2,1,1)$
a2:if sign(part(f2,3))=neg then -part(f2,3,1,1) else part(f2,3,1,1)$
alpha2:atan(c2/(a2*cos(beta)+b2*sin(beta)));
root2:atan(-a2/b2)$
if root2<-1.57 or root2>1.57 then *lag2:out else flag2:in;
a2*cos(root2+.01)+b2*sin(root2+.01)$
sign2:if sign(%)=neg then "alpha < for Beta > "+root2 else "alpha > for Beta > "+root2;

c3:if sign(part(f3,1))=neg then -part(f3,1,1,1) else part(f3,1,1,1)$
b3:if sign(part(f3,2))=neg then -part(f3,2,1,1) else part(f3,2,1,1)$
a3:if sign(part(f3,3))=neg then -part(f3,3,1,1) else part(f3,3,1,1)$
alpha3:atan(c3/(a3*cos(beta)+b3*sin(beta)));
root3:atan(-a3/b3)$
if root3<-1.57 or root3>1.57 then flag3:out else flag3:in;
a3*cos(root3+.01)+b3*sin(root3+.01)$
sign3:if sign(%)=neg then "alpha < for Beta > "+root3 else "alpha > for Beta > "+root3;

c4:if sign(part(f4,1))=neg then -part(f4,1,1,1) else part(f4,1,1,1)$
b4:if sign(part(f4,2))=neg then -part(f4,2,1,1) else part(f4,2,1,1)$
a4:if sign(part(f4,3))=neg then -part(f4,3,1,1) else part(f4,3,1,1)$
alpha4:atan(c4/(a4*cos(beta)+b4*sin(beta)));
root4:atan(-a4/b4)$
if root4<-1.57 or root4>1.57 then flag4:out else flag4:in;
a4*cos(root4+.01)+b4*sin(root4+.01)$
sign4:if sign(%)=neg then "alpha < for Beta > "+root4 else "alpha > for Beta > "+root4;

fx:sin(alpha)*cos(beta)$
fy:sin(alpha)*sin(beta)$
fz:-cos(alpha)$

expmx:expand(ev(exp~x))$
"Mx/F="+expmx;
expmy:expand(ev(exp~y))$
"My/F="+expmy;
expmz:expand(ev(exp~z))$
"Mz/F="+expmz;

/* To interface with PLOT MEC, change f4 specifications to f1 */

alpha1:alpha4$
root1:root4$
flag1:flag4$
sign1:sign4$

save([cs4val,mec,dsk,dy],caseno,runname,phi2,signx,signy,signz,expmx,
expmy,expmz,alpha2,root2,flag2,sign2,alpha3,root3,
flag3,sign3,alpha1,root1,flag1,sign1);

```



# Appendix B

## Expressions for Case One

As an example of the expressions generated by the MACSYMA™ code of the previous appendix, we provide the following listing from case one. The list of variables are:

For the peg and hole parameters:

L =  $l$   
W =  $w$   
LP =  $l'$   
WP =  $w'$   
LL =  $L$   
U =  $\mu$

For the position terms:

$$TX = \theta_x'$$

$$TY = \theta_y'$$

$$TZ = \theta_z'$$

$$A = a$$

$$B = b$$

$$C = c$$

$$AP = a'$$

$$BP = b'$$

$$CP = c'$$

$$RX = Rx_0$$

$$RY = Ry_0$$

$$RZ = Rz_0$$

For the velocity terms:

$$DELTX = \dot{\theta}_x'$$

$$DELT Y = \dot{\theta}_y'$$

$$DELT Z = \dot{\theta}_z'$$

$$DEL.B = \dot{b}$$

$$DEL.BP = \dot{b}'$$

$$DEL.CP = \dot{c}'$$

$$DEL.RX = \dot{R}x_0$$

$$DEL.RY = \dot{R}y_0$$

$$DEL.RZ = \dot{R}z_0$$

For the vector components of the force balance terms:

$$FX = F_x$$

$$FY = F_y$$

$$FZ = F_z$$

$$MX = M_x$$

$$MY = M_y$$

$$MZ = M_z$$

$$R1X = R_1 \quad x - \text{component}$$

$$R1Y = R_1 \quad y - \text{component}$$

$$R1Z = R_1 \quad z - \text{component} \quad (R_2, R_3 \text{ similar})$$

$$N1X = n_1 \quad x - \text{component}$$

$$N1Y = n_1 \quad y - \text{component}$$

$$N1Z = n_1 \quad z - \text{component} \quad (n_2, n_3 \text{ similar})$$

$$V1X = v_1 \quad x - \text{component}$$

$$V1Y = v_1 \quad y - \text{component}$$

$$V1Z = v_1 \quad z - \text{component} \quad (v_2, v_3 \text{ similar})$$

**Kinematic expressions from KIN1 MEC**

THE EDGE CONTACT PARAMETERS ARE:

A=

$$\frac{LP \cos(TX) \sin(TY) \sin(TZ) - LP \sin(TX) \cos(TZ) + L \sin(TX) \cos(TY)}{\sin(TX) \cos(TY)}$$

B=

$$\begin{aligned} & - (LP \cos(TX) \sin(TX) \sin(TY) \sin^2(TZ) \\ & + (-LP \cos^2(TX) \cos^2(TY) + 2LP \cos^2(TX) - LP) \cos(TZ) \sin(TZ) \\ & - LP \cos(TX) \sin(TX) \sin(TY) \cos^2(TZ)) / (\cos(TX) \sin(TX) \cos^2(TY)) \end{aligned}$$

C=

$$\frac{\sin(TX) \cos(TY) W + LP \sin(TX) \sin(TZ) + LP \cos(TX) \sin(TY) \cos(TZ)}{\sin(TX) \cos(TY)}$$

AP=

$$\frac{LP \cos(TX) \sin(TY) \sin^2(TZ) - LP \sin(TX) \cos(TZ) \sin(TZ)}{\cos(TX) \sin(TX) \cos(TY)}$$

BP=

$$\begin{aligned} & (LP \cos(TX) \sin(TX) \sin^2(TZ) + (LP - 2LP \sin^2(TX)) \sin(TY) \cos(TZ) \\ & \sin(TZ) - LP \cos(TX) \sin(TX) \sin^2(TY) \cos^2(TZ)) / (\cos(TX) \sin(TX) \cos^2(TY)) \end{aligned}$$

CP=

$$(\cos(TX) \sin(TX) \cos(TY) WP + LP \sin(TX) \cos(TZ) \sin(TZ))$$

$$+ LP \cos(TX) \sin(TY) \cos^2(TZ) / (\cos(TX) \sin(TX) \cos(TY))$$

THE COMPONENTS OF THE VECTOR RO ARE:

RX=

$$\begin{aligned} & ((4 LP \sin^2(TX) - 2 LP) \sin(TY) \cos^2(TZ) \sin(TZ) \\ & + (2 LP \cos(TX) \sin(TX) \sin^2(TY) + 2 LP \cos(TX) \sin(TX)) \cos^3(TZ) \\ & - 2 LP \cos(TX) \sin(TX) \cos(TZ) - L \cos(TX) \sin(TX) \cos(TY)) \\ & / (2 \cos(TX) \sin(TX) \cos(TY)) \end{aligned}$$

RY=

$$\begin{aligned} & - (\cos(TX) \sin(TX) \cos(TY) W + (2 LP \cos(TX) \sin(TX) \sin^2(TY) \\ & + 2 LP \cos(TX) \sin(TX)) \sin^3(TZ) + (2 LP - 4 LP \sin^2(TX)) \sin(TY) \cos(TZ) \\ & \sin^2(TZ) - 2 LP \cos(TX) \sin(TX) \sin^2(TY) \sin(TZ)) / (2 \cos(TX) \sin(TX) \cos(TY)) \end{aligned}$$

RZ=

$$\frac{LP \sin(TX) \cos(TZ) \sin(TZ) + LP \cos(TX) \sin(TY) \cos^2(TZ) - LL \cos(TX)}{\cos(TX)}$$

**Force and moment balance results from FRCBL1 MEC**

## FORCE BALANCE, X DIRECTION

$$(V3X + V2X + V1X) \text{ DEL} + F3 \text{ U } V3X + F2 \text{ U } V2X + F1 \text{ U } V1X + F3 \text{ N3X} + F2 \text{ N2X} \\ + F1 \text{ N1X} + FX = 0$$

## FORCE BALANCE, Y DIRECTION

$$(V3Y + V2Y + V1Y) \text{ DEL} + F3 \text{ U } V3Y + F2 \text{ U } V2Y + F1 \text{ U } V1Y + F3 \text{ N3Y} + F2 \text{ N2Y} \\ + F1 \text{ N1Y} + FY = 0$$

## FORCE BALANCE, Z DIRECTION

$$(V3Z + V2Z + V1Z) \text{ DEL} + F3 \text{ U } V3Z + F2 \text{ U } V2Z + F1 \text{ U } V1Z + F3 \text{ N3Z} + F2 \text{ N2Z} \\ + F1 \text{ N1Z} + FZ = 0$$

## MOMENT BALANCE, X DIRECTION

$$(R3Y \text{ V3Z} - R3Z \text{ V3Y} + R2Y \text{ V2Z} - R2Z \text{ V2Y} + R1Y \text{ V1Z} - R1Z \text{ V1Y}) \text{ DEL} \\ + F3 \text{ U } (R3Y \text{ V3Z} - R3Z \text{ V3Y}) + F2 \text{ U } (R2Y \text{ V2Z} - R2Z \text{ V2Y}) \\ + F1 \text{ U } (R1Y \text{ V1Z} - R1Z \text{ V1Y}) + F3 \text{ (N3Z R3Y} - \text{N3Y R3Z)} + F2 \text{ (N2Z R2Y} - \text{N2Y R2Z)} \\ + F1 \text{ (N1Z R1Y} - \text{N1Y R1Z)} + MX = 0$$

## MOMENT BALANCE, Y DIRECTION

$$(- R3X \text{ V3Z} + R3Z \text{ V3X} - R2X \text{ V2Z} + R2Z \text{ V2X} - R1X \text{ V1Z} + R1Z \text{ V1X}) \text{ DEL} \\ + F3 \text{ U } (R3Z \text{ V3X} - R3X \text{ V3Z}) + F2 \text{ U } (R2Z \text{ V2X} - R2X \text{ V2Z}) \\ + F1 \text{ U } (R1Z \text{ V1X} - R1X \text{ V1Z}) + F3 \text{ (N3X R3Z} - \text{N3Z R3X)} + F2 \text{ (N2X R2Z} - \text{N2Z R2X)} \\ + F1 \text{ (N1X R1Z} - \text{N1Z R1X)} + MY = 0$$

## MOMENT BALANCE, Z DIRECTION

$$(R3X \text{ V3Y} - R3Y \text{ V3X} + R2X \text{ V2Y} - R2Y \text{ V2X} + R1X \text{ V1Y} - R1Y \text{ V1X}) \text{ DEL} \\ + F3 \text{ U } (R3X \text{ V3Y} - R3Y \text{ V3X}) + F2 \text{ U } (R2X \text{ V2Y} - R2Y \text{ V2X}) \\ + F1 \text{ U } (R1X \text{ V1Y} - R1Y \text{ V1X}) + F3 \text{ (N3Y R3X} - \text{N3X R3Y)} + F2 \text{ (N2Y R2X} - \text{N2X R2Y)}$$

APPENDIX B. EXPRESSIONS FOR CASE ONE

239

$$+ F_1 (N_{1Y} R_{1X} - N_{1X} R_{1Y}) + M_Z = 0$$

REACTION FORCE,  $F_1 =$

$$\begin{aligned} & - \left( (U^2 V_{1X} + (-N_{3X} - N_{2X}) U \right) V_{2Y} + ((N_{3Y} + N_{2Y}) U - U^2 V_{1Y}) V_{2X} \\ & - N_{2X} U V_{1Y} + N_{2Y} U V_{1X} + N_{2X} N_{3Y} - N_{2Y} N_{3X}) V_{3Z} \\ & + \left( (N_{3X} + N_{2X}) U - U^2 V_{1X} \right) V_{2Z} + (U^2 V_{1Z} + (-N_{3Z} - N_{2Z}) U) V_{2X} + N_{2X} U V_{1Z} \\ & - N_{2Z} U V_{1X} - N_{2X} N_{3Z} + N_{2Z} N_{3X}) V_{3Y} + ((U^2 V_{1Y} + (-N_{3Y} - N_{2Y}) U) V_{2Z} \\ & + ((N_{3Z} + N_{2Z}) U - U^2 V_{1Z}) V_{2Y} - N_{2Y} U V_{1Z} + N_{2Z} U V_{1Y} + N_{2Y} N_{3Z} - N_{2Z} N_{3Y}) \\ & V_{3X} + (N_{3X} U V_{1Y} - N_{3Y} U V_{1X} + N_{2X} N_{3Y} - N_{2Y} N_{3X}) V_{2Z} \\ & + (-N_{3X} U V_{1Z} + N_{3Z} U V_{1X} - N_{2X} N_{3Z} + N_{2Z} N_{3X}) V_{2Y} \\ & + (N_{3Y} U V_{1Z} - N_{3Z} U V_{1Y} + N_{2Y} N_{3Z} - N_{2Z} N_{3Y}) V_{2X} + (N_{2X} N_{3Y} - N_{2Y} N_{3X}) V_{1Z} \\ & + (N_{2Z} N_{3X} - N_{2X} N_{3Z}) V_{1Y} + (N_{2Y} N_{3Z} - N_{2Z} N_{3Y}) V_{1X}) \text{ DEL} \\ & + (F_X U^2 V_{2Y} - F_Y U^2 V_{2X} + (F_X N_{2Y} - F_Y N_{2X}) U) V_{3Z} \\ & + (-F_X U^2 V_{2Z} + F_Z U^2 V_{2X} + (F_Z N_{2X} - F_X N_{2Z}) U) V_{3Y} \\ & + (F_Y U^2 V_{2Z} - F_Z U^2 V_{2Y} + (F_Y N_{2Z} - F_Z N_{2Y}) U) V_{3X} + (F_Y N_{3X} - F_X N_{3Y}) U V_{2Z} \\ & + (F_X N_{3Z} - F_Z N_{3X}) U V_{2Y} + (F_Z N_{3Y} - F_Y N_{3Z}) U V_{2X} + (F_X N_{2Y} - F_Y N_{2X}) N_{3Z} \\ & + (F_Z N_{2X} - F_X N_{2Z}) N_{3Y} + (F_Y N_{2Z} - F_Z N_{2Y}) N_{3X}) \\ & / \left( (U^3 V_{1X} + N_{1X} U^2) V_{2Y} + (-U^3 V_{1Y} - N_{1Y} U^2) V_{2X} - N_{2X} U^2 V_{1Y} + N_{2Y} U^2 V_{1X} \right. \\ & + (N_{1X} N_{2Y} - N_{1Y} N_{2X}) U) V_{3Z} + ((-U^3 V_{1X} - N_{1X} U^2) V_{2Z} \\ & + (U^3 V_{1Z} + N_{1Z} U^2) V_{2X} + N_{2X} U^2 V_{1Z} - N_{2Z} U^2 V_{1X} + (N_{1Z} N_{2X} - N_{1X} N_{2Z}) U) \\ & V_{3Y} + ((U^3 V_{1Y} + N_{1Y} U^2) V_{2Z} + (-U^3 V_{1Z} - N_{1Z} U^2) V_{2Y} - N_{2Y} U^2 V_{1Z} \\ & \left. + N_{2Z} U^2 V_{1Y} + (N_{1Y} N_{2Z} - N_{1Z} N_{2Y}) U) V_{3X} \right) \end{aligned}$$

$$\begin{aligned}
& + (N3X U^2 V1Y - N3Y U^2 V1X + (N1Y N3X - N1X N3Y) U) V2Z \\
& + (- N3X U^2 V1Z + N3Z U^2 V1X + (N1X N3Z - N1Z N3X) U) V2Y \\
& + (N3Y U^2 V1Z - N3Z U^2 V1Y + (N1Z N3Y - N1Y N3Z) U) V2X \\
& + (N2X N3Y - N2Y N3X) U V1Z + (N2Z N3X - N2X N3Z) U V1Y \\
& + (N2Y N3Z - N2Z N3Y) U V1X + (N1X N2Y - N1Y N2X) N3Z \\
& + (N1Z N2X - N1X N2Z) N3Y + (N1Y N2Z - N1Z N2Y) N3X
\end{aligned}$$

REACTION FORCE, F2=

$$\begin{aligned}
& - (((U^2 V1X + N1X U) V2Y + (- U^2 V1Y - N1Y U) V2X + (N3X + N1X) U V1Y \\
& + (- N3Y - N1Y) U V1X - N1X N3Y + N1Y N3X) V3Z \\
& + ((- U^2 V1X - N1X U) V2Z + (U^2 V1Z + N1Z U) V2X + (- N3X - N1X) U V1Z \\
& + (N3Z + N1Z) U V1X + N1X N3Z - N1Z N3X) V3Y \\
& + ((U^2 V1Y + N1Y U) V2Z + (- U^2 V1Z - N1Z U) V2Y + (N3Y + N1Y) U V1Z \\
& + (- N3Z - N1Z) U V1Y - N1Y N3Z + N1Z N3Y) V3X \\
& + (N3X U V1Y - N3Y U V1X - N1X N3Y + N1Y N3X) V2Z \\
& + (- N3X U V1Z + N3Z U V1X + N1X N3Z - N1Z N3X) V2Y \\
& + (N3Y U V1Z - N3Z U V1Y - N1Y N3Z + N1Z N3Y) V2X + (N1Y N3X - N1X N3Y) V1Z \\
& + (N1X N3Z - N1Z N3X) V1Y + (N1Z N3Y - N1Y N3Z) V1X) DEL \\
& + (- FX U^2 V1Y + FY U^2 V1X + (FY N1X - FX N1Y) U) V3Z \\
& + (FX U^2 V1Z - FZ U^2 V1X + (FX N1Z - FZ N1X) U) V3Y \\
& + (- FY U^2 V1Z + FZ U^2 V1Y + (FZ N1Y - FY N1Z) U) V3X \\
& + (FX N3Y - FY N3X) U V1Z + (FZ N3X - FX N3Z) U V1Y + (FY N3Z - FZ N3Y) U V1X \\
& + (FY N1X - FX N1Y) N3Z + (FX N1Z - FZ N1X) N3Y + (FZ N1Y - FY N1Z) N3X)
\end{aligned}$$



APPENDIX B. EXPRESSIONS FOR CASE ONE

241

$$\begin{aligned}
 & /(((U^3 V1X + N1X U^2) V2Y + (-U^3 V1Y - N1Y U^2) V2X - N2X U^2 V1Y + N2Y U^2 V1X \\
 & + (N1X N2Y - N1Y N2X) U) V3Z + ((-U^3 V1X - N1X U^2) V2Z \\
 & + (U^3 V1Z + N1Z U^2) V2X + N2X U^2 V1Z - N2Z U^2 V1X + (N1Z N2X - N1X N2Z) U) \\
 & V3Y + ((U^3 V1Y + N1Y U^2) V2Z + (-U^3 V1Z - N1Z U^2) V2Y - N2Y U^2 V1Z \\
 & + N2Z U^2 V1Y + (N1Y N2Z - N1Z N2Y) U) V3X \\
 & + (N3X U^2 V1Y - N3Y U^2 V1X + (N1Y N3X - N1X N3Y) U) V2Z \\
 & + (-N3X U^2 V1Z + N3Z U^2 V1X + (N1X N3Z - N1Z N3X) U) V2Y \\
 & + (N3Y U^2 V1Z - N3Z U^2 V1Y + (N1Z N3Y - N1Y N3Z) U) V2X \\
 & + (N2X N3Y - N2Y N3X) U V1Z + (N2Z N3X - N2X N3Z) U V1Y \\
 & + (N2Y N3Z - N2Z N3Y) U V1X + (N1X N2Y - N1Y N2X) N3Z \\
 & + (N1Z N2X - N1X N2Z) N3Y + (N1Y N2Z - N1Z N2Y) N3X)
 \end{aligned}$$

REACTION FORCE, F3=

$$\begin{aligned}
 & - (((U^2 V1X + N1X U) V2Y + (-U^2 V1Y - N1Y U) V2X - N2X U V1Y \\
 & + N2Y U V1X + N1X N2Y - N1Y N2X) V3Z + ((-U^2 V1X - N1X U) V2Z \\
 & + (U^2 V1Z + N1Z U) V2X + N2X U V1Z - N2Z U V1X - N1X N2Z + N1Z N2X) V3Y \\
 & + ((U^2 V1Y + N1Y U) V2Z + (-U^2 V1Z - N1Z U) V2Y - N2Y U V1Z + N2Z U V1Y \\
 & + N1Y N2Z - N1Z N2Y) V3X + ((-N2X - N1X) U V1Y + (N2Y + N1Y) U V1X + N1X N2Y \\
 & - N1Y N2X) V2Z + ((N2X + N1X) U V1Z + (-N2Z - N1Z) U V1X - N1X N2Z \\
 & + N1Z N2X) V2Y + ((-N2Y - N1Y) U V1Z + (N2Z + N1Z) U V1Y + N1Y N2Z \\
 & - N1Z N2Y) V2X + (N1X N2Y - N1Y N2X) V1Z + (N1Z N2X - N1X N2Z) V1Y
 \end{aligned}$$

$$\begin{aligned}
& + (N1Y N2Z - N1Z N2Y) V1X) DEL + (FX U^2 V1Y - FY U^2 V1X \\
& + (FX N1Y - FY N1X) U) V2Z + (- FX U^2 V1Z + FZ U^2 V1X + (FZ N1X - FX N1Z) U) \\
& V2Y + (FY U^2 V1Z - FZ U^2 V1Y + (FY N1Z - FZ N1Y) U) V2X \\
& + (FY N2X - FX N2Y) U V1Z + (FX N2Z - FZ N2X) U V1Y + (FZ N2Y - FY N2Z) U V1X \\
& + (FX N1Y - FY N1X) N2Z + (FZ N1X - FX N1Z) N2Y + (FY N1Z - FZ N1Y) N2X) \\
& /(((U^3 V1X + N1X U^2) V2Y + (- U^3 V1Y - N1Y U^2) V2X - N2X U^2 V1Y + N2Y U^2 V1X \\
& + (N1X N2Y - N1Y N2X) U) V3Z + ((- U^3 V1X - N1X U^2) V2Z \\
& + (U^3 V1Z + N1Z U^2) V2X + N2X U^2 V1Z - N2Z U^2 V1X + (N1Z N2X - N1X N2Z) U) \\
& V3Y + ((U^3 V1Y + N1Y U^2) V2Z + (- U^3 V1Z - N1Z U^2) V2Y - N2Y U^2 V1Z \\
& + N2Z U^2 V1Y + (N1Y N2Z - N1Z N2Y) U) V3X \\
& + (N3X U^2 V1Y - N3Y U^2 V1X + (N1Y N3X - N1X N3Y) U) V2Z \\
& + (- N3X U^2 V1Z + N3Z U^2 V1X + (N1X N3Z - N1Z N3X) U) V2Y \\
& + (N3Y U^2 V1Z - N3Z U^2 V1Y + (N1Z N3Y - N1Y N3Z) U) V2X \\
& + (N2X N3Y - N2Y N3X) U V1Z + (N2Z N3X - N2X N3Z) U V1Y \\
& + (N2Y N3Z - N2Z N3Y) U V1X + (N1X N2Y - N1Y N2X) N3Z \\
& + (N1Z N2X - N1X N2Z) N3Y + (N1Y N2Z - N1Z N2Y) N3X)
\end{aligned}$$

The components of the force and moment balance equations are:

From VEL1 MEC

DELTX=

$$\begin{aligned}
 & - (((2 \text{ DEL\_B} \text{ COS}^5(\text{TX}) - 2 \text{ DEL\_B} \text{ COS}^3(\text{TX})) \text{ COS}^2(\text{TY})) \\
 & + (4 \text{ DEL\_CP} \text{ COS}(\text{TX}) - 4 \text{ DEL\_CP} \text{ COS}^3(\text{TX})) \text{ SIN}(\text{TX}) \text{ COS}(\text{TY}) - 8 \text{ DEL\_B} \text{ COS}^5(\text{TX}) \\
 & + 14 \text{ DEL\_B} \text{ COS}^3(\text{TX}) - 6 \text{ DEL\_B} \text{ COS}(\text{TX})) \text{ SIN}(\text{TY}) \\
 & + (6 \text{ DEL\_BP} \text{ COS}^5(\text{TX}) - 10 \text{ DEL\_BP} \text{ COS}^3(\text{TX}) + 4 \text{ DEL\_BP} \text{ COS}(\text{TX})) \text{ COS}^2(\text{TY}) \\
 & - 8 \text{ DEL\_BP} \text{ COS}^5(\text{TX}) + 14 \text{ DEL\_BP} \text{ COS}^3(\text{TX}) - 6 \text{ DEL\_BP} \text{ COS}(\text{TX})) \text{ SIN}^4(\text{TZ}) \\
 & + (((2 \text{ DEL\_BP} \text{ COS}^2(\text{TX}) - 2 \text{ DEL\_BP} \text{ COS}^4(\text{TX})) \text{ SIN}(\text{TX}) \text{ COS}^2(\text{TY})) \\
 & + (8 \text{ DEL\_BP} \text{ COS}^4(\text{TX}) - 10 \text{ DEL\_BP} \text{ COS}^2(\text{TX}) + 2 \text{ DEL\_BP}) \text{ SIN}(\text{TX})) \text{ SIN}(\text{TY}) \\
 & + (2 \text{ DEL\_CP} \text{ COS}^4(\text{TX}) - 2 \text{ DEL\_CP} \text{ COS}^2(\text{TX})) \text{ COS}^3(\text{TY}) \\
 & + (6 \text{ DEL\_B} \text{ COS}^2(\text{TX}) - 6 \text{ DEL\_B} \text{ COS}^4(\text{TX})) \text{ SIN}(\text{TX}) \text{ COS}^2(\text{TY}) \\
 & + (-4 \text{ DEL\_CP} \text{ COS}^4(\text{TX}) + 6 \text{ DEL\_CP} \text{ COS}^2(\text{TX}) - 2 \text{ DEL\_CP}) \text{ COS}(\text{TY}) \\
 & + (8 \text{ DEL\_B} \text{ COS}^4(\text{TX}) - 10 \text{ DEL\_B} \text{ COS}^2(\text{TX}) + 2 \text{ DEL\_B}) \text{ SIN}(\text{TX})) \text{ COS}(\text{TZ}) \text{ SIN}^3(\text{TZ}) \\
 & + (((2 \text{ DEL\_B} \text{ COS}^3(\text{TX}) - 2 \text{ DEL\_B} \text{ COS}^5(\text{TX})) \text{ COS}^2(\text{TY})) \\
 & + (4 \text{ DEL\_CP} \text{ COS}^3(\text{TX}) - 4 \text{ DEL\_CP} \text{ COS}(\text{TX})) \text{ SIN}(\text{TX}) \text{ COS}(\text{TY}) + 6 \text{ DEL\_B} \text{ COS}^5(\text{TX}) \\
 & - 11 \text{ DEL\_B} \text{ COS}^3(\text{TX}) + 5 \text{ DEL\_B} \text{ COS}(\text{TX})) \text{ SIN}(\text{TY}) \\
 & + (-5 \text{ DEL\_BP} \text{ COS}^5(\text{TX}) + 9 \text{ DEL\_BP} \text{ COS}^3(\text{TX}) - 4 \text{ DEL\_BP} \text{ COS}(\text{TX})) \text{ COS}^2(\text{TY}) \\
 & + 6 \text{ DEL\_BP} \text{ COS}^5(\text{TX}) - 11 \text{ DEL\_BP} \text{ COS}^3(\text{TX}) + 5 \text{ DEL\_BP} \text{ COS}(\text{TX})) \text{ SIN}^2(\text{TZ}) \\
 & + (((\text{DEL\_BP} \text{ COS}^4(\text{TX}) - \text{DEL\_BP} \text{ COS}^2(\text{TX})) \text{ SIN}(\text{TX}) \text{ COS}^2(\text{TY}))
 \end{aligned}$$

$$\begin{aligned}
 & + (-2 \text{DEL\_BP}^4 \text{COS}^2(\text{TX}) + 3 \text{DEL\_BP}^2 \text{COS}^2(\text{TX}) - \text{DEL\_BP}) \text{SIN}(\text{TX}) \text{SIN}(\text{TY}) \\
 & + (\text{DEL\_CP}^2 \text{COS}^2(\text{TX}) - \text{DEL\_CP}^4 \text{COS}^3(\text{TX})) \text{COS}^3(\text{TY}) \\
 & + (2 \text{DEL\_B}^4 \text{COS}^4(\text{TX}) - 2 \text{DEL\_B}^2 \text{COS}^2(\text{TX})) \text{SIN}(\text{TX}) \text{COS}^2(\text{TY}) \\
 & + (2 \text{DEL\_CP}^4 \text{COS}^4(\text{TX}) - 3 \text{DEL\_CP}^2 \text{COS}^2(\text{TX}) + \text{DEL\_CP}) \text{COS}(\text{TY}) \\
 & + (-2 \text{DEL\_B}^4 \text{COS}^4(\text{TX}) + 3 \text{DEL\_B}^2 \text{COS}^2(\text{TX}) - \text{DEL\_B}) \text{SIN}(\text{TX}) \text{COS}(\text{TZ}) \text{SIN}(\text{TZ}) \\
 & + ((\text{DEL\_CP}^3 \text{COS}(\text{TX}) - \text{DEL\_CP}^3 \text{COS}^3(\text{TX})) \text{SIN}(\text{TX}) \text{COS}(\text{TY}) + \text{DEL\_B}^3 \text{COS}^3(\text{TX}) \\
 & - \text{DEL\_B} \text{COS}(\text{TX})) \text{SIN}(\text{TY}) + (\text{DEL\_BP}^3 \text{COS}(\text{TX}) - \text{DEL\_BP}^3 \text{COS}^2(\text{TX})) \text{COS}^2(\text{TY}) \\
 & + \text{DEL\_BP}^3 \text{COS}^3(\text{TX}) - \text{DEL\_BP}^3 \text{COS}(\text{TX})) / (((\text{LP}^3 \text{COS}^3(\text{TX}) \text{COS}^2(\text{TY}) - 4 \text{LP}^3 \text{COS}^3(\text{TX}) \\
 & + 3 \text{LP} \text{COS}(\text{TX})) \text{SIN}(\text{TY}) \text{COS}^3(\text{TZ}) + (\text{LP}^3 \text{COS}^3(\text{TX}) - \text{LP} \text{COS}(\text{TX})) \text{SIN}(\text{TY}) \\
 & \text{COS}(\text{TZ})) \text{SIN}(\text{TZ}) + ((4 \text{LP}^2 \text{COS}^2(\text{TX}) - \text{LP}) \text{SIN}(\text{TX}) \\
 & - 3 \text{LP}^2 \text{COS}^2(\text{TX}) \text{SIN}(\text{TX}) \text{COS}^2(\text{TY})) \text{COS}^4(\text{TZ}) \\
 & + (2 \text{LP}^2 \text{COS}^2(\text{TX}) \text{SIN}(\text{TX}) \text{COS}^2(\text{TY}) + (\text{LP} - 3 \text{LP}^2 \text{COS}^2(\text{TX})) \text{SIN}(\text{TX})) \text{COS}^2(\text{TZ})
 \end{aligned}$$

DELTY=

$$\begin{aligned}
 & - (((2 \text{DEL\_B}^5 \text{COS}^5(\text{TX}) - 2 \text{DEL\_B}^3 \text{COS}^3(\text{TX})) \text{COS}^3(\text{TY}) \\
 & + (4 \text{DEL\_CP}^3 \text{COS}(\text{TX}) - 4 \text{DEL\_CP}^3 \text{COS}^3(\text{TX})) \text{SIN}(\text{TX}) \text{COS}^2(\text{TY}) \\
 & + (-8 \text{DEL\_B}^5 \text{COS}^5(\text{TX}) + 14 \text{DEL\_B}^3 \text{COS}^3(\text{TX}) - 6 \text{DEL\_B} \text{COS}(\text{TX})) \text{COS}(\text{TY}) \text{SIN}(\text{TY}) \\
 & + (6 \text{DEL\_BP}^5 \text{COS}^5(\text{TX}) - 10 \text{DEL\_BP}^3 \text{COS}^3(\text{TX}) + 4 \text{DEL\_BP} \text{COS}(\text{TX})) \text{COS}^3(\text{TY}) \\
 & + (-8 \text{DEL\_BP}^5 \text{COS}^5(\text{TX}) + 14 \text{DEL\_BP}^3 \text{COS}^3(\text{TX}) - 6 \text{DEL\_BP} \text{COS}(\text{TX})) \text{COS}(\text{TY})
 \end{aligned}$$

$$\begin{aligned}
& \text{SIN}^3(\text{TZ}) + (((2 \text{DEL\_BP}^2 \text{COS}^2(\text{TX}) - 2 \text{DEL\_BP}^4 \text{COS}^4(\text{TX})) \text{SIN}(\text{TX}) \text{COS}^3(\text{TY}) \\
& + (8 \text{DEL\_BP}^4 \text{COS}^4(\text{TX}) - 10 \text{DEL\_BP}^2 \text{COS}^2(\text{TX}) + 2 \text{DEL\_BP}) \text{SIN}(\text{TX}) \text{COS}(\text{TY})) \\
& \text{SIN}(\text{TY}) + (2 \text{DEL\_CP}^4 \text{COS}^4(\text{TX}) - 2 \text{DEL\_CP}^2 \text{COS}^2(\text{TX})) \text{COS}^4(\text{TY}) \\
& + (6 \text{DEL\_B}^2 \text{COS}^2(\text{TX}) - 6 \text{DEL\_B}^4 \text{COS}^4(\text{TX})) \text{SIN}(\text{TX}) \text{COS}^3(\text{TY}) \\
& + (-4 \text{DEL\_CP}^4 \text{COS}^4(\text{TX}) + 6 \text{DEL\_CP}^2 \text{COS}^2(\text{TX}) - 2 \text{DEL\_CP}) \text{COS}^2(\text{TY}) \\
& + (8 \text{DEL\_B}^4 \text{COS}^4(\text{TX}) - 10 \text{DEL\_B}^2 \text{COS}^2(\text{TX}) + 2 \text{DEL\_B}) \text{SIN}(\text{TX}) \text{COS}(\text{TY})) \text{COS}(\text{TZ}) \\
& \text{SIN}^2(\text{TZ}) + (((2 \text{DEL\_B}^3 \text{COS}^3(\text{TX}) - 2 \text{DEL\_B}^5 \text{COS}^5(\text{TX})) \text{COS}^3(\text{TY}) \\
& + (2 \text{DEL\_CP}^3 \text{COS}^3(\text{TX}) - 2 \text{DEL\_CP} \text{COS}(\text{TX})) \text{SIN}(\text{TX}) \text{COS}^2(\text{TY}) \\
& + (6 \text{DEL\_B}^5 \text{COS}^5(\text{TX}) - 9 \text{DEL\_B}^3 \text{COS}^3(\text{TX}) + 3 \text{DEL\_B} \text{COS}(\text{TX})) \text{COS}(\text{TY})) \text{SIN}(\text{TY}) \\
& + (-5 \text{DEL\_BP}^5 \text{COS}^5(\text{TX}) + 7 \text{DEL\_BP}^3 \text{COS}^3(\text{TX}) - 2 \text{DEL\_BP} \text{COS}(\text{TX})) \text{COS}^3(\text{TY}) \\
& + (6 \text{DEL\_BP}^5 \text{COS}^5(\text{TX}) - 9 \text{DEL\_BP}^3 \text{COS}^3(\text{TX}) + 3 \text{DEL\_BP} \text{COS}(\text{TX})) \text{COS}(\text{TY})) \text{SIN}(\text{TZ}) \\
& + ((\text{DEL\_BP}^4 \text{COS}^4(\text{TX}) \text{SIN}(\text{TX}) \text{COS}^3(\text{TY}) + (\text{DEL\_BP}^2 \text{COS}^2(\text{TX}) - 2 \text{DEL\_BP} \text{COS}^4(\text{TX})) \\
& \text{SIN}(\text{TX}) \text{COS}(\text{TY})) \text{SIN}(\text{TY}) + (2 \text{DEL\_B}^4 \text{COS}^4(\text{TX}) - \text{DEL\_B}^2 \text{COS}^2(\text{TX})) \text{SIN}(\text{TX}) \\
& \text{COS}^3(\text{TY}) + (\text{DEL\_B}^2 \text{COS}^2(\text{TX}) - 2 \text{DEL\_B}^4 \text{COS}^4(\text{TX})) \text{SIN}(\text{TX}) \text{COS}(\text{TY})) \text{COS}(\text{TZ}) \\
& / (((\text{LP}^3 \text{COS}^3(\text{TX}) \text{COS}^2(\text{TY}) - 4 \text{LP}^3 \text{COS}^3(\text{TX}) + 3 \text{LP} \text{COS}(\text{TX})) \text{SIN}(\text{TY}) \text{COS}^2(\text{TZ}) \\
& + (\text{LP}^3 \text{COS}^3(\text{TX}) - \text{LP} \text{COS}(\text{TX})) \text{SIN}(\text{TY})) \text{SIN}(\text{TZ}) \\
& + ((4 \text{LP}^2 \text{COS}^2(\text{TX}) - \text{LP}) \text{SIN}(\text{TX}) - 3 \text{LP}^2 \text{COS}^2(\text{TX}) \text{SIN}(\text{TX}) \text{COS}^2(\text{TY})) \text{COS}^3(\text{TZ}) \\
& + (2 \text{LP}^2 \text{COS}^2(\text{TX}) \text{SIN}(\text{TX}) \text{COS}^2(\text{TY}) + (\text{LP} - 3 \text{LP}^2 \text{COS}^2(\text{TX})) \text{SIN}(\text{TX})) \text{COS}(\text{TZ})
\end{aligned}$$

DELTZ=

$$\begin{aligned}
& - ((2 \text{ DEL\_CP} \cos(\text{TX}) \sin^2(\text{TX}) \cos^2(\text{TY}) - 4 \text{ DEL\_CP} \cos(\text{TX}) \sin^2(\text{TX})) \\
& \sin(\text{TY}) + 2 \text{ DEL\_BP} \cos(\text{TX}) \sin^5(\text{TX}) \cos^3(\text{TY}) \\
& - 4 \text{ DEL\_BP} \cos(\text{TX}) \sin^5(\text{TX}) \cos(\text{TY})) \sin^3(\text{TZ}) \\
& + (((\text{DEL\_BP} - 2 \text{ DEL\_BP} \cos^4(\text{TX})) \sin^2(\text{TX}) - \text{DEL\_BP} \sin^4(\text{TX})) \cos^3(\text{TY}) \\
& + (4 \text{ DEL\_BP} \sin^6(\text{TX}) - 2 \text{ DEL\_BP} \sin^4(\text{TX})) \\
& + (4 \text{ DEL\_BP} \cos^4(\text{TX}) - \text{DEL\_BP} \sin^2(\text{TX})) \cos(\text{TY})) \sin(\text{TY}) \\
& + (2 \text{ DEL\_CP} \sin^5(\text{TX}) - 3 \text{ DEL\_CP} \sin^3(\text{TX}) + \text{DEL\_CP} \sin(\text{TX})) \cos^4(\text{TY}) \\
& + (-2 \text{ DEL\_B} \sin^6(\text{TX}) + \text{DEL\_B} \sin^4(\text{TX}) + \text{DEL\_B} \sin^2(\text{TX})) \cos^3(\text{TY}) \\
& + (-4 \text{ DEL\_CP} \sin^5(\text{TX}) + 4 \text{ DEL\_CP} \sin^3(\text{TX}) - \text{DEL\_CP} \sin(\text{TX})) \cos^2(\text{TY}) \\
& + (4 \text{ DEL\_B} \sin^6(\text{TX}) - 2 \text{ DEL\_B} \sin^4(\text{TX}) - \text{DEL\_B} \sin^2(\text{TX})) \cos(\text{TY})) \cos(\text{TZ}) \\
& \sin^2(\text{TZ}) + (((2 \text{ DEL\_B} \cos^3(\text{TX}) \sin^3(\text{TX}) - \text{DEL\_B} \cos^3(\text{TX}) \sin(\text{TX})) \cos^3(\text{TY}) \\
& + (4 \text{ DEL\_CP} \cos(\text{TX}) \sin^2(\text{TX}) - 4 \text{ DEL\_CP} \cos(\text{TX}) \sin^4(\text{TX})) \cos^2(\text{TY}) \\
& + (4 \text{ DEL\_B} \cos(\text{TX}) \sin^5(\text{TX}) - 4 \text{ DEL\_B} \cos^3(\text{TX}) \sin^3(\text{TX}) \\
& + (2 \text{ DEL\_B} \cos^3(\text{TX}) - \text{DEL\_B} \cos(\text{TX})) \sin(\text{TX})) \cos(\text{TY}) \\
& - 4 \text{ DEL\_CP} \cos(\text{TX}) \sin^2(\text{TX})) \sin(\text{TY}) + ((4 \text{ DEL\_BP} \cos^3(\text{TX}) - \text{DEL\_BP} \cos(\text{TX})) \\
& \sin^3(\text{TX}) + (\text{DEL\_BP} \cos(\text{TX}) - 2 \text{ DEL\_BP} \cos^3(\text{TX})) \sin(\text{TX})) \cos^3(\text{TY}) \\
& + ((2 \text{ DEL\_BP} \cos^3(\text{TX}) - \text{DEL\_BP} \cos(\text{TX})) \sin(\text{TX}) - 4 \text{ DEL\_BP} \cos^3(\text{TX}) \sin^3(\text{TX}))
\end{aligned}$$

$$\begin{aligned}
& \cos(TY) \cos^2(TZ) + ((2 \text{ DEL\_CP} \cos(TX) \sin^4(TX) - 3 \text{ DEL\_CP} \cos(TX) \sin^2(TX)) \\
& \cos^2(TY) - 2 \text{ DEL\_B} \cos(TX) \sin^5(TX) \cos(TY) + 4 \text{ DEL\_CP} \cos(TX) \sin^2(TX)) \\
& \sin(TY) - \text{ DEL\_BP} \cos(TX) \sin^5(TX) \cos^3(TY) \\
& + 2 \text{ DEL\_BP} \cos(TX) \sin^5(TX) \cos(TY) \sin(TZ) \\
& + (4 \text{ DEL\_B} \cos^4(TX) \sin^2(TX) \cos^3(TY) - 4 \text{ DEL\_B} \cos^4(TX) \sin^2(TX) \cos(TY)) \\
& \cos^3(TZ) + ((\text{ DEL\_BP} \cos^4(TX) \sin^2(TX) \cos^3(TY) \\
& - 2 \text{ DEL\_BP} \cos^4(TX) \sin^2(TX) \cos(TY)) \sin(TY) \\
& - 2 \text{ DEL\_B} \cos^4(TX) \sin^2(TX) \cos^3(TY) + 2 \text{ DEL\_B} \cos^4(TX) \sin^2(TX) \cos(TY)) \\
& \cos(TZ)) / (((\text{ LP} \cos^4(TX) \cos^3(TY) + (3 \text{ LP} \cos^2(TX) - 4 \text{ LP} \cos^4(TX)) \cos(TY)) \\
& \sin(TY) \cos^2(TZ) + (\text{ LP} \cos^4(TX) - \text{ LP} \cos^2(TX)) \cos(TY) \sin(TY)) \sin(TZ) \\
& + ((4 \text{ LP} \cos^3(TX) - \text{ LP} \cos(TX)) \sin(TX) \cos(TY) \\
& - 3 \text{ LP} \cos^3(TX) \sin(TX) \cos^3(TY)) \cos^3(TZ) \\
& + (2 \text{ LP} \cos^3(TX) \sin(TX) \cos^3(TY) + (\text{ LP} \cos(TX) - 3 \text{ LP} \cos^3(TX)) \sin(TX) \\
& \cos(TY)) \cos(TZ))
\end{aligned}$$

## From NANDV1 MEC

$$\begin{aligned}
\text{DEL\_RX} = & \\
& - (\text{DELTX} (((3 \text{ LP COS}^4(\text{TX}) - 3 \text{ LP COS}^2(\text{TX})) \text{ COS}^3(\text{TY}) \\
& + (6 \text{ LP COS}^2(\text{TX}) - 6 \text{ LP COS}^4(\text{TX})) \text{ COS}^2(\text{TY})) \text{ COS}^2(\text{TZ}) \\
& + (\text{LP COS}^4(\text{TX}) - \text{LP COS}^2(\text{TX})) \text{ COS}^2(\text{TY})) \text{ SIN}(\text{TZ}) \\
& + (6 \text{ LP COS}^3(\text{TX}) - 3 \text{ LP COS}(\text{TX})) \text{ SIN}(\text{TX}) \text{ COS}(\text{TY}) \text{ SIN}(\text{TY}) \text{ COS}^3(\text{TZ}) \\
& + (2 \text{ LP COS}(\text{TX}) - 4 \text{ LP COS}^3(\text{TX})) \text{ SIN}(\text{TX}) \text{ COS}(\text{TY}) \text{ SIN}(\text{TY}) \text{ COS}(\text{TZ})) \\
& + \text{DELTY} ((2 \text{ LP COS}^3(\text{TX}) - \text{LP COS}(\text{TX})) \text{ SIN}(\text{TX}) \text{ COS}^2(\text{TZ}) \text{ SIN}(\text{TZ}) \\
& + ((\text{LP COS}^4(\text{TX}) - \text{LP COS}^2(\text{TX})) \text{ COS}^2(\text{TY}) + 2 \text{ LP COS}^4(\text{TX}) - 2 \text{ LP COS}^2(\text{TX})) \\
& \text{SIN}(\text{TY}) \text{ COS}^3(\text{TZ}) + (\text{LP COS}^2(\text{TX}) - \text{LP COS}^4(\text{TX})) \text{ SIN}(\text{TY}) \text{ COS}(\text{TZ})) \\
& - \text{LP DELTX} \text{ COS}(\text{TY}) \text{ SIN}(\text{TY}) \text{ COS}^2(\text{TZ}) \text{ SIN}(\text{TZ})) / (\text{COS}^2(\text{TX}) \text{ SIN}^2(\text{TX}) \text{ COS}^2(\text{TY}))
\end{aligned}$$

$$\begin{aligned}
\text{DEL\_RY} = & \\
& (\text{DELTY} (((\text{LP COS}^2(\text{TX}) - \text{LP COS}^4(\text{TX})) \text{ SIN}^3(\text{TY}) \\
& + (3 \text{ LP COS}^4(\text{TX}) - 3 \text{ LP COS}^2(\text{TX})) \text{ SIN}(\text{TY})) \text{ SIN}^3(\text{TZ}) \\
& + (\text{LP COS}(\text{TX}) - 2 \text{ LP COS}^3(\text{TX})) \text{ SIN}(\text{TX}) \text{ COS}(\text{TZ}) \text{ SIN}^2(\text{TZ}) \\
& + ((\text{LP COS}^4(\text{TX}) - \text{LP COS}^2(\text{TX})) \text{ SIN}^3(\text{TY}) \\
& + (2 \text{ LP COS}^2(\text{TX}) - 2 \text{ LP COS}^4(\text{TX})) \text{ SIN}(\text{TY})) \text{ SIN}(\text{TZ})) \\
& + \text{DELTX} ((6 \text{ LP COS}^3(\text{TX}) - 3 \text{ LP COS}(\text{TX})) \text{ SIN}(\text{TX}) \text{ COS}(\text{TY}) \text{ SIN}(\text{TY}) \text{ SIN}^3(\text{TZ}) \\
& + ((3 \text{ LP COS}^4(\text{TX}) - 3 \text{ LP COS}^2(\text{TX})) \text{ COS}(\text{TY}) \text{ SIN}^2(\text{TY})
\end{aligned}$$



APPENDIX B. EXPRESSIONS FOR CASE ONE

$$\begin{aligned}
 &+ (3 LP \cos^4(TX) - 3 LP \cos^2(TX)) \cos(TY) \cos(TZ) \sin^2(TZ) \\
 &+ (2 LP \cos(TX) - 4 LP \cos^3(TX)) \sin(TX) \cos(TY) \sin(TY) \sin(TZ) \\
 &+ (LP \cos^2(TX) - LP \cos^4(TX)) \cos(TY) \sin^2(TY) \cos(TZ) \\
 &+ LP \text{ DELTX } \cos(TY) \sin(TY) \cos(TZ) \sin^2(TZ) / (\cos^2(TX) \sin^2(TX) \cos^2(TY))
 \end{aligned}$$

DEL\_RZ=

$$\begin{aligned}
 &- (\text{DELTZ } (2 LP \cos^2(TX) \sin(TY) \cos(TZ) \sin(TZ) \\
 &- 2 LP \cos(TX) \sin(TX) \cos^2(TZ) + LP \cos(TX) \sin(TX)) \\
 &- LP \text{ DELTX } \cos(TZ) \sin(TZ) - LP \text{ DELTY } \cos^2(TX) \cos(TY) \cos^2(TZ)) / \cos^2(TX)
 \end{aligned}$$

R1X=

$$\frac{L LP \cos(TX) \sin(TY) \sin(TZ) - LP \sin(TX) \cos(TZ) + L \sin(TX) \cos(TY)}{2 \sin(TX) \cos(TY)}$$

R1Y=

$$- \frac{W}{2}$$

R1Z=

$$- LL$$

R2X=

$$- \frac{L}{2}$$

R2Y=

$$- \frac{W}{2}$$

$$\begin{aligned}
 R2Z = & \\
 & - (LP \cos(TX) \sin(TX) \sin(TY) \sin^2(TZ)) \\
 & + (- LP \cos^2(TX) \cos^2(TY) + 2 LP \cos^2(TX) - LP) \cos(TZ) \sin(TZ) \\
 & - LP \cos(TX) \sin(TX) \sin(TY) \cos^2(TZ)) / (\cos(TX) \sin(TX) \cos^2(TY)) - LL
 \end{aligned}$$

$$\begin{aligned}
 R3X = & \\
 & \frac{L}{2}
 \end{aligned}$$

$$\begin{aligned}
 R3Y = & \\
 & \frac{W \sin(TX) \cos(TY) W + LP \sin(TX) \sin(TZ) + LP \cos(TX) \sin(TY) \cos(TZ)}{2 \sin(TX) \cos(TY)}
 \end{aligned}$$

$$\begin{aligned}
 R3Z = & \\
 & - LL
 \end{aligned}$$

$$\begin{aligned}
 N1X = & \\
 & 0
 \end{aligned}$$

$$\begin{aligned}
 N1Y = & \\
 & \frac{\sin(TX) \cos(TY)}{\sqrt{(\sin(TX) \sin(TY) \sin(TZ) + \cos(TX) \cos(TZ))^2 + \sin^2(TX) \cos^2(TY)}}
 \end{aligned}$$

$$\begin{aligned}
 N1Z = & \\
 & \frac{\sin(TX) \sin(TY) \sin(TZ) + \cos(TX) \cos(TZ)}{\sqrt{(\sin(TX) \sin(TY) \sin(TZ) + \cos(TX) \cos(TZ))^2 + \sin^2(TX) \cos^2(TY)}}
 \end{aligned}$$

N2X=

$$\frac{\cos(TY) \sin(TZ)}{\sqrt{\cos^2(TY) \sin^2(TZ) + \cos^2(TY) \cos^2(TZ)}}$$

N2Y=

$$\frac{\cos(TY) \cos(TZ)}{\sqrt{\cos^2(TY) \sin^2(TZ) + \cos^2(TY) \cos^2(TZ)}}$$

N2Z=

0

N3X=

$$\frac{\sin(TX) \cos(TY)}{\sqrt{(\sin(TX) \sin(TY) \cos(TZ) - \cos(TX) \sin(TZ))^2 + \sin^2(TX) \cos^2(TY)}}$$

N3Y=

0

N3Z=

$$\frac{\sin(TX) \sin(TY) \cos(TZ) - \cos(TX) \sin(TZ)}{\sqrt{(\sin(TX) \sin(TY) \cos(TZ) - \cos(TX) \sin(TZ))^2 + \sin^2(TX) \cos^2(TY)}}$$

V1X=

$$\begin{aligned} & (AP (\text{DELTZ} (- \sin(TX) \sin(TY) \sin(TZ) - \cos(TX) \cos(TZ))) \\ & + \text{DELTX} \sin(TX) \sin(TZ) + \text{DELTX} \cos(TX) \sin(TY) \cos(TZ) \\ & + \text{DEITY} \sin(TX) \cos(TY) \cos(TZ)) + LP (- \text{DELTZ} \cos(TY) \sin(TZ) \\ & - \text{DEITY} \sin(TY) \cos(TZ)) + \text{DEL\_RX}) / \text{SQRT}(\text{EXPT}(AP \\ & (\text{DELTZ} (- \sin(TX) \sin(TY) \sin(TZ) - \cos(TX) \cos(TZ)) + \text{DELTX} \sin(TX) \sin(TZ) \end{aligned}$$

**APPENDIX B. EXPRESSIONS FOR CASE ONE**

252

$$\begin{aligned}
 &+ \text{DELTX COS(TX) SIN(TY) COS(TZ) + DELTY SIN(TX) COS(TY) COS(TZ))} \\
 &+ \text{LP (- DELTZ COS(TY) SIN(TZ) - DELTY SIN(TY) COS(TZ)) + DEL_RX, 2)} \\
 &+ \text{EXPT(AP (DELTZ (SIN(TX) SIN(TY) COS(TZ) - COS(TX) SIN(TZ))} \\
 &+ \text{DELTX COS(TX) SIN(TY) SIN(TZ) + DELTY SIN(TX) COS(TY) SIN(TZ)} \\
 &- \text{DELTX SIN(TX) COS(TZ)) + LP (DELTZ COS(TY) COS(TZ) - DELTY SIN(TY) SIN(TZ))} \\
 &+ \text{DEL_RY, 2) + EXPT(AP (DELTX COS(TX) COS(TY) - DELTY SIN(TX) SIN(TY))} \\
 &- \text{DELTY LP COS(TY) + DEL_RZ, 2))}
 \end{aligned}$$

V1Y=

$$\begin{aligned}
 &(\text{AP (DELTZ (SIN(TX) SIN(TY) COS(TZ) - COS(TX) SIN(TZ))} \\
 &+ \text{DELTX COS(TX) SIN(TY) SIN(TZ) + DELTY SIN(TX) COS(TY) SIN(TZ)} \\
 &- \text{DELTX SIN(TX) COS(TZ)) + LP (DELTZ COS(TY) COS(TZ) - DELTY SIN(TY) SIN(TZ))} \\
 &+ \text{DEL_RY)/SQRT(EXPT(AP (DELTZ (- SIN(TX) SIN(TY) SIN(TZ) - COS(TX) COS(TZ))} \\
 &+ \text{DELTX SIN(TX) SIN(TZ) + DELTX COS(TX) SIN(TY) COS(TZ)} \\
 &+ \text{DELTY SIN(TX) COS(TY) COS(TZ)) + LP (- DELTZ COS(TY) SIN(TZ)} \\
 &- \text{DELTY SIN(TY) COS(TZ)) + DEL_RX, 2) + EXPT(AP} \\
 &(\text{DELTZ (SIN(TX) SIN(TY) COS(TZ) - COS(TX) SIN(TZ))} \\
 &+ \text{DELTX COS(TX) SIN(TY) SIN(TZ) + DELTY SIN(TX) COS(TY) SIN(TZ)} \\
 &- \text{DELTX SIN(TX) COS(TZ)) + LP (DELTZ COS(TY) COS(TZ) - DELTY SIN(TY) SIN(TZ))} \\
 &+ \text{DEL_RY, 2) + EXPT(AP (DELTX COS(TX) COS(TY) - DELTY SIN(TX) SIN(TY))} \\
 &- \text{DELTY LP COS(TY) + DEL_RZ, 2))}
 \end{aligned}$$

V1Z=

$$\begin{aligned}
 &(\text{AP (DELTX COS(TX) COS(TY) - DELTY SIN(TX) SIN(TY)) - DELTY LP COS(TY)} \\
 &+ \text{DEL_RZ)/SQRT(EXPT(AP (DELTZ (- SIN(TX) SIN(TY) SIN(TZ) - COS(TX) COS(TZ))} \\
 &+ \text{DELTX SIN(TX) SIN(TZ) + DELTX COS(TX) SIN(TY) COS(TZ)} \\
 &+ \text{DELTY SIN(TX) COS(TY) COS(TZ)) + LP (- DELTZ COS(TY) SIN(TZ)} \\
 &- \text{DELTY SIN(TY) COS(TZ)) + DEL_RX, 2) + EXPT(AP} \\
 &(\text{DELTZ (SIN(TX) SIN(TY) COS(TZ) - COS(TX) SIN(TZ))}
 \end{aligned}$$

APPENDIX B. EXPRESSIONS FOR CASE ONE

253

$$\begin{aligned}
 & + \text{DELTX} \cos(\text{TX}) \sin(\text{TY}) \sin(\text{TZ}) + \text{DELTY} \sin(\text{TX}) \cos(\text{TY}) \sin(\text{TZ}) \\
 & - \text{DELTX} \sin(\text{TX}) \cos(\text{TZ}) + \text{LP} (\text{DELTZ} \cos(\text{TY}) \cos(\text{TZ}) - \text{DELTY} \sin(\text{TY}) \sin(\text{TZ})) \\
 & + \text{DEL\_RY, 2}) + \text{EXPT}(\text{AP} (\text{DELTX} \cos(\text{TX}) \cos(\text{TY}) - \text{DELTY} \sin(\text{TX}) \sin(\text{TY})) \\
 & - \text{DELTY} \text{LP} \cos(\text{TY}) + \text{DEL\_RZ, 2}))
 \end{aligned}$$

$$\begin{aligned}
 \text{V2X} = & (\text{BP} (- \text{DELTZ} \cos(\text{TY}) \sin(\text{TZ}) - \text{DELTY} \sin(\text{TY}) \cos(\text{TZ})) + \text{DEL\_RX}) \\
 & / \text{SQRT}((\text{BP} (\text{DELTZ} \cos(\text{TY}) \cos(\text{TZ}) - \text{DELTY} \sin(\text{TY}) \sin(\text{TZ})) + \text{DEL\_RY})^2 \\
 & + (\text{BP} (- \text{DELTZ} \cos(\text{TY}) \sin(\text{TZ}) - \text{DELTY} \sin(\text{TY}) \cos(\text{TZ})) + \text{DEL\_RX})^2 \\
 & + (\text{DEL\_RZ} - \text{BP} \text{DELTY} \cos(\text{TY}))^2)
 \end{aligned}$$

$$\begin{aligned}
 \text{V2Y} = & (\text{BP} (\text{DELTZ} \cos(\text{TY}) \cos(\text{TZ}) - \text{DELTY} \sin(\text{TY}) \sin(\text{TZ})) + \text{DEL\_RY}) \\
 & / \text{SQRT}((\text{BP} (\text{DELTZ} \cos(\text{TY}) \cos(\text{TZ}) - \text{DELTY} \sin(\text{TY}) \sin(\text{TZ})) + \text{DEL\_RY})^2 \\
 & + (\text{BP} (- \text{DELTZ} \cos(\text{TY}) \sin(\text{TZ}) - \text{DELTY} \sin(\text{TY}) \cos(\text{TZ})) + \text{DEL\_RX})^2 \\
 & + (\text{DEL\_RZ} - \text{BP} \text{DELTY} \cos(\text{TY}))^2)
 \end{aligned}$$

$$\begin{aligned}
 \text{V2Z} = & (\text{DEL\_RZ} - \text{BP} \text{DELTY} \cos(\text{TY})) \\
 & / \text{SQRT}((\text{BP} (\text{DELTZ} \cos(\text{TY}) \cos(\text{TZ}) - \text{DELTY} \sin(\text{TY}) \sin(\text{TZ})) + \text{DEL\_RY})^2 \\
 & + (\text{BP} (- \text{DELTZ} \cos(\text{TY}) \sin(\text{TZ}) - \text{DELTY} \sin(\text{TY}) \cos(\text{TZ})) + \text{DEL\_RX})^2 \\
 & + (\text{DEL\_RZ} - \text{BP} \text{DELTY} \cos(\text{TY}))^2)
 \end{aligned}$$

$$\begin{aligned}
 \text{V3X} = & ((\text{DELTZ} (- \sin(\text{TX}) \sin(\text{TY}) \sin(\text{TZ}) - \cos(\text{TX}) \cos(\text{TZ})) \\
 & + \text{DELTX} \sin(\text{TX}) \sin(\text{TZ}) + \text{DELTX} \cos(\text{TX}) \sin(\text{TY}) \cos(\text{TZ}))
 \end{aligned}$$

APPENDIX B. EXPRESSIONS FOR CASE ONE

254

$$\begin{aligned}
 & + \text{DELTY SIN(TX) COS(TY) COS(TZ)) (WP - CP) + DEL\_RX} \\
 & / \text{SQRT(EXPT((DELTZ (- SIN(TX) SIN(TY) SIN(TZ) - COS(TX) COS(TZ))} \\
 & + \text{DELTX SIN(TX) SIN(TZ) + DELTX COS(TX) SIN(TY) COS(TZ)} \\
 & + \text{DELTY SIN(TX) COS(TY) COS(TZ)) (WP - CP) + DEL\_RX, 2)} \\
 & + \text{EXPT((DELTZ (SIN(TX) SIN(TY) COS(TZ) - COS(TX) SIN(TZ))} \\
 & + \text{DELTX COS(TX) SIN(TY) SIN(TZ) + DELTY SIN(TX) COS(TY) SIN(TZ)} \\
 & - \text{DELTX SIN(TX) COS(TZ)) (WP - CP) + DEL\_RY, 2)} \\
 & + ((\text{DELTX COS(TX) COS(TY) - DELTY SIN(TX) SIN(TY)}) (\text{WP - CP} + \text{DEL\_RZ})^2)
 \end{aligned}$$

$$\begin{aligned}
 \text{V3Y=} & ((\text{DELTZ (SIN(TX) SIN(TY) COS(TZ) - COS(TX) SIN(TZ))} \\
 & + \text{DELTX COS(TX) SIN(TY) SIN(TZ) + DELTY SIN(TX) COS(TY) SIN(TZ)} \\
 & - \text{DELTX SIN(TX) COS(TZ)) (WP - CP) + DEL\_RY} \\
 & / \text{SQRT(EXPT((DELTZ (- SIN(TX) SIN(TY) SIN(TZ) - COS(TX) COS(TZ))} \\
 & + \text{DELTX SIN(TX) SIN(TZ) + DELTX COS(TX) SIN(TY) COS(TZ)} \\
 & + \text{DELTY SIN(TX) COS(TY) COS(TZ)) (WP - CP) + DEL\_RX, 2)} \\
 & + \text{EXPT((DELTZ (SIN(TX) SIN(TY) COS(TZ) - COS(TX) SIN(TZ))} \\
 & + \text{DELTX COS(TX) SIN(TY) SIN(TZ) + DELTY SIN(TX) COS(TY) SIN(TZ)} \\
 & - \text{DELTX SIN(TX) COS(TZ)) (WP - CP) + DEL\_RY, 2)} \\
 & + ((\text{DELTX COS(TX) COS(TY) - DELTY SIN(TX) SIN(TY)}) (\text{WP - CP} + \text{DEL\_RZ})^2)
 \end{aligned}$$

$$\begin{aligned}
 \text{V3Z=} & ((\text{DELTX COS(TX) COS(TY) - DELTY SIN(TX) SIN(TY)}) (\text{WP - CP} + \text{DEL\_RZ}) \\
 & / \text{SQRT(EXPT((DELTZ (- SIN(TX) SIN(TY) SIN(TZ) - COS(TX) COS(TZ))} \\
 & + \text{DELTX SIN(TX) SIN(TZ) + DELTX COS(TX) SIN(TY) COS(TZ)} \\
 & + \text{DELTY SIN(TX) COS(TY) COS(TZ)) (WP - CP) + DEL\_RX, 2)} \\
 & + \text{EXPT((DELTZ (SIN(TX) SIN(TY) COS(TZ) - COS(TX) SIN(TZ))} \\
 & + \text{DELTX COS(TX) SIN(TY) SIN(TZ) + DELTY SIN(TX) COS(TY) SIN(TZ)}
 \end{aligned}$$

- DELTX SIN(TX) COS(TZ)) (WP - CP) + DEL\_RY, 2)

+ ((DELTX COS(TX) COS(TY) - DELTY SIN(TX) SIN(TY)) (WP - CP) + DEL\_RZ)<sup>2</sup>)





# Appendix C

## Constraint Surfaces

In this appendix we present the constraint surfaces (cases one through four) used to select the commanded forces and moments for the strategy implemented in Section 4.3.

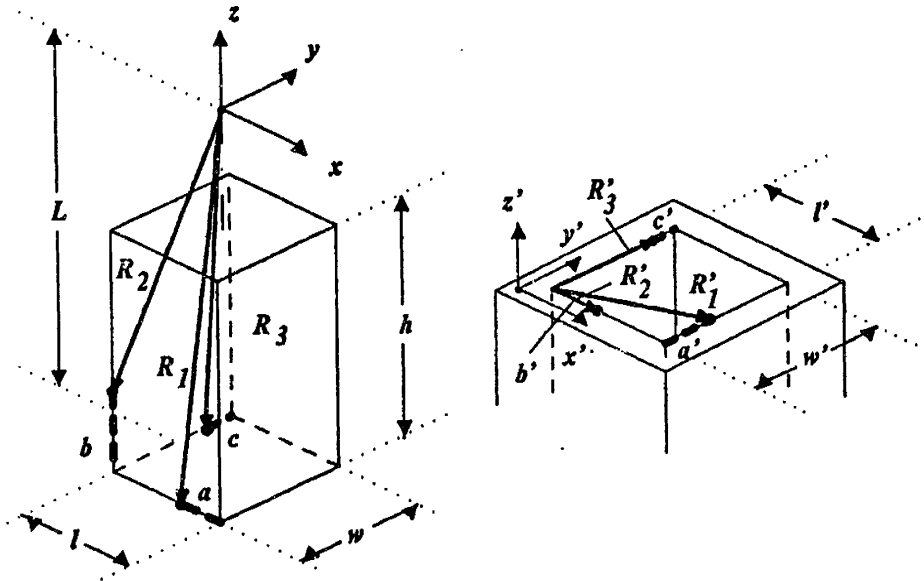


Figure C.1: Parameters for Case One

The state variables for case one are:

- $a = 0.1446$
- $b = 0.2390$
- $c = 0.1621$
- $a' = 0.2666$
- $b' = 0.1560$
- $c' = 0.1478$
- $R_x = -0.7752$
- $R_y = -0.4646$
- $R_z = -1.7386$
- $\dot{b} = -0.3000$
- $\dot{b}' = -0.5000$
- $\dot{c}' = -1.0000$

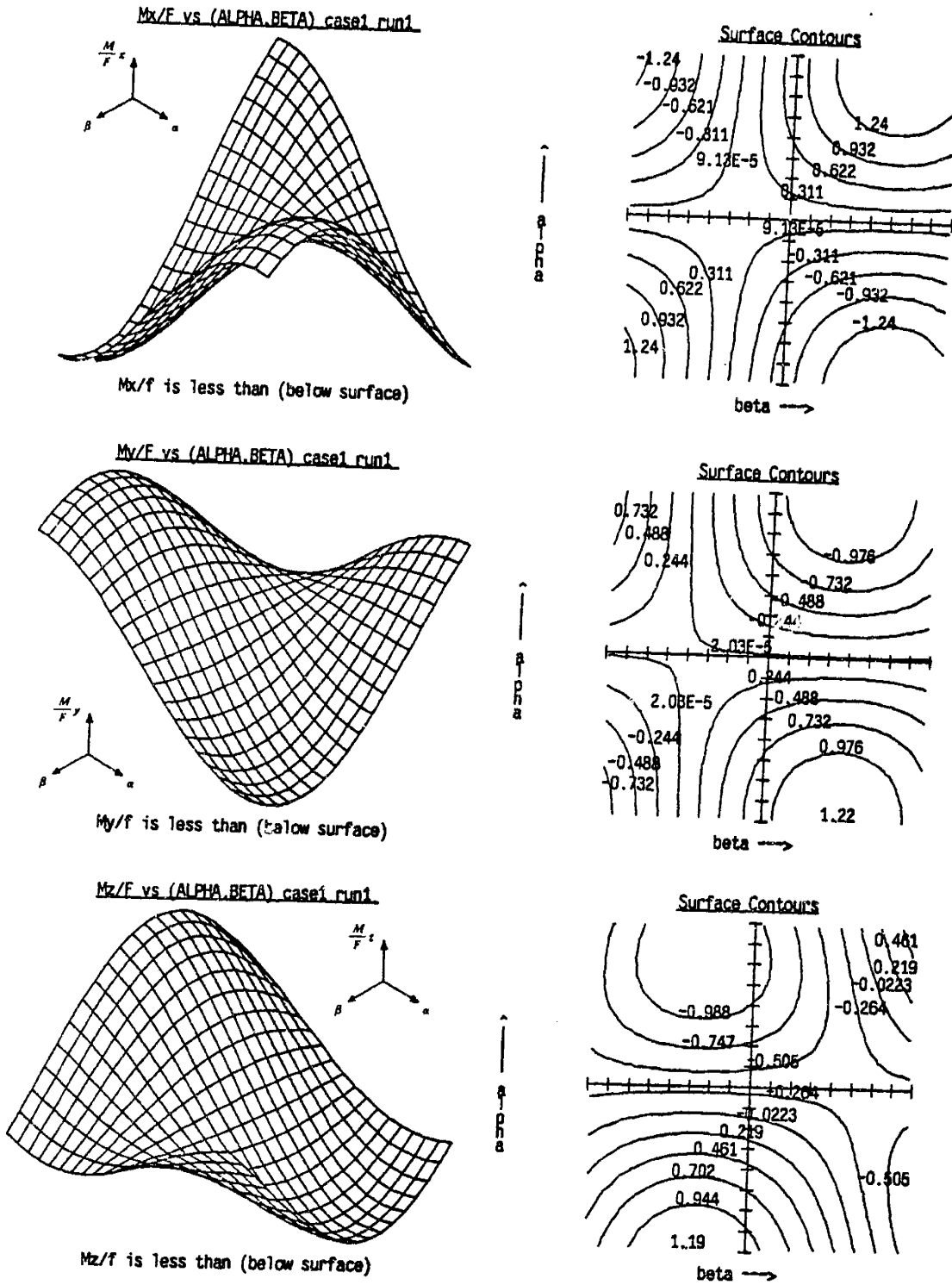


Figure C.2: Sliding Constraint Curves, Case One

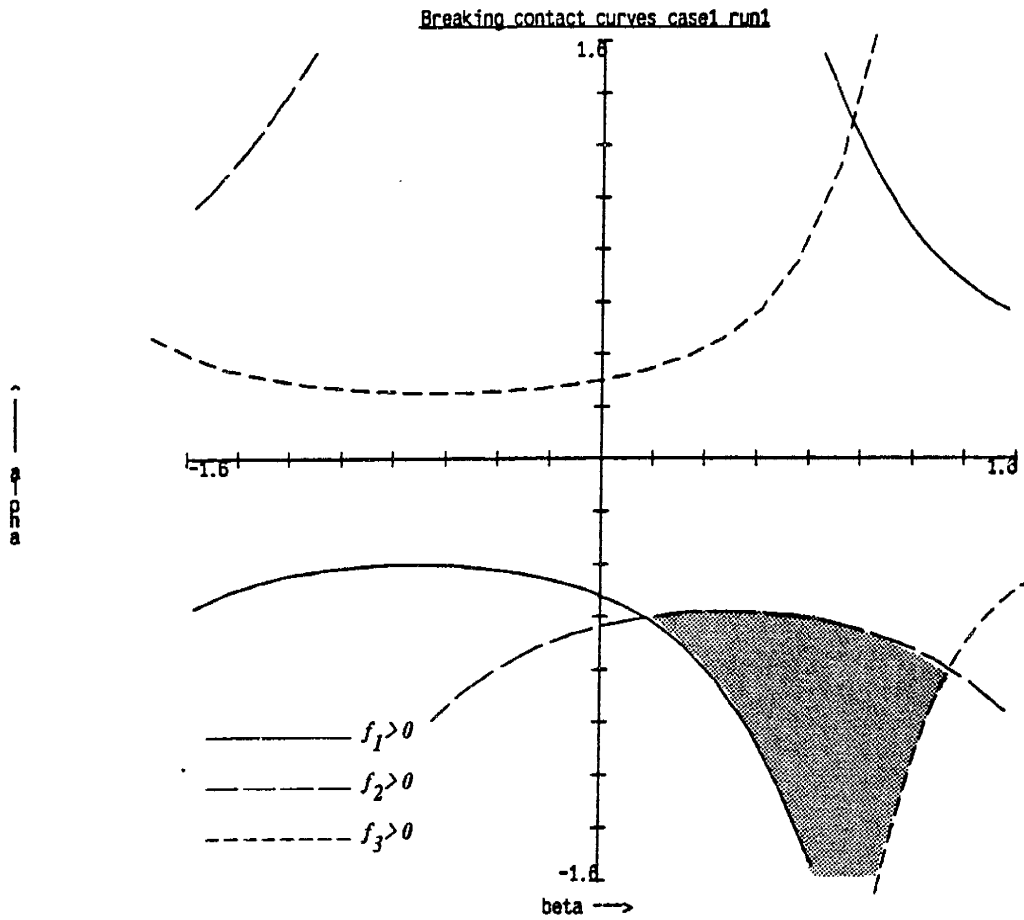


Figure C.3: Breaking Contact Constraint Curves, Case One

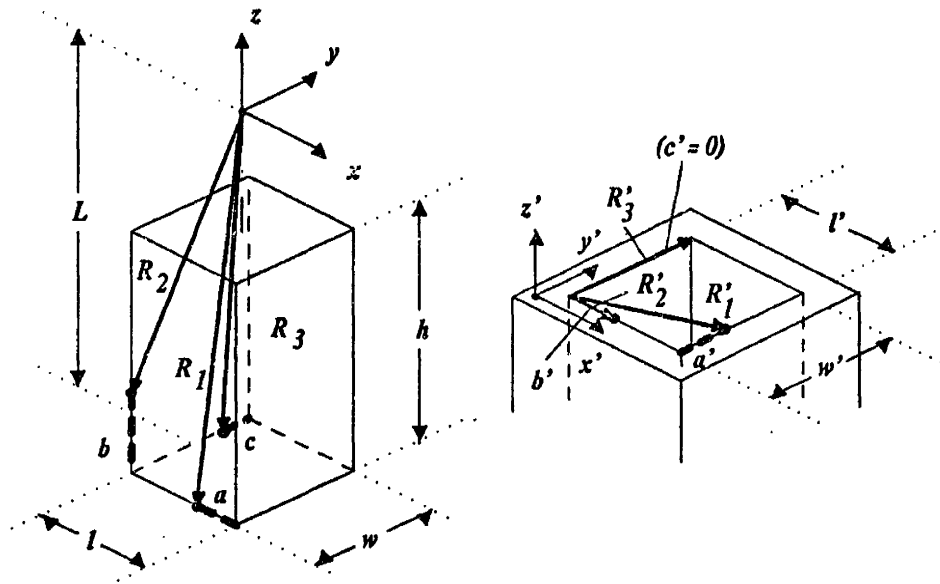


Figure C.4: Parameters for Case Two

The state variables for case two are:

$$\begin{aligned}
 a &= 0.0854 \\
 b &= 0.1949 \\
 c &= 0.0107 \\
 a' &= 0.1333 \\
 b' &= 0.0818 \\
 c' &= 0.0000 \\
 R_x &= -0.7055 \\
 R_y &= -0.4909 \\
 R_z &= -1.7935 \\
 \dot{b} &= -0.5000 \\
 \dot{b}' &= -1.0000 \\
 \phi_3 &= \frac{\pi}{4}
 \end{aligned}$$

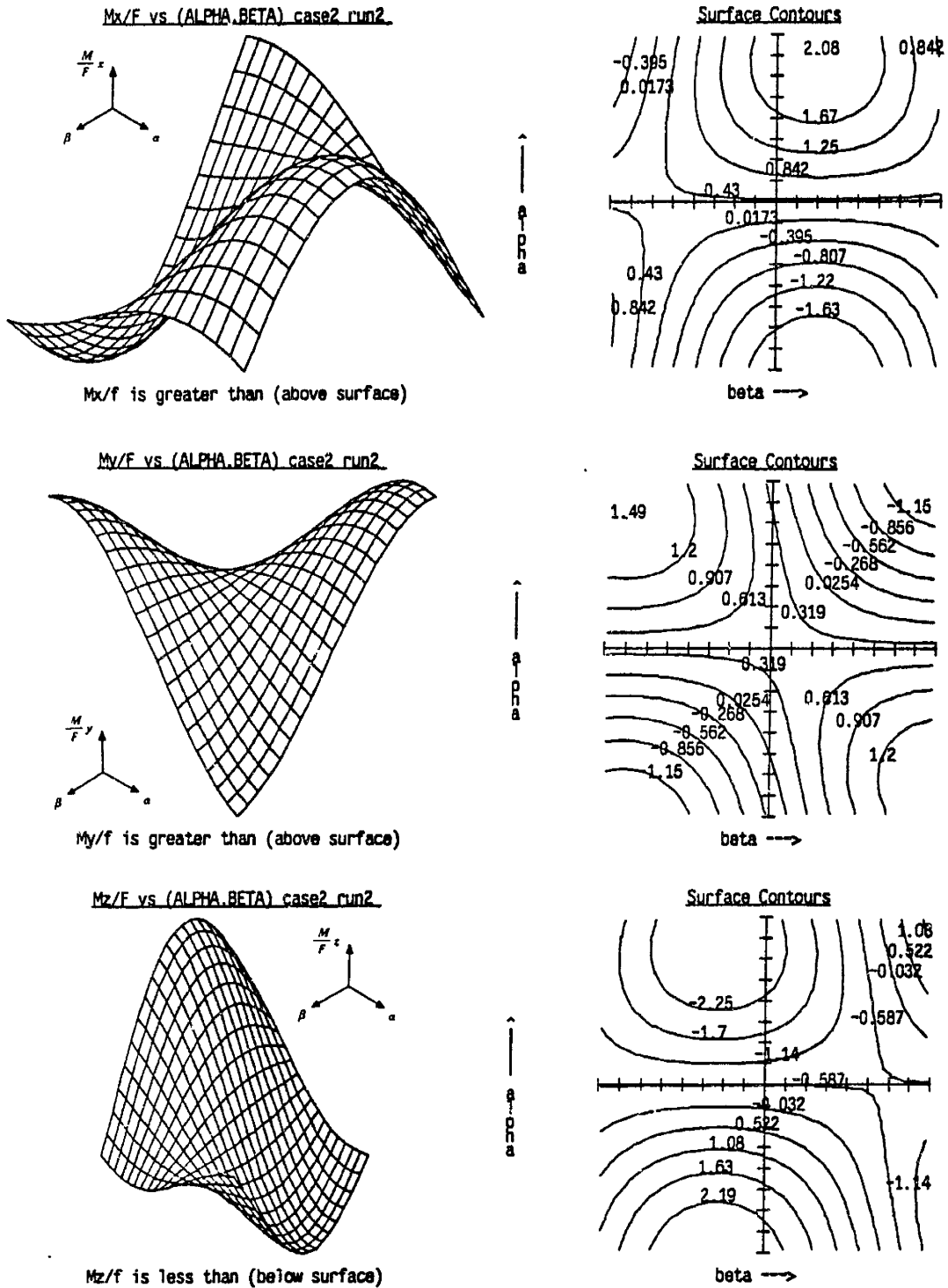


Figure C.5: Sliding Constraint Curves, Case Two

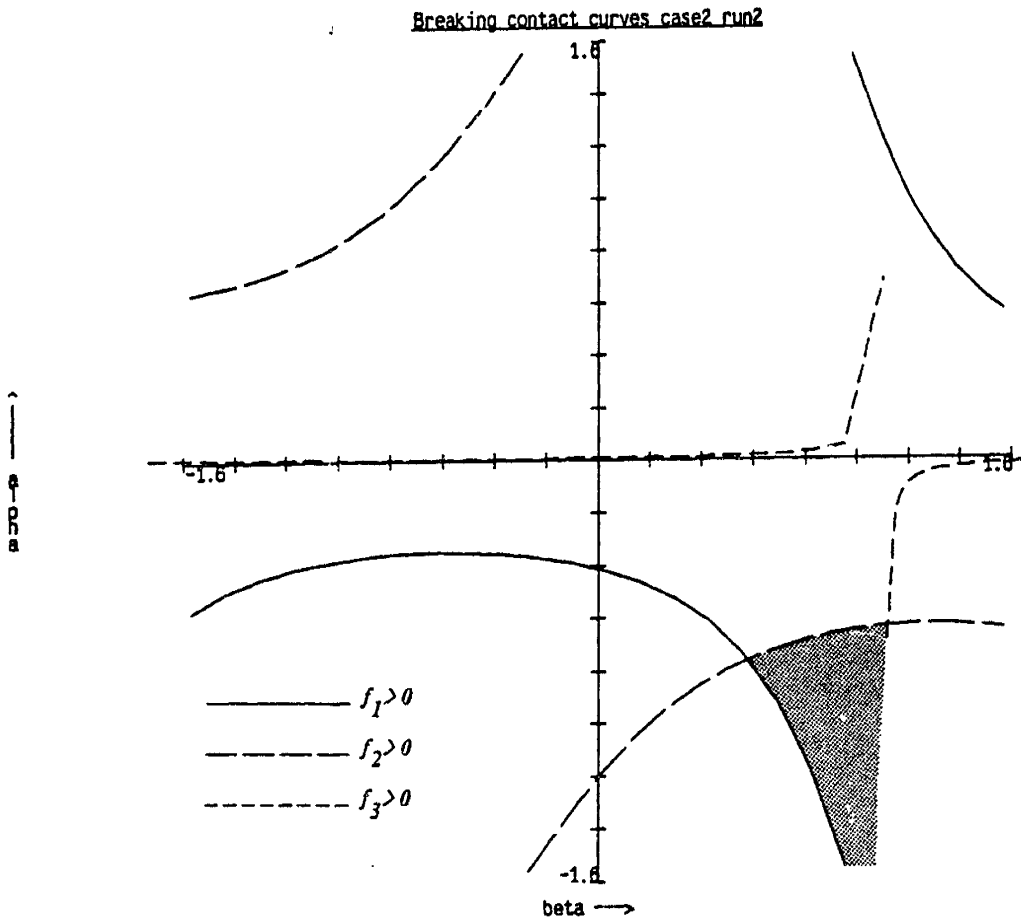


Figure C.6: Breaking Contact Constraint Curves, Case Two

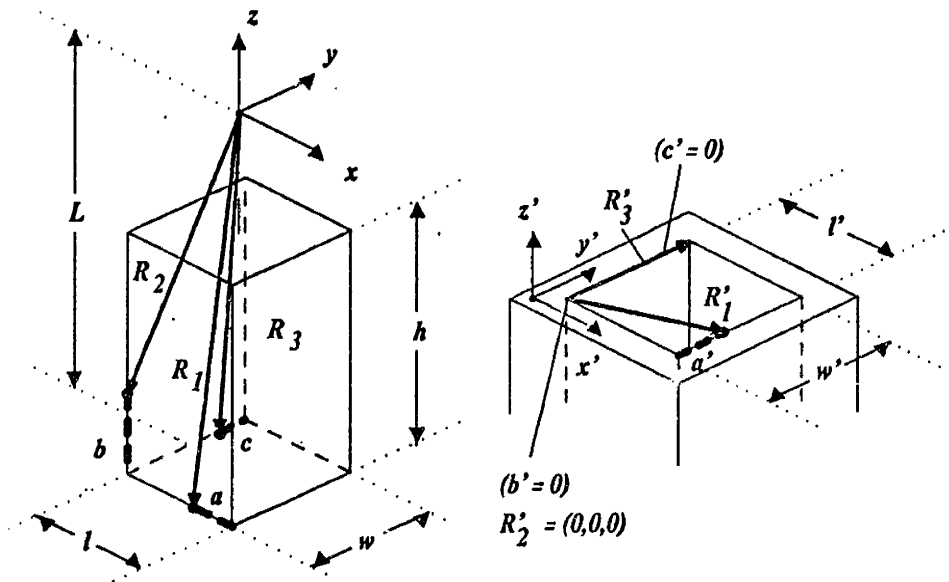


Figure C.7: Parameters for Case Three

The state variables for case three are:

$$\begin{aligned}
 a &= 0.0043 \\
 b &= 0.1540 \\
 c &= 0.0069 \\
 a' &= 0.0236 \\
 b' &= 0.0000 \\
 c' &= 0.0000 \\
 R_x &= -0.6250 \\
 R_y &= -0.5000 \\
 R_z &= -1.8461 \\
 \dot{b} &= -1.0000 \\
 \phi_2 &= \frac{\pi}{4} \\
 \phi_3 &= \frac{\pi}{4}
 \end{aligned}$$



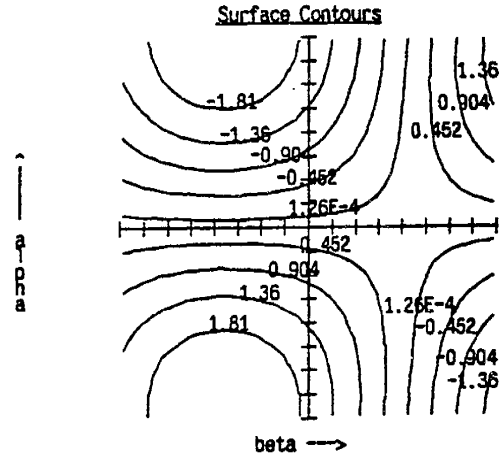
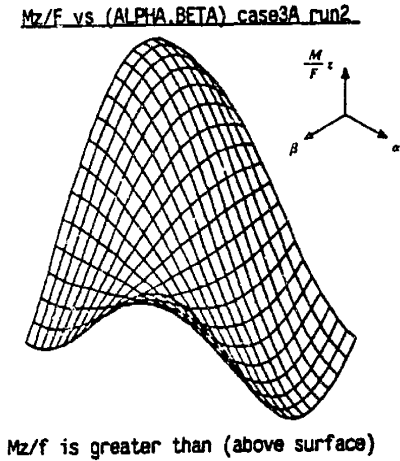
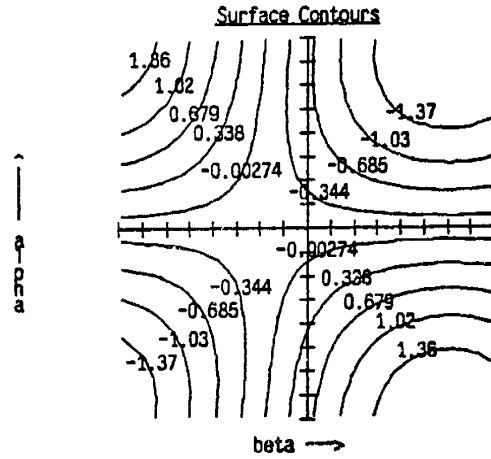
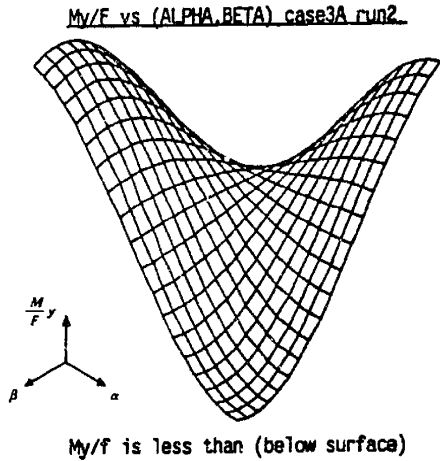
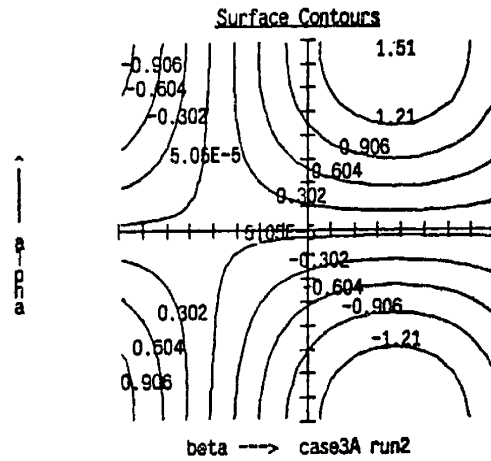
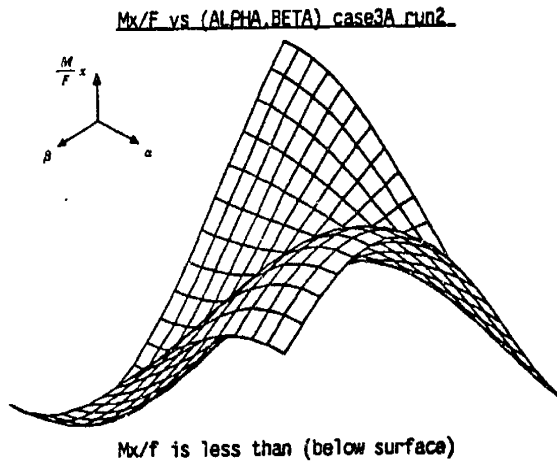


Figure C.8: Sliding Constraint Curves, Case Three

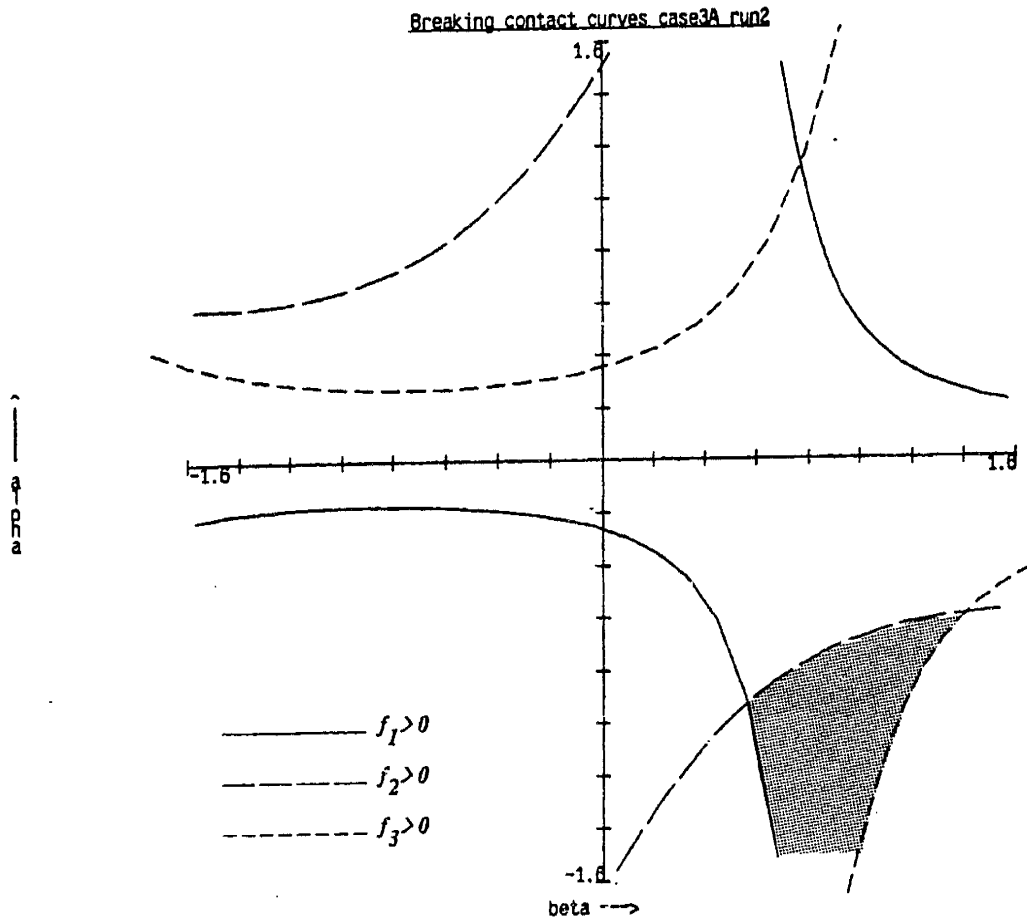


Figure C.9: Breaking Contact Constraint Curves, Case Three

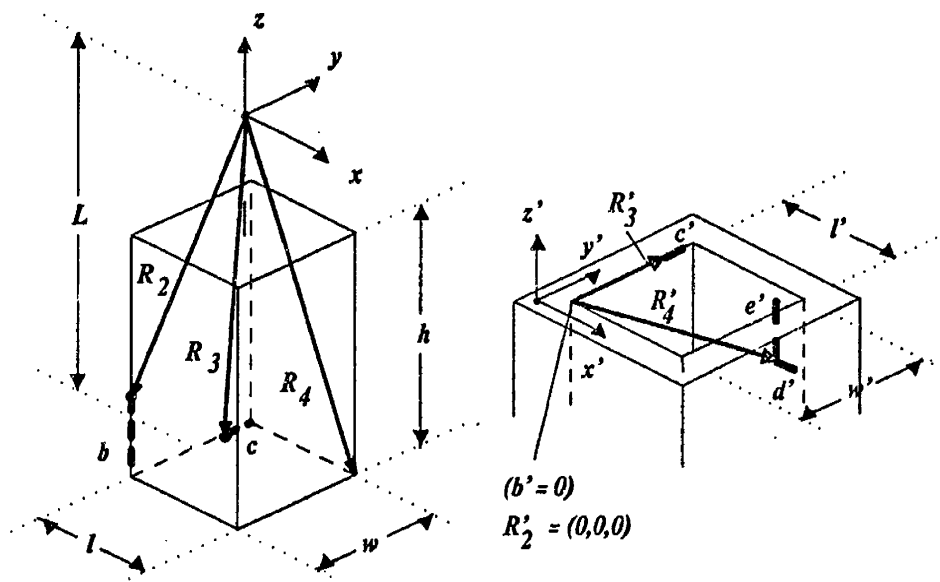


Figure C.10: Parameters for Case Four

The state variables for case four are:

$$\begin{aligned}
 b &= 0.1041 \\
 nc &= 0.0282 \\
 c' &= 0.0021 \\
 d' &= 0.0047 \\
 e' &= 0.0102 \\
 R_x &= -0.6250 \\
 R_y &= -0.5000 \\
 R_z &= -1.8959 \\
 \dot{d}' &= -0.0000 \\
 \dot{e}' &= 1.0000 \\
 \phi_2 &= \frac{\pi}{8}
 \end{aligned}$$

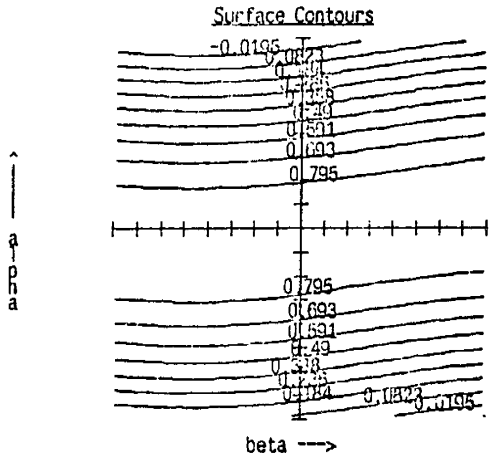
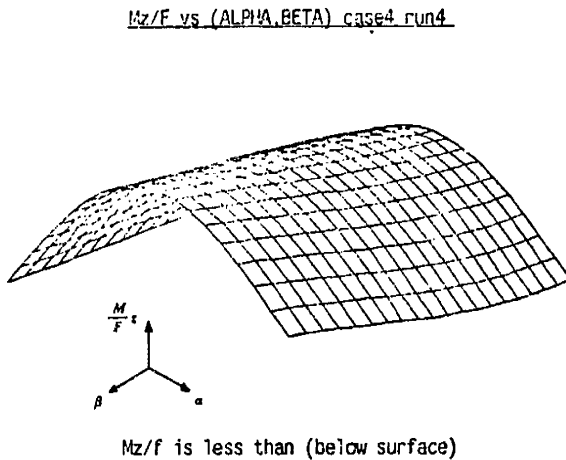
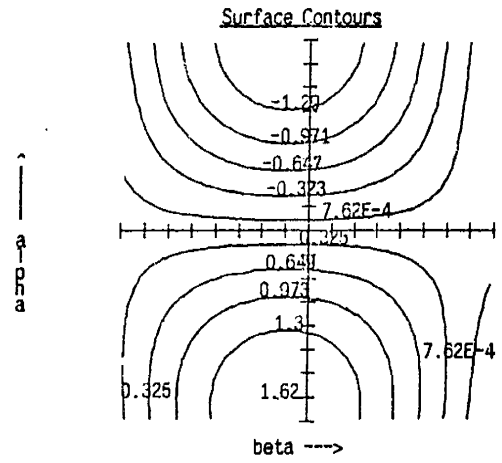
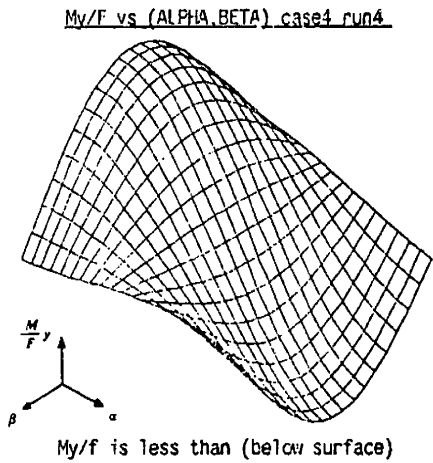
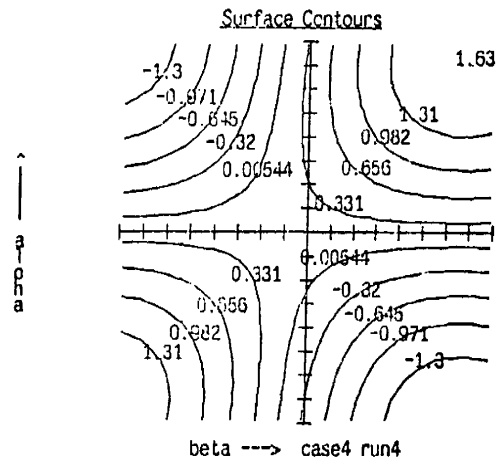
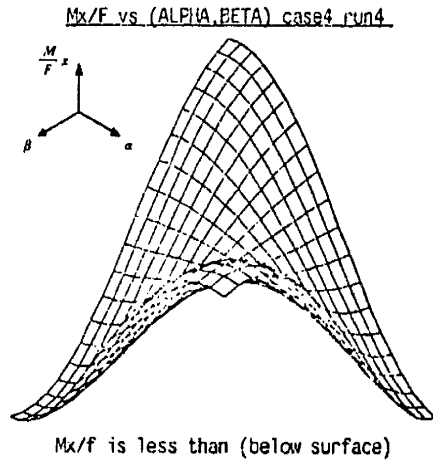


Figure C.11: Sliding Constraint Curves, Case Four

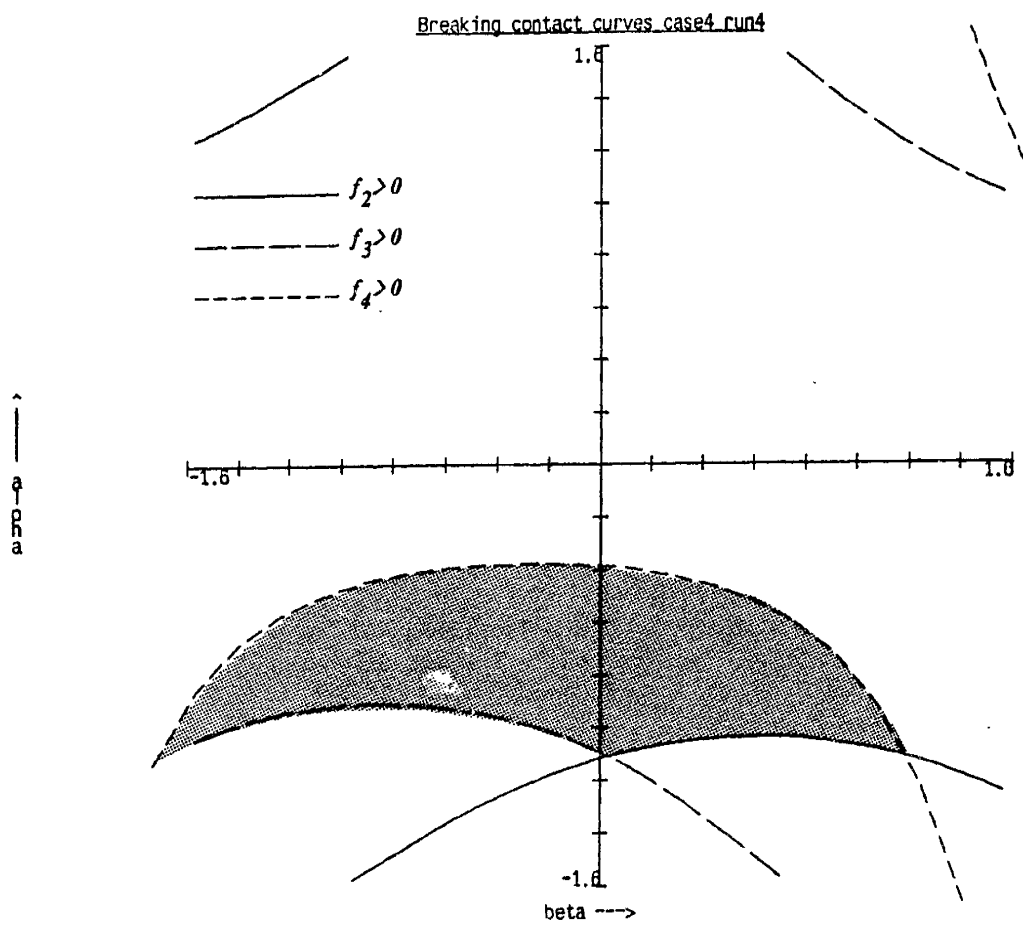


Figure C.12: Breaking Contact Constraint Curves, Case Four

## References

1. Andrews, J. Randolph, "Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator", S.M. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Feb. 1983.
2. Arai, T., and Kinoshita, N., "The Part Mating Forces that Arise When Using a Worktable with Compliance", *Assembly Automation*, Vol. 1, No. 4, Aug. 1981, pp. 204-210.
3. Brooks, Rodney A., "Symbolic Error Analysis and Robot Planning", Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Technical Report 685, Sept. 1982.
4. Benjamin, Michael H., "Design and Analysis of a Control System for the M.I.T. Precision Assembly Robot", S.M. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Jan. 1985.
5. Crandall, Stephen H. et al., *Dynamics of Mechanical and Electromechanical Systems*, Robert E. Krieger Publishing Company, Malabar, FL, 1982.
6. Crochetiere, William J., and Johnson, Timothy L., "Coordinated Compliant Motion", *22nd IEEE Conference on Decision and Control*, 1983.
7. Donald, Bruce R., "Motion Planning with Six Degrees of Freedom", Artificial Intelligence Laboratory, Massachusetts Institute of Technology,

- Technical Report 791, May 1984.
8. **The Charles Stark Draper Laboratory**, *Third Annual Seminar on Advanced Assembly Automation*, Nov. 1982, pp. 8(c).1-8(c).14.
  9. **Erdmann, Michael Andreas**, "On Motion Planning with Uncertainty", Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Technical Report 810, May 1984.
  10. **Giraud, Alain**, "Generalized Active Compliance for Part Mating with Assembly Robots", *Proceedings 1st International Symposium on Robotics Research*, Bretton Woods, NH, MIT Press, Sept. 1984.
  11. **Inoue, Hirochika**, "Force Feedback in Precise Assembly Tasks", Artificial Intelligence Laboratory, Massachusetts Institute of Technology, AI Memo-308, Aug. 1974 (Reprinted in Winston, P. H., and Brown, R. H., eds., *Artificial Intelligence: An MIT Perspective*, MIT Press, 1979).
  12. **Jones, Joe L.**, and **O'Donnell, Patrick A.**, "Using the PUMA System", Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Working Paper 271, April 1985.
  13. **LCS Mathlab Group**, "MACSYMA Reference Manual", Version 10, Volumes I and II, The Mathlab Group, Laboratory for Computer Science, Massachusetts Institute of Technology, 1983.
  14. **Lozano-Pérez, Tomás**, "The Design of a Mechanical Assembly System", Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Technical Report 397, 1976 (Reprinted in part in Winston, P. H., and Brown, R. H., eds., *Artificial Intelligence: An MIT Perspective*, MIT Press, 1979).
  15. **Lozano-Pérez, Tomás**, "Robot Programming", Artificial Intelligence Laboratory, Massachusetts Institute of Technology, AI Memo-698, Dec. 1982.

16. Lozano-Pérez, T., Mason, M. T., and Taylor, R. H., "Automatic Synthesis of Fine-Motion Strategies for Robots", *Proceedings, International Symposium of Robotics Research*, Bretton Woods, NH, MIT Press, Sept. 1984.
17. Mason, Matthew Thomas, "Compliance and Force Control for Computer Controlled Manipulators", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11, No. 6, June 1981 (Reprinted in Brady, M. et al., eds., *Robot Motion*, MIT Press, 1983).
18. Merlet, J-P., "Some Considerations on Feedback-Strategy for Assembly Robot", *5th CISM IFToMM Sym, Ro Man Sy 84, On the Theory and Practice of Robots and Manipulators*, Udine, Italy, 1984.
19. Ohwovoriole, M. S., Hill, J. W., and Roth, B., "On the Theory of Single and Multiple Insertions in Industrial Assemblies", *Proceedings, 10th International Symposium on Industrial Robots*, Milan, Italy, March 1980, pp. 545-558.
20. Ohwovoriole, M. S., and Roth, B., "A Theory of Parts Mating for Assembly Automation", *Proceedings of the Robot and Man Symposium 81*, Warsaw, Poland, Sept. 1981.
21. Paul, Richard P., *Robot Manipulators*, M.I.T. Press, Cambridge, MA, 1981.
22. Popplestone, R.J., Ambler, A. P., and Bellos, I., "RAPT: A Language for Describing Assemblies", *The Industrial Robot*, Sept. 1978.
23. Salisbury, J. Kenneth, "Active Stiffness Control of a Manipulator in Cartesian Coordinates", *19th IEEE Conference on Decision and Control*, Albuquerque, NM, Dec. 1980.
24. Salisbury, J. Kenneth, "Kinematic and Force Analysis of Articulated Hands", Ph.D. Thesis, Department of Mechanical Engineering, Stanford University, May 1982.



25. Salisbury, J. Kenneth, "Interpretation of Contact Geometries from Force Measurements", *Proceedings 1st International Symposium on Robotics Research*, Bretton Woods, NH, MIT Press, Sept. 1984.
26. Seltzer, Donald S., "Tactile Sensory Feedback for Difficult Robot Tasks", The Charles Stark Draper Laboratory, Cambridge, MA, CSDL-P-1432, 1981.
27. Simunovic, Sergio N., "An Information Approach to Parts Mating", Ph.D. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, April 1979.
28. Van Brussel, H., Simons, J., "Automatic Assembly by Active Force Feedback Accommodation", *Proceedings, 8th International Symposium on Industrial Robots*, 1978.
29. Whitney, D. E., "Quasi-Static Assembly of Compliantly Supported Rigid Parts", *Journal of Dynamic Systems, Measurement, and Control*, Vol. 104, March 1982, pp. 65-77.
30. Whitney, Daniel E., "Historical Perspective and State of the Art in Robot Force Control", *IEEE International Conference on Robotics and Automation*, St. Louis, MO, March 1985.