Grasp Stiffness Solutions for Threaded Insertion

By

Edward John Nicolson

B.S. (Princeton University) 1987

THESIS

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

ENGINEERING
ELECTRICAL ENGINEERING AND COMPUTER SCIENCES

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA at BERKELEY

Approved:
Chair: Ronald S. Fearing 12/16/90
Date
John Canny 12/4/90
12/14/90



**********

# Grasp Stiffness Solutions for Threaded Insertion





Edward John Nicolson

## Acknowledgements

I would like to thank everybody who helped me with this project, especially my advisor, Professor Ronald Fearing, with whom I spent many hours discussing threaded fasteners. I would also like to thank Matthew Berkemeier for being an attentive, if not sometimes captive, listener, as well as for providing the code for the integration routine. In addition, I greatly appreciated the comments of my readers, Professor Shankar Sastry and Professor John Canny.

This work was funded in part by: California Microelectronics Fellowship, NSF Grant IRI-8810585, ONR DURIP Grant N00014-89-J-1463, NSF-PYI grant MIP-9057466, and Joint Services Electronics Project, California State Program MICRO.

# Contents

# List of Figures

# Chapter 1

# Introduction

According to a study by Nevins and Whitney in 1980 [Nevins 80] the insertion and tightening of threaded fasteners is one of the twelve most common assembly tasks, yet no published work to date has considered the type of control best suited for the assembly of threaded parts. Maintenance, repair, and assembly tasks require the insertion and tightening of threaded fasteners. If these tasks are to be accomplished by robots a robust method for controlling the parts must be established. The ideal control for insertion of threaded fasteners will be based on force and not position control. We are interested in developing a control that will be robust to uncertainties in initial orientation and position of the parts. Object stiffness control, as Whitney has shown, is ideal for this task.

This thesis investigates a model for the motion of threaded fasteners during mating that allows force control schemes to be tested. As a first step in this investigation the current state of threaded fastener technology in industry was researched. A summary of the findings of this literature search is presented in Chapter 2. It was discovered that little had been done to model the initial mating phase during fastener insertion, so the first part of the modeling phase was to create mathematical equations to describe the surfaces of threaded parts. Once these equations were derived, the configuration space during mating could be analyzed. This geometric analysis is presented in Chapter 3. Given the geometry the next question was how to best model the motion during insertion. Quasistatic analysis, in combination with the minimum power principle, has been shown to be advantageous for this type of low velocity, low inertia problem. It was discovered, however, that the motion of the bolt would not allow for the necessary assumptions, hence a full dynamic model for motion under contact was developed. Chapter 4 gives the derivations for this model. A

C language program was written to simulate the dynamic motion of the bolt under grasp stiffness control. Chapter 5 discusses the general outline of this program along with results of simulations of various grasp stiffness control schemes proposed for threaded fasteners.

At the outset of this research we did not appreciate the complexity of the problem. In robotics literature screw threading is often referred to as a typical task, yet, unlike the smooth peg-in-hole problem, a robust control solution for inserting threaded fasteners has not been presented. Recently Tao et. al. [Tao 90] implemented a bolt threading operation with generalized stiffness controlled manipulators. Tao chose to emulate a Remote Center of Compliance (RCC) type device. Whitney and his colleagues at the Draper Laboratories [Nevins 80, Whitney 82, Drake 78] have shown this type of stiffness is best suited for the peg-in-hole problem. Tao discovered that the stiffness worked, but did not show why or under what conditions. In [Loncaric 87] a stiffness solution for threaded fasteners is presented as an example of a stiffness control scheme. In this thesis the proposed control is simulated. We have discovered that the mating problem is in fact quite difficult when orientation errors exist. Weak spring solutions like those for the RCC do not work. The simulation as well as an implementation on a robot system show this to be the case. In essence, if the RCC were considered to be a stab le control solution for the peg-in-hole problem, a similar stabilizing solution for threaded fasteners may be difficult to find.

The results of this thesis can be used in a number of ways. First of all it can be used to simulate various control strategies. Secondly if one were interested in redesigning the chamfer of threaded fasteners to ease insertion, the new design could be simulated. Finally the simulation method could be generalized to cover other part mating problems and allow for the investigation of control schemes for other fastener types.

# Chapter 2

# Screw Threading Technology

This chapter provides the reader with a background in present threaded fastener technology. Types of threaded fasteners, along with methods for inserting, tightening, and sensing improper insertion are discussed. This information was compiled and summarized from [Allen 88, Bickford 81, Blake 86, Breitinger 82, Davern 85, Fisher 74, Jerney 84, Oberg 88, Smith 80, Treer 79]. The reader will discover that these methods are not easily adaptable to robotics. The remaining chapters will present a method that can be used in robotics, however this chapter indicates situations in which the proposed robotic methods might break down.

## 2.1 Threaded Fasteners in Industry

Of the fastening methods used in the assembly of finished goods, threaded fasteners provide the biggest challenge for fully automated assembly. Alternative methods such as welding, adhesion, riveting, and press-fit are used when it is not necessary to be able to take apart the assembly in the future. These alternative forms of fastening are more easily accomplished automatically than is threaded fastening. One reason is that little is known about the characteristics of the internal and external thread interface during the initial insertion period. Most of the research on threaded fasteners has concentrated on their ability to achieve a desired preload, or joining force.

### 2.1.1 Types of Threaded Fasteners

Threaded fasteners are divided into two main groups: screws and bolts. The term *threaded fastener* refers to the large class of cylindrical (but not necessarily circular cross-section) objects around which a thread has been cut and a head has been attached. If the fastener is to be inserted into a threaded hole it is, in most cases, called a *bolt*. One exception is the Lag-Bolt which is inserted into wood. When the fastener is inserted into a blank hole (i.e. a cylindrical hole with a circular cross-section and no threading) then it is often called a *screw*. According to [Blake 86], *bolt* refers to the whole class of threaded fasteners, but I shall stick to the above distinction for the purposes of this thesis.

Screws and bolts are used for different types of assembly. Often bolts are used for parts that require a high clamping force. Obvious examples of this are engine components. Screws are often used for the assembly of body work, in the case of cars, or the bodies of things like small appliances. Currently many manufacturers are trying to replace screws with integral or press-fit fasteners, which are easier to use and cheaper to assemble [Warnecke 82].

A 1982 report by the Braun company indicates that in a kitchen appliance assembly factory there are more than 200 different kinds of threaded fastener being used [Breitinger 82]. These vary in the following manner:

| | |
|---|---|
| Diameter (1.4 - 4.8 mm) | Length (3 - 25 mm) |
| Threads (6 types) | Head Shape (6 types) |
| Driving Surface (3 types) | Metal and Surface Treatment (7 types) |

It can be expensive to maintain tools and inventory for the many different types of fastener, so it is often recommended to keep variations at a minimum. A general theory for threaded fastener insertion that would allow for all these types is desirable.

### 2.1.2 Starting Threaded Fasteners

As was stated earlier, there is little documentation on deterministic methods for the starting of threaded fasteners. In some cases bolts may be started by hand and then tightened with a machine, or they may be driven in entirely by machine, but without a guaranteed proper threading. Often when bolts are started manually heuristics are used to determine when the bolt has been properly started. A common heuristic is to rotate the fastener the wrong way for half of a turn and then start rotating in the correct direction.

Another form of this method, which requires a high degree of tactile sensitivity, is to rotate the bolt the wrong way until the bolt is felt to drop slightly, which indicates that the starting point for the two threads is lined up. In both of these cases, the idea is essentially to match the starting points of the two threads before the actual insertion is attempted.

Self tapping screws are simpler to start since one need not worry about matching the threads, however one must be sure that the screw is kept aligned with the hole. The Braun report indicates that 20% of their screws are driven in fully automatically with the remaining 80% driven in with hand-held motor driven screwdrivers.

### 2.1.3 Bolt Tensioning

In the automotive industry quality control engineers are concerned with the quality of clamping provided by bolts in the assembly of engines. In many cases product specifications require a certain clamping force, or preload, between parts of the engine. The tension of the bolts holding this assembly together is equal to the clamping force. The method used to determine when a bolt has been tightened to provide the desired clamping force is usually based in some way on the torque applied about the head of the bolt.

#### Determining the Tightening Torque

The torque required to provide a given preload, normally called the *tightening torque*, depends on the geometry of the fastener and the coefficient of friction for the fastener thread surface. Often lubricants are used to vary the friction coefficient. Usually the following formula is used:

$$M_t = K D F_i$$

where $D$ is the nominal fastener diameter, $K$ is a constant depending on the frictional surfaces and the thread geometry, $F_i$ is the desired preload, and $M_t$ is the tightening torque. $K$ is typically 0.20 (the value for unlubricated steel) [Blake 86, Page 155]. The importance of this relation is the approximately linear relationship between tightening torque and preload.

#### Impact Wrenches

Impact wrenches, commonly known as crescent wrenches, are cheap and provide a simple means of bolt tensioning. Use of them in a manufacturing environment is not

advisable, however, when it is important to maintain a consistent tensioning across a group of bolts. An example is the assembly of a double U-clamp. Two bolts secure two semicircular clamps together so that they form a circle, in theory. In practice, if the bolts are not evenly tensioned the interior area will not form a circle, giving an undesirable assembly.

Consistent tensioning is necessary, but impact wrenches satisfy this constraint poorly due to the tension measuring methods. The standard method for measuring tension is with a torque wrench. Unfortunately the torque wrench is used *after* the bolt has been tightened. This means rotation about the axis of the screw is halted while the impact wrench is removed and the torque wrench is attached. The subsequent introduction of the "slip-stick" friction (or static friction) variable gives rise to a tensioning scatter of ± 25% or more [Smith 80]. The skill of the person using the tool goes a long way in determining the accuracy.

### Stall Torquing Devices

A stall-torquing device will thread a bolt into a hole until a certain limiting torque is reached, thus removing the problem of slip-stick friction encountered in the previous example. The limiting torque may be sensed by mechanical means (such as a spring mechanism) or by an electronic transducer. The advantage with the latter is that it has greater repeatability. Unfortunately there are still problems with this method that will cause it to give at best a torque repeatability of ± 10%. Variations in fastener hardness, friction at the head of the fastener and the part, and friction of the threads caused by surface condition, lubricants, temperature, batch, corrosion, and non-uniformly applied coatings all cause a torque limiting method to vary in actual fastener tension.

### Turn of the Nut

This method is similar to the previous in that the bolt is initially threaded to threshold level, except that in this case the threshold is just sufficient to seat and mate the parts. The bolt is then rotated an additional fixed number of degrees. The number is predetermined experimentally. Ideally, with uniform bolts, this will produce uniform tension as stress and fastener rotation are directly proportional. Often this method is preferable when precision fasteners are involved. One immediately apparent problem occurs when the parts do not seat properly during the initial tensioning. This will give rise to a

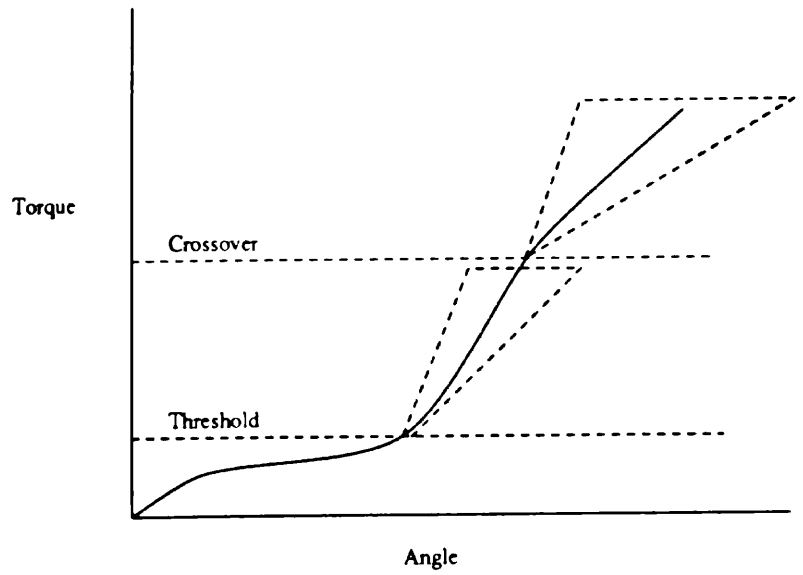


Figure 2.1: Fastening signature for bolt insertion

highly unacceptable join. If this method is used with assemblies involving gaskets, then, in addition, the gaskets must uniformly collapse.

### Torque Rate and Angle Monitoring

This method involves monitoring of both the angle of rotation of the fastener and the applied torque. The critical quantity is the change in torque rate as it is proportional to the degree of fastener rotation. This method allows the fastener to be taken to its yield point giving its maximum possible performance. Additionally the *fastening signature* (see Figure 2.1, adapted from [Smith 80]) gives an indication of the quality of the fit. In the figure *threshold* refers to the lower limit of the torque when it begins to tighten. The *crossover point*, or point of inflection on the curve, is the point at which the bolt begins to plastically yield.

It is possible to divide the fastening signature into two separate phases. These are indicated by the two triangles in the figure. The first triangle encloses the curve until the crossover point is reached. The second triangle encloses the curve from the crossover point. If the curve stays within these two triangles then one can conclude that the bolt has been tensioned properly. In fact this method can detect all of the following problems that are encountered in bolt tensioning [Smith 80]:

| | | |
|---|---|---|
| Faulty Tool | Transducer Failure | Stripped Threads |
| Burrs/Chips | Foreign Material | Crossed Threads |

*Chips* or *burrs* left in the bottom of a tapped hole would give a torque limiting device a false indication of proper fastening. Using torque rate and angle monitoring this problem becomes evident when the fastening signature becomes vertical, indicating a very large increase in torque to rotate the fastener a small amount. This occurs as the bolt has bottomed out.

A *crossed-thread* occurs when the first thread of a nut is chewed out, but the bolt continues to be inserted, and follows a normal signature thereafter. This can be detected by the presence of a dip in the signature curve. The dip will occur after the torque surpasses the threshold value for the first time and indicates a decrease in the torque per angle increment relation after the first thread is chewed out. Crossed threads are undesirable as it causes a weak join.

*Excessive yielding* is caused by soft bolts and may be detected by a leveling off of the fastening signature.

### 2.1.4 Insertion of Self-Tapping Screws

The criteria for proper insertion of self tapping screws is similar to that for bolt tensioning, except that the term *binding force* is often used instead of clamping force. This is due to the type of assembly involved. Often self tapping screws are used in plastics and other deformable materials. It is advantageous to use the combination of metal screws in plastic to reduce rattling. The binding force is achieved through screw tension which is proportional to the torque applied about the head of the screw. The following terms are often used when describing the screw insertion:

- Tapping torque: The maximum torque required to cut the threads
- Stripping torque: The lower limit of the torque required to strip the threads
- Tightening torque: The torque applied when the desired binding force is reached

The fastening signature for proper screw insertion is similar to that for bolt insertion, except that the torque per degree of rotation drops off sharply after the threads are stripped.

According to specifications by Braun the ratio of the stripping torque to tapping torque must be greater than or equal to 1.7 for the 11th insertion of the screw for a given screw and hole combination to be considered reliable. This is termed the *10 times screwing*

*in requirement.* The point of this specification seems to be an acknowledgement that assemblies using this fastening technique can be taken apart, but not too often. The number 10 was apparently chosen as being the number of times an assembly must be reassembled.

Common problems with the insertion of self-tapping screws are analogous to those with bolts: angled screws, reproducibility, and surface damage. If the screw is started at an angle then the tightening torque will be reached before the binding force is achieved. If the tightening torque is not reproducible then there will not be uniform reliability of assembly. If the driving surfaces are damaged by the torques applied then proper tensioning will not occur. It is important to note that the first problem listed above can be alleviated when completely automating screw insertion if a proper length/diameter ratio for the screw is used.

## 2.2 Implications for Robotic Solutions

The previous material has pointed out the problems that can occur during fastening. In order to accomplish fully-autonomous robotic-fastening the following points should be considered:

- There is currently no reliable method for the automatic starting of bolts.
- There are many different types of threaded fastener. An automatic solution should handle as wide a range as possible.
- Regrasping of the fastener cannot be permitted during the tensioning phase due to the slip-stick friction involved.
- Close monitoring of the applied torque and rotation angle will provide valuable information on the progress of the fastening.
- The initial insertion of self-tapping screws does not require alignment of the threads as for the case of bolts and nuts. However, subsequent insertions may strip the threads. For this reason self-tapping screws are likely to be replaced with snap fit assemblies where possible.
- Even if the threading problem is solved for idealized fasteners, how should a system deal with things like chips and burrs? Any solution proposed should be robust to such factors.

Allen [Allen 88] reports that faulty threaded fasteners have caused problems in many industries, especially aerospace. Manufacturers would like to replace them with something easier to assemble, but continuously variable joint tension cannot be accomplished by any other fastener. Over the years small steps forward have been made to apply modern technology to the insertion operation in order to automate it, but these have yet to become 100% reliable. This thesis presents an analysis of threaded fasteners and models force based control solutions for insertion.

# Chapter 3

# The Geometry of Threaded Parts

In this chapter a mathematical description of a screw thread is developed. This model is used to analyze the contact configurations that occur during insertion, derive motion constraint equations, and determine the evolution of the contact points on the surfaces of the nut and bolt. It is important to note that a rigid body model is given. Fasteners do undergo a degree of deformation when they are tightened and when cross threading occurs. These effects are not modeled.

## 3.1 Terminology

The discussion of threaded fasteners is facilitated with the introduction of some basic terms. The following description is a brief summary from [Blake 86] and [Bickford 81]. Some of the terms are shown in Figure 3.2. A full list of terms used for threaded fasteners is given in Appendix B.

A *screw thread* is a ridge of uniform section in the form of a helix. The *external screw thread* is the thread on the bolt and the *internal screw thread* is that on a nut. The *thread profile* refers to the configuration of the thread in the axial plane. The *root* of the profile is at the smallest diameter and the *crest* is at the largest. Note that for the nut the largest diameter, or *internal thread major diameter,* is at the root, not at the crest. The *flank* is the straight part of the thread joining the roots and the crests. The *flank angle*, which in most cases is 30 degrees, is the angle between the flank and the axis perpendicular to the thread. If the thread were extended to a full V (see Figure 3.2) the *fundamental triangle height* would be reached. Instead it is rounded off or flattened at the roots and

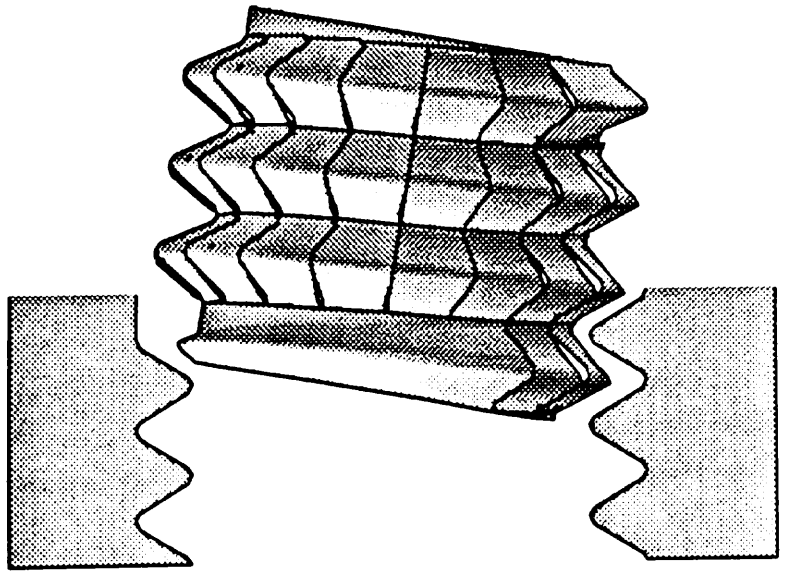

Figure 3.1: A cross-threaded bolt and nut configuration.

crests.

The *pitch* is the distance measured parallel to the thread axis between corresponding points of the thread. Pitch is often described as coarse or fine. A *clearance fit* provides free running assembly by the means of a non zero *allowance*. Allowance is the amount by which the external thread diameter is decreased as compared to the internal thread. This thesis will discusses the *allowance ratio* which expresses allowance as a fraction of the internal thread major diameter.

Threads do not start immediately on a but or bolt, but undergo a thread run-up, also called an *incomplete thread*. The form and length of the run-up plays an important role in the avoidance of *cross-threading*. Cross-threading, which leads to an incomplete and wedged assembly, occurs when the first external thread crosses the internal thread in such a way that the thread engaged on one side of the internal thread is not on the same revolution as the thread engaged on the opposite side. Figure 3.1 shows a bolt in a crossed thread configuration. Our goal is to describe grasp stiffnesses that will avoid cross-threading during threaded insertion.

## 3.2 Functional Description of a Screw Thread

Threaded parts may be described as generalized cylinders with a straight axis. The functions in the following sections are used to create generalized cylinders for a bolt

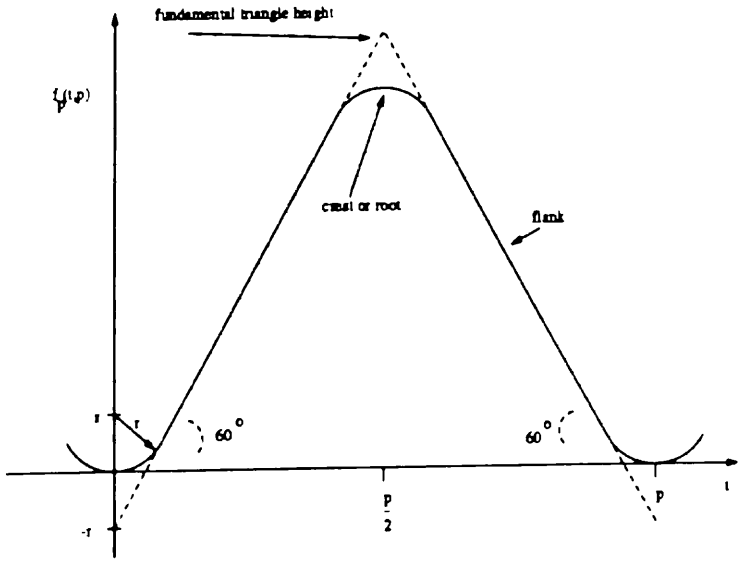


Figure 3.2: Thread profile.

and a nut.

### 3.2.1 Thread Profile

Based on the definitions given in chapter 1, a rounded crest and root thread profile can be made with the following four variables.

$p$: pitch.

$d$: internal thread basic major diameter.

$a$: allowance ratio, $0 \leq a \leq 1$, where the actual allowance is $a * d$

$r$: root and crest radius. In the simulations referred to in this paper $r = \frac{p}{10}$.

The ratio $\frac{d}{p}$ determines if the bolt has fine or coarse threads. Typically $(\frac{d}{p})_{fine} = 2(\frac{d}{p})_{coarse}$.

The thread profile can be parameterized by the following function of position, $t$ and thread pitch $p$. Figure 3.2 shows the function with $p = 1$ and $r = \frac{p}{10}$. Note that $f_p$ and $\dot{f}_p$ are continuous functions. In this function $t$ is understood to be $t$ mod $p$.

$$f_p(t,p) = \begin{cases} r - \sqrt{r^2 - t^2} & 0 \le t \le \frac{\sqrt{3}}{2}r \\ \sqrt{3}t - r & \frac{\sqrt{3}}{2}r \le t \le \frac{p}{2} - \frac{\sqrt{3}}{2}r \\ \frac{\sqrt{3}}{2}p - 3r + \sqrt{r^2 - (t - \frac{p}{2})^2} & \frac{p}{2} - \frac{\sqrt{3}}{2}r \le t \le \frac{p}{2} + \frac{\sqrt{3}}{2}r \\ \sqrt{3}(p - t) - r & \frac{p}{2} + \frac{\sqrt{3}}{2}r \le t \le p - \frac{\sqrt{3}}{2}r \\ r - \sqrt{r^2 - (t - p)^2} & p - \frac{\sqrt{3}}{2}r \le t \le p \end{cases}$$

### 3.2.2 The Screw Thread as a Generalized Cylinder

If you extend the thread profile for an integral number of threads and align the t-axis with the z-axis in a right hand coordinate system, a screw thread can be created by rotating the profile about the z-axis at a radius $\frac{d}{2}$ and vertically shifting at the same time. The vertical shift is $p(\frac{\theta}{2\pi})$ where $\theta$ is the amount of rotation.

A surface in $\Re^3$ may be parameterized by 2 variables. The parameters $z$ and $\theta$ have been chosen. Thus a point on the surface of the nut is given by the cylindrical coordinates: $(r_{int}(z,\theta),\theta,z)$. Similarly a point on the surface of the bolt is given by: $(r_{ext}(z,\theta),\theta,z)$. $r_{ext}$ and $r_{int}$ represent the external thread radius and internal thread radius respectively. For $z < 0$, $r_{ext}$ is undefined. Similarly $r_{int}$ is undefined for $z > 0$.

The image in Figure 3.1 was created using these functions. (In the Figure $\frac{d}{p} = 6.0, \frac{r}{p} = 0.1, a = 0.05$ .)

$$r_{int}(z,\theta) = \begin{cases} \frac{d}{2} & \pi \le \theta \le 2\pi \quad z \ge -p\frac{2\pi-\theta}{2\pi} \\ \frac{d}{2} - f_p(-z, p\frac{\pi-\theta}{\pi}) & 0 \le \theta \le \pi \quad z \ge -p\frac{2\pi-\theta}{2\pi} \\ \frac{d}{2} - f_p(p - (z + p\frac{2\pi-\theta}{2\pi}), p) & 0 \le \theta \le 2\pi \quad z \le -p\frac{2\pi-\theta}{2\pi} \end{cases}$$

$$r_{ext}(z,\theta) = \begin{cases} \frac{d}{2}(1-a) - (p\frac{\sqrt{3}}{2} - 2r) & 0 \le \theta \le \pi \quad z \le p\frac{\theta}{2\pi} \\ \frac{d}{2}(1-a) - (p\frac{\sqrt{3}}{2} - 2r) + f_p(z, p\frac{\theta-\pi}{\pi}) & \pi \le \theta \le 2\pi \quad z \le p\frac{\theta}{2\pi} \\ \frac{d}{2}(1-a) - (p\frac{\sqrt{3}}{2} - 2r) + f_p(z - p\frac{\theta}{2\pi}, p) & 0 \le \theta \le 2\pi \quad z \ge p\frac{\theta}{2\pi} \end{cases}$$

The above descriptions include a linear thread run-up over 180 degrees. This is the longest possible run-up. In simulations, modifying the length of the run-up should have a considerable effect on the assembly. The origin of the nut and bolt, about which rotations will be described, lies at $(0,0,0)$. In cartesian coordinates a point on the internal surface of the nut is given by the *function* $\mathbf{x}_g(z,\theta)$. Correspondingly a point on the surface of the

bolt is given by the function $\mathbf{x}_b(z,\theta)$. The subscript $g$ refers to the *global* or inertial frame, whereas the subscript $b$ refers to the *body* frame. In future analysis the nut will be assumed to be fixed in space, hence it is associated with the inertial frame.

$$\mathbf{x}_g(z,\theta) = \begin{bmatrix} r_{int}(z,\theta)\cos\theta \\ r_{int}(z,\theta)\sin\theta \\ z \end{bmatrix} \quad , \quad \mathbf{x}_b(z,\theta) = \begin{bmatrix} r_{ext}(z,\theta)\cos\theta \\ r_{ext}(z,\theta)\sin\theta \\ z \end{bmatrix}$$

### 3.2.3 Surface Normals

The unit surface normals for the nut and bolt can also be parameterized as functions of $z$ and $\theta$. The method for constructing the unit normal for a generalized cylinder is given by Ponce [Ponce 87:1] and is:

$$\hat{\mathbf{n}}_g(z,\theta) = \frac{-1}{\sqrt{(\frac{\partial r_{int}(z,\theta)}{\partial\theta})^2+(r_{int}(z,\theta))^2(1+\frac{\partial r_{int}(z,\theta)}{\partial z})}} \begin{bmatrix} \frac{\partial r_{int}(z,\theta)}{\partial\theta}\sin\theta + r_{int}(z,\theta)\cos\theta \\ -\frac{\partial r_{int}(z,\theta)}{\partial\theta}\cos\theta + r_{int}(z,\theta)\sin\theta \\ -r_{int}(z,\theta)\frac{\partial r_{int}(z,\theta)}{\partial z} \end{bmatrix}$$

$$\hat{\mathbf{n}}_b(z,\theta) = \frac{1}{\sqrt{(\frac{\partial r_{ext}(z,\theta)}{\partial\theta})^2+(r_{ext}(z,\theta))^2(1+\frac{\partial r_{ext}(z,\theta)}{\partial z})}} \begin{bmatrix} \frac{\partial r_{ext}(z,\theta)}{\partial\theta}\sin\theta + r_{ext}(z,\theta)\cos\theta \\ -\frac{\partial r_{ext}(z,\theta)}{\partial\theta}\cos\theta + r_{ext}(z,\theta)\sin\theta \\ -r_{ext}(z,\theta)\frac{\partial r_{ext}(z,\theta)}{\partial z} \end{bmatrix}$$

where $\hat{\mathbf{n}}_g$ is the normal of the nut in the inertial frame and $\hat{\mathbf{n}}_b$ is the normal of the bolt in the body frame. The function $\hat{\mathbf{n}}_g$ requires the negative sign as the nut is a generalized cylindrical hole.

Note that the surface normals vary continuously with the parameters $z$ and $\theta$, except, of course, at $z = 0$. This is important for determining the velocity of the contact point on the surface during sliding. When contacts occur at the end of the nut or bolt, which is common during the initial insertion, tangents to the circular bases are used to determine the constrained and sliding directions. Thus for the case of contacts between the ends of the two parts, the function $\hat{\mathbf{n}}_t$ is defined to indicate the constraint direction.

$$\hat{\mathbf{n}}_t(\theta_g,\theta_b) = \alpha\frac{\hat{\mathbf{t}}(\theta_g)\times(\mathbf{R}(\theta)\hat{\mathbf{t}}(\theta_b))}{\|\hat{\mathbf{t}}(\theta_g)\times(\mathbf{R}(\theta)\hat{\mathbf{t}}(\theta_b))\|}$$

where:

$$\hat{\mathbf{t}}(\theta) = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix} \quad , \quad \alpha = \begin{cases} 1 & \text{if } [\mathbf{R}(\theta)\hat{\mathbf{t}}(\theta_b)]_z > 0 \\ -1 & \text{if } [\mathbf{R}(\theta)\hat{\mathbf{t}}(\theta_b)]_z < 0 \end{cases} . \qquad (3.1)$$

$\hat{\mathbf{t}}$ is the tangent to the circle at the base. The multiplier $\alpha$ is needed to cause the constraint to point towards the center of the nut. $\mathbf{R}(\boldsymbol{\theta})$ is a rotation matrix, to be discussed shortly, giving the rotational transformation from the body frame to the fixed frame. $\theta_g$ and $\theta_b$ correspond to the contact point parameters. As $z_b$ and $z_g$ are zero for this special case, they do not appear in the function.

### 3.2.4 Geometric Simulation

The representation of the bolt shown in Figure 3.1 was generated using the UNIGRAFIX object oriented graphics software [Sequin 85]. Essentially one 20 degree slice is described and then copied at increasing height and orientation to derive the full thread. The thread run-up was generated separately as were the top and bottom circles.

## 3.3 Analyzing the Configuration Space

The configuration space for threaded insertion is more complicated than for the peg-in-hole problem. The geometry of threaded insertion does not allow a simple reduction from 3 dimensional space to 2 dimensional space. Due to the symmetry of the peg one need only consider the peg and hole cross section. For threads this is not the case. There is one point on the bolt and another on the nut at which the thread begins. To start the insertion process these points must coincide so that rotation of the bolt will allow correct insertion.

The contact configuration transitions in Figure 3.3 give an example of possible contact configuration switches during a forward directed insertion under stiffness control. This diagram is motivated by the analysis of the peg-in-hole problem by [Lozano-Perez 84]. The figure indicates major transitions in the degrees of freedom of motion of the bolt by the number of points at which the nut and bolt are in contact. The more contacts there are, the fewer the degrees of freedom for motion. Two point peg-like contact occurs when two points contact and these contact points lie at the bottom of both the nut and the bolt. This is similar to the 1 point contact configuration considered in the planar peg-in-hole problem.

The sequence of images in 5.1 show a typical progression through these contact configurations during insertion. Unlike the peg-in-hole problem, the contact configuration cannot be determined from a 2D picture. It can be shown that the configurations progress from no contact to one point of contact, then to some combination of two and three points of contact and finally to insertion. Due to the complexity the configuration space has no

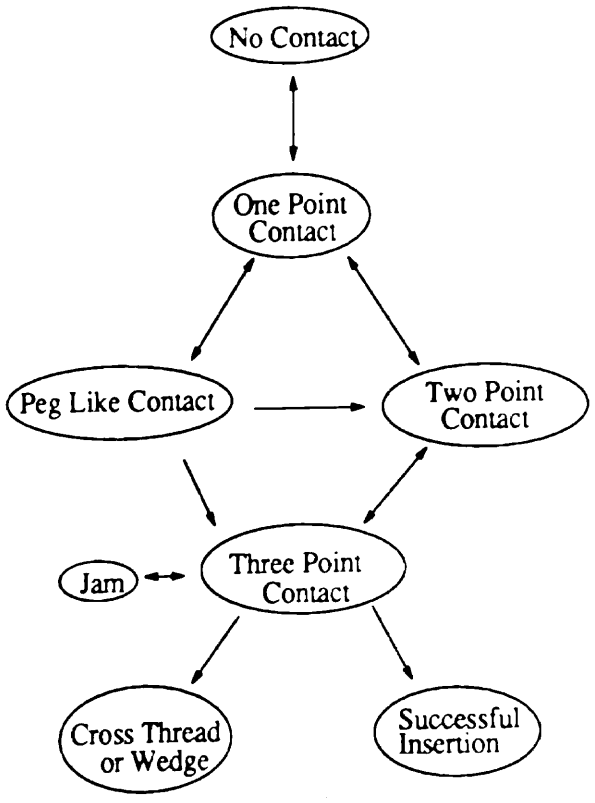


Figure 3.3: Example of Configuration Type Switching.

simple description except in the special case of peg-like contact. In the following sections the configuration space is described by equations, depending on the contact types, which constrain the velocity of the bolt at each contact. Motions that satisfy the combined constraints are the free directions of motion in the configuration space.

## 3.4 Contacts under Rigid Body Motion

Given models for the nut and bolt, the next step is to derive the motion under contact. First we make the assumption of rigid body motion. Under this assumption we cannot model deformation of the parts. Next assume that the parts are put in motion under some control method and that they come into contact. Once the parts are in contact both the constraints on the motion of the bolt and the evolution of the contact points on the surfaces must be determined. Three factors determine the nature of the motion constraint equations: the number of contact points, whether each contact points slides or rolls, and the type of contact. Two of these factors also determine the nature of the contact evolution equation for each contact: whether sliding occurs and the type of contact.

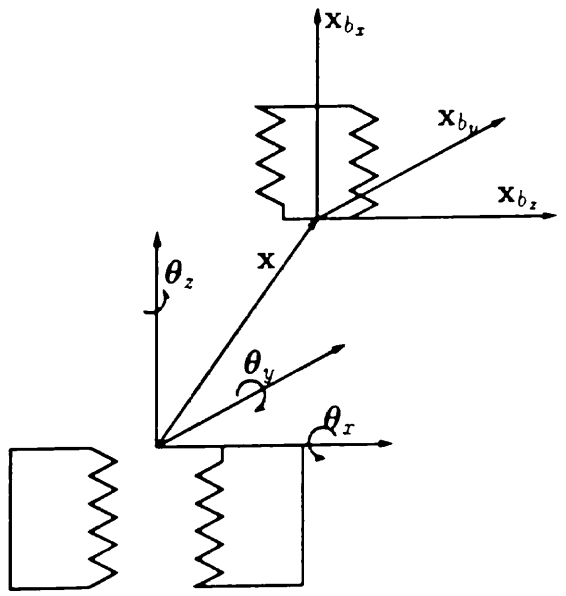


Figure 3.4: Rigid body notation.

This section will define these factors, and the following sections will define the constraint and evolution equations for some of the possible permutations of the aforementioned factors.

### 3.4.1 Rigid Body Motion Notation

For the purposes of this thesis we assume the nut to be fixed in space. Thus the configuration of the bolt with respect to the nut is described by a rigid-body translation and rotation. We will denote the 6 dimensional configuration vector by **c** as:

$$\mathbf{c} = \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\theta} \end{bmatrix}$$

where $\mathbf{x} \in \Re^3$ and $\boldsymbol{\theta} \in SO(3)$, the rotation space. The global origin is located at the center of the top of the nut. The origin of the body, or bolt, frame is at the center of the base of the bolt. The components of $\boldsymbol{\theta}$ and $\mathbf{x}$ will be referred to as:

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} \quad , \quad \mathbf{x} = \begin{bmatrix} x_x \\ x_y \\ x_z \end{bmatrix}$$

Figure 3.4 shows the two frames and the coordinate axes. For contact locations, the notation of [Cole 89] is used. That is we let:

$$\xi_{g_i} = \begin{bmatrix} z_{g_i} \\ \theta_{g_i} \end{bmatrix} \quad , \quad \xi_{b_i} = \begin{bmatrix} z_{b_i} \\ \theta_{b_i} \end{bmatrix}$$

Thus in global coordinates the location of the $i^{th}$ contact between the nut and bolt is given by $\mathbf{x}_g(\xi_{g_i})$. The corresponding point of contact on the bolt, in bolt body coordinates, is $\mathbf{x}_b(\xi_{b_i})$. So for each contact $i$ we must have:

$$\mathbf{x}_g(\xi_{g_i}) = \mathbf{x} + \mathbf{R}(\boldsymbol{\theta})\mathbf{x}_b(\xi_{b_i}) \tag{3.2}$$

where $\mathbf{R}(\boldsymbol{\theta})$ is a rotation matrix. Similarly the surface normals are related by:

$$\hat{\mathbf{n}}_g(\xi_{g_i}) = -\mathbf{R}(\boldsymbol{\theta})\hat{\mathbf{n}}_b(\xi_{b_i}) \tag{3.3}$$

The above relations will be called the *contact constraint equations.*

### 3.4.2 Finding Contact Points

To determine if the nut and bolt are in contact when in a particular configuration **c**, numerical methods must be used. A review of current geometric intersection detection theory reveals that only for situations in which one surface is parametrically defined and the other implicitly defined is there a known numerical method that guarantees known accuracy [Hoffman 89, Patrikalakis 90]. In [Ponce 87:2] a box based method is presented which is computationally intensive, but accurate for the intersections of straight generalized cylinders such as the nut and bolt. We can avoid these computationally intensive methods by utilizing the special geometry of threaded parts.

There are "critical points" on the thread profile at which contact is most likely to occur. We may make the following argument that the internal thread will contact the external thread at the crest of the external thread. As the thread is a smooth surface, contact must occur when the normal of the external thread is opposite to that of the internal thread. The screw thread described is made up of a flat, or rather a chamfer like section, the flank, and a highly curved part, the roots and crests. When the bolt is rotated, i.e. $\theta_x \neq 0$ or $\theta_y \neq 0$, it is not possible for the internal thread flank to touch the external thread flank. Thus contacts must occur between the external crest and the internal flank or the internal root and the external flank.

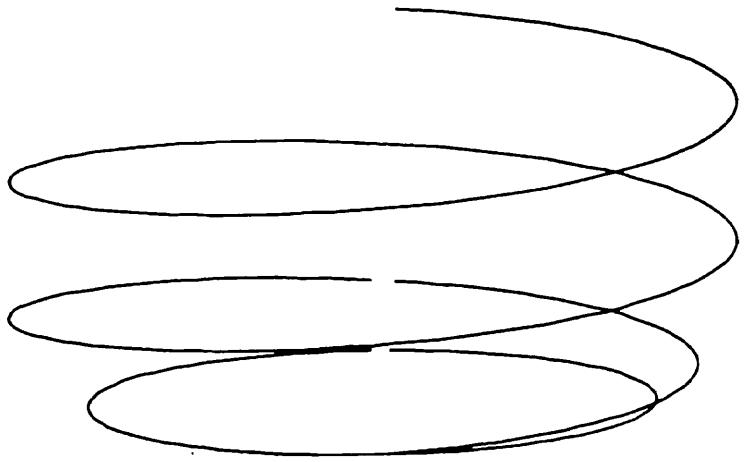

Figure 3.5: Helical thread approximation for intersection detection.

This observation leads to a simple method for surface intersection detection. Consider modeling the bolt as a helix, corresponding to the crest of the external thread. This is shown in Figure 3.5. By mapping the helix to global coordinates via **c** we can determine if a point on the helix lies inside the surface of the nut, implying an intersection. For example consider the point $(\tilde{x}, \tilde{y}, \tilde{z})$. This point lies inside the surface of the nut if $\tilde{z} \leq 0$ and :

$$\sqrt{\tilde{x}^2 + \tilde{y}^2} > r_{int}(\tilde{z}, \arctan(\tilde{y}, \tilde{x}))$$

We can create a similar spiral helix for the nut and an analogous condition for a point of this spiral being inside the bolt by mapping from the global coordinates to the bolt body coordinates. That is, a point from the nut spiral, given by $\tilde{\mathbf{x}}_g$ in the inertial frame, can be mapped to $\tilde{\mathbf{x}}_b$ by $\tilde{\mathbf{x}}_b = \mathbf{R}^T(\tilde{\mathbf{x}}_g - \mathbf{x})$ and this point lies inside the bolt if:

$$\sqrt{\tilde{x}_{b_x}^2 + \tilde{x}_{b_y}^2} < r_{ext}(\tilde{x}_{b_z}, \arctan(\tilde{x}_{b_y}, \tilde{x}_{b_x}))$$

The spiral discussed above also includes a circle at the base to find intersections at $z = 0$ on the bolt. During the initial insertion phase this is the most likely place for contact.

### 3.4.3 Types of Contact

There are two different contact features that give rise to four possible contact types for each contact. The two features are line and surface. A line feature occurs at the end of the nut or bolt. The rest of the nut or bolt is characterized as a surface feature. All contacts considered between the nut and bolt occur at single points. The only time

| Bolt Feature | Nut Feature |
|---|---|
| Surface | Surface |
| Surface | Line |
| Line | Surface |
| Line | Line |

Table 3.1: The relation between contact features and types

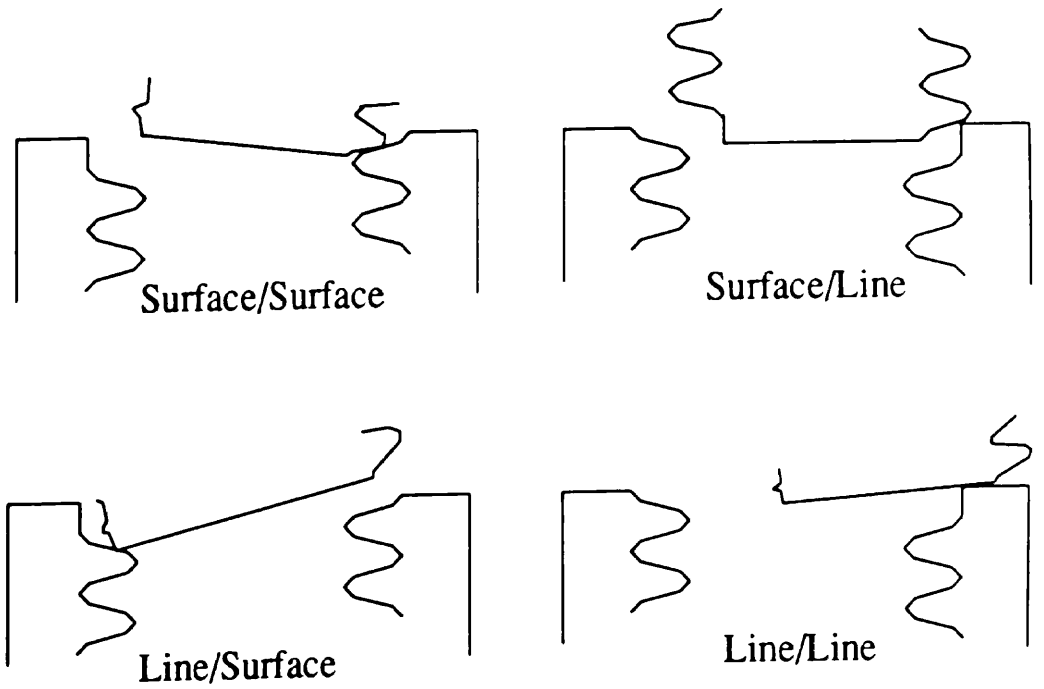


Figure 3.6: Illustrations of contact types.

when this is not the case occurs when the nut and bolt are aligned exactly along the z-axis. In this case there is a continuous line of contact along the helix. At all other times, the existence of an allowance causes the curvatures of the two surfaces to be different. Thus at a contact the surfaces are tangent but the differing curvatures allow the contact to be only at a single point. We also note that unlike polygonal objects, there are no vertex features in this model. The four contact types are listed in table 3.1 and are shown in Figure 3.6.

The stylized pictures in Figure 3.6 show how these contact types would appear for a planar nut and bolt. Planar slices of actual contact configurations would not show the parts to be in contact unless the slices were taken at a particular cross section angle.

To determine the evolution of a contact point its type must be known. A surface feature has two degrees of freedom as the contact point moves on the surface. A line feature has one degree of freedom as the contact moves along the line. In the simulation it was discovered that surface/surface contacts are the most difficult to integrate due to 4 degrees

of freedom. Small errors in the initial contact location estimation would lead to increasing errors in the estimates for the velocities of the contact points on each surface.

### 3.4.4 Definition of Sliding Velocity

To define the meaning of rolling, and hence sliding, the concept of the sliding velocity of a contact point must be introduced. By differentiating the contact constraint equation 3.2 with respect to time the following is derived [Cole 89]:

$$\frac{\partial \mathbf{x}_g}{\partial \xi_g}\dot{\xi}_{g_i} = \dot{\mathbf{x}} + \dot{\mathbf{R}}(\boldsymbol{\theta})\mathbf{x}_b(\xi_{b_i}) + \mathbf{R}(\boldsymbol{\theta})\frac{\partial \mathbf{x}_b}{\partial \xi_b}\dot{\xi}_{b_i} \tag{3.4}$$

Now consider the quantity $\mathbf{v}_{c_i}$ defined as:

$$\mathbf{v}_{c_i} = \frac{\partial \mathbf{x}_g}{\partial \xi_g}\dot{\xi}_{g_i} - \mathbf{R}(\boldsymbol{\theta})\frac{\partial \mathbf{x}_b}{\partial \xi_b}\dot{\xi}_{b_i} = \dot{\mathbf{x}} + \dot{\mathbf{R}}(\boldsymbol{\theta})\mathbf{x}_b(\xi_{b_i}) \tag{3.5}$$

$\mathbf{v}_{c_i}$ is the velocity of the point of contact on the surface of the fixed object minus the velocity of the point of contact on the surface of the moving object. When this quantity is zero, then contact $i$ is a rolling contact, otherwise it is a sliding contact. $\mathbf{v}_{c_i}$ is thus called the *sliding velocity* of contact $i$. The sliding velocity is important for determination of the force due to coulomb friction and for determination of the constrained motion directions as the sliding velocity must be tangential to the contact surface.

## 3.5 Motion Constraint Equations

Each contact has associated with it a constraint on the velocity of the bolt. The constraint depends on the type of contact, and whether that contact is sliding. The constraints for rolling contacts differ considerably according to the number of contacts. For sliding contacts the constraint imposed by each contact can be tabulated. Sliding contacts are discussed first.

### 3.5.1 Sliding Contacts

For the $i^{th}$ sliding contact, the constraint on the sliding velocity $\mathbf{v}_{c_i}$ is given by:

$$\mathbf{v}_{c_i} \cdot \hat{\mathbf{n}}_c(\xi_{g_i}, \xi_{b_i}) = 0 \tag{3.6}$$

| Type<br>Bolt Feature/Nut Feature | Constraint<br>Direction, $\hat{\mathbf{n}}_{c_i}$ |
|---|---|
| Surface/Surface | $\hat{\mathbf{n}}_g(\xi_{g_i})$<br>or $-\mathbf{R}(\theta)\hat{\mathbf{n}}_b(\xi_{b_i})$ |
| Surface/Line | $-\mathbf{R}(\theta)\hat{\mathbf{n}}_b(\xi_{b_i})$ |
| Line/Surface | $\hat{\mathbf{n}}_g(\xi_{g_i})$ |
| Line/Line | $\hat{\mathbf{n}}_t(\xi_{g_i}, \xi_{b_i})$ |

Table 3.2: Constraint direction as determined by contact type.

where the constraint direction $\hat{\mathbf{n}}_{c_i}$ is given in table 3.2.

For surface/surface contacts we have a choice between which normal function to use for the constraint. In the simulation of the bolt motion, it was discovered that it is advantageous to choose the normal which is the slower varying. For example if a root of the nut touches a flank of the bolt, then it would be better to use the function $\hat{\mathbf{n}}_b$ as it will not vary nearly as quickly as $\hat{\mathbf{n}}_g$.

The importance of the direction $\hat{\mathbf{n}}_{c_i}$ cannot be over emphasized. This direction determines the motion of the bolt and the magnitude of the constraint forces. During simulations it was found that any problem, such as numerical error or instability of the motion, which caused the surface parameters $\xi_{b_i}$ and $\xi_{g_i}$, and hence the constraint direction, to vary quickly, could lead the simulation to go unstable and give erroneous results.

### 3.5.2 Rolling Contacts

In this thesis we are not concerned with rolling contacts as much as we are sliding contacts, but for completeness we note the form of the equations. Rolling contacts are not as important as the motion of the bolt should always give sliding contacts. Of course when the bolt crosses threads, it will be in a jammed three point "rolling" contact. The term "rolling" here is used in the general context of a contact that does not slide.

For one point of contact, the rolling constraint is:

$$\mathbf{v}_{c_1} = \mathbf{0}.$$

and the bolt has three degrees of freedom. For two rolling contacts the bolt has 1 degree of freedom for rotation about the line between the contact points. The equations of motion for one rolling contact are given in the dynamics section.

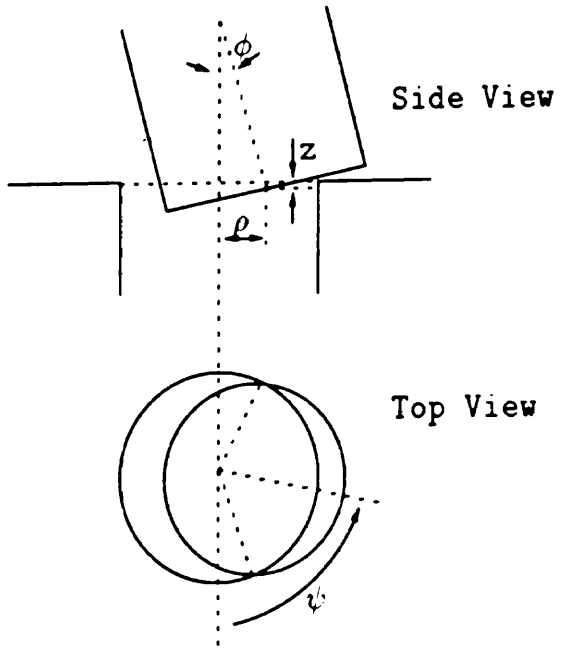


Figure 3.7: Configuration variables for Peg-Like contact.

### 3.5.3 Special Contact Configurations

For the case of two line/line contacts, (i.e. both contacts occur between the end-points of the nut and bolt), a convenient four variable configuration space may be developed. This configuration space is the same as for the round peg-in-hole problem, and has hence been dubbed *peg-like-contact.*

Additionally a special two point contact configuration occurs once the bolt has been inserted into the nut by more than one thread. These two special cases are analyzed below.

#### Peg-Like Contact

A closed form expression may be developed for the four degrees of freedom during two point contact at the ends of the nut and bolt. This has been adapted from Simunovic's [Simunovic 79] three degree of freedom expression for the peg-in-hole which did not include rotation about the z-axis of the bolt. The advantage of using these variables is that they represent unconstrained directions and always give two point contacts at $z_{g_1} = z_{g_2} = z_{b_1} = z_{b_2} = 0$. The four independent configuration parameters are:

$x_z$: depth of insertion of the origin of the bolt.

$\theta_z$: rotation about z-axis of bolt.

$\phi$: angle of the bolt from vertical.

$\psi$: angle of the axis along which the origin of the bolt is displaced from the origin of the nut.

Figure 3.7 shows these parameters. Additionally one dependent configuration parameter is $\rho$ which is the amount of displacement along the axis given by $\psi$.

We may define and derive the following by solving the rotation matrix about the axes described above for the z-y-x angles $\boldsymbol{\theta}$ . Craig [Craig 89] gives formulas for doing this.

$$
\begin{aligned}
u &= \tfrac{-x_z}{\sin\phi} & \theta_{b_1} &= -\cos^{-1}\left(\tfrac{u}{r_{ext}(0,0)}\right) + \psi - \theta_z + \theta_{bz} \\
v &= \sqrt{r_{int}(0,0)^2 - r_{ext}(0,0)^2 + u^2} & \theta_{b_2} &= \cos^{-1}\left(\tfrac{u}{r_{ext}(0,0)}\right) + \psi - \theta_z + \theta_{bz} \\
\rho &= v - u\cos\phi & \theta_{g_1} &= -\cos^{-1}\left(\tfrac{v}{r_{int}(0,0)}\right) + \psi \\
x_x &= \rho\cos\psi & \theta_{g_1} &= \cos^{-1}\left(\tfrac{v}{r_{int}(0,0)}\right) + \psi \\
x_y &= \rho\sin\psi & \theta_{bz} &= \tan^{-1}\left[\tfrac{\cos\psi\sin\psi(1-\cos\phi)}{\sin^2\psi(1-\cos\phi)+\cos\phi}\right] \\
\theta_x &= \tan^{-1}(\sin\psi\tan\phi) \\
\theta_y &= \sin^{-1}(\cos\psi\sin\phi)
\end{aligned}
$$

The dependent variables $\theta_{b_i} = \theta_{b_i}(x_z, \theta_z, \phi, \psi)$ and $\theta_{g_i} = \theta_{g_i}(x_z, \theta_z, \phi, \psi)$ refer to the angle of the location of the contact points on the bolt and nut respectively. That is, during the initial insertion the contact locations are:

$$
\xi_{g_i} = \begin{bmatrix} 0 \\ \theta_{g_i} \end{bmatrix} \quad , \quad \xi_{b_i} = \begin{bmatrix} 0 \\ \theta_{b_i} \end{bmatrix}
$$

The map from the four dimensional motion space to the configuration variables has now been defined.

$$
\begin{bmatrix} x_z \\ \theta_z \\ \phi \\ \psi \end{bmatrix} \xrightarrow{M} \begin{bmatrix} \mathbf{c} \\ \xi_{g_1} \\ \xi_{b_1} \\ \xi_{g_2} \\ \xi_{b_2} \end{bmatrix}
$$

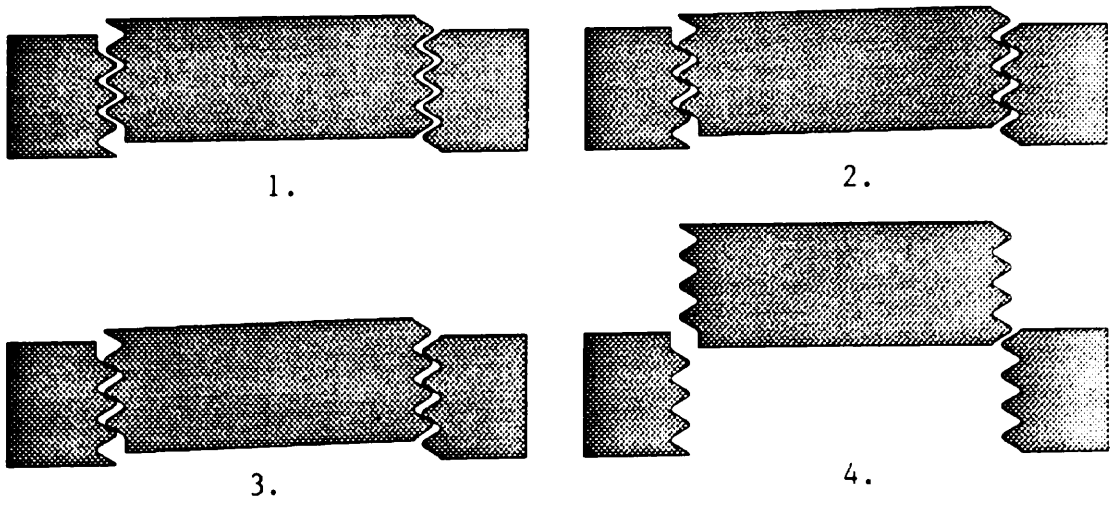


Figure 3.8: Axial slices of the bolt and nut.

### Insertion beyond one full thread

Once the bolt has been inserted so that the threads overlap by more than one full revolution the motion is highly constrained. The sequence of images in Figure 3.8 show planar slices through the nut and bolt at various relative orientations. In all the images pitch $p = 1$, diameter $d = 6p$, and the allowance ratio $a = .05$.

In the first image (going from left to right, top to bottom) the axes of the nut and bolt coincide and there is no rotation. The second image shows the bolt rotated about the axis perpendicular to the page by 1 degree. This is a one point contact configuration. The third image shows a combination of rotation and translation that gives a two point contact. The translation allows an increase in rotation to 1.25 degrees. The last image appears to indicate that motion about the axis perpendicular to the page is allowed only in the clockwise direction. One must be careful in drawing such conclusions as only a slice is shown, not the full 3 dimensional surface.

One can in fact solve for the amount of "slop", or freedom of rotation about the axis out of the page, due to the clearance fit. By solving for the angle and translation at which the "critical points" on the external thread lie on the line given by the flanks of the internal thread. After making small angle assumptions we derive:

$$\phi_{max} = \frac{a}{\sqrt{3}(\frac{d}{2} - r + p\frac{l-0.5}{2\sqrt{3}} - a)}$$

where $l$ is an integer corresponding to the number of full threads that have been inserted and $lp$ is the depth of thread engagement. This result confirms the intuition that as $a$ increases so does the maximum "slop" angle.

## 3.6 Contact Evolution Equations

To determine $\dot{\xi}_g$ and $\dot{\xi}_b$ for each contact point, the contact evolution equations must be solved. By taking the time derivative of the contact constraint equations 3.2 and 3.3 one can construct the appropriate equations, depending on the type of contact.

### 3.6.1 Surface/Surface

For a surface/surface contact the following 6x4 matrix equation must be solved. Cole [Cole 89] notes that these equations are internally consistent. The six equations come from the derivatives of the contact constraint equations 3.2 and 3.3.

$$\begin{bmatrix} \frac{\partial \mathbf{x}_g}{\partial \xi_g} & -\mathbf{R}(\theta)\frac{\partial \mathbf{x}_b}{\partial \xi_b} \\ \frac{\partial \hat{\mathbf{n}}_g}{\partial \xi_g} & \mathbf{R}(\theta)\frac{\partial \hat{\mathbf{n}}_b}{\partial \xi_b} \end{bmatrix} \begin{bmatrix} \dot{\xi}_{g_i} \\ \dot{\xi}_{b_i} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{c_i} \\ -\dot{\mathbf{R}}(\theta)\mathbf{x}_b(\xi_{b_i}) \end{bmatrix} \tag{3.7}$$

### 3.6.2 Surface/Line

In the case of a surface/line or line/surface contact, there are 3 equations given by the derivatives of the constraint equations as well as one equation derived from the fact that the dot product of the tangent of the line and the surface normal must be zero.

$$\begin{bmatrix} \frac{\partial \mathbf{x}_g}{\partial \theta_g} & -\mathbf{R}(\theta)\frac{\partial \mathbf{x}_b}{\partial z_b} & -\mathbf{R}(\theta)\frac{\partial \mathbf{x}_b}{\partial \theta_b} \\ \frac{\partial \hat{\mathbf{t}}}{\partial \theta} \cdot \mathbf{R}\hat{\mathbf{n}}_b & \hat{\mathbf{t}} \cdot \mathbf{R}\frac{\partial \hat{\mathbf{n}}_b}{\partial z_b} & \hat{\mathbf{t}} \cdot \mathbf{R}\frac{\partial \hat{\mathbf{n}}_b}{\partial \theta_b} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{g_i} \\ \dot{z}_{b_i} \\ \dot{\theta}_{b_i} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{c_i} \\ -\hat{\mathbf{t}} \cdot \dot{\mathbf{R}}\hat{\mathbf{n}}_b \end{bmatrix} \tag{3.8}$$

### 3.6.3 Line/Surface

A line on surface contact gives a similar relation:

$$\begin{bmatrix} \frac{\partial \mathbf{x}_g}{\partial z_g} & \frac{\partial \mathbf{x}_g}{\partial \theta_g} & -\mathbf{R}(\theta)\frac{\partial \mathbf{x}_b}{\partial \theta_b} \\ \frac{\partial \hat{\mathbf{n}}_g}{\partial z_g} \cdot \mathbf{R}\hat{\mathbf{t}} & \frac{\partial \hat{\mathbf{n}}_g}{\partial \theta_g} \cdot \mathbf{R}\hat{\mathbf{t}} & \hat{\mathbf{n}}_g \cdot \mathbf{R}\frac{\partial \hat{\mathbf{t}}}{\partial \theta} \end{bmatrix} \begin{bmatrix} \dot{z}_{g_i} \\ \dot{\theta}_{g_i} \\ \dot{\theta}_{b_i} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{c_i} \\ -\hat{\mathbf{n}}_g \cdot \dot{\mathbf{R}}\hat{\mathbf{t}} \end{bmatrix} \tag{3.9}$$

### 3.6.4 Line/Line

For line/line contact there are two contact parameters and the same three equations as for the line/surface case. These equations will be consistent as long as $\mathbf{v}_{c_i}$ satisfies

the motion constraint.

$$\begin{bmatrix} \frac{\partial \mathbf{x}_g}{\partial \theta_g} & -\mathbf{R}(\boldsymbol{\theta})\frac{\partial \mathbf{x}_b}{\partial \theta_b} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{g_i} \\ \dot{\theta}_{b_i} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{c_i} \end{bmatrix} \tag{3.10}$$

# Chapter 4

# Motion Prediction and Control under Contact

Given a model for the bolt and nut interface, the next question is how to control the bolt to move towards successful insertion. Grasp stiffnesses have been successful for describing solutions to similar types of problems, so a motion model is developed with this type of solution in mind.

Quasistatics, dynamics and impulsive forces were used to analyze the motion of the bolt. Quasistatics and the minimum power principle may be used when the velocity of the bolt is small,(i.e. the bolt is not rotating or it is jammed.) When the bolt is rotating, the angular velocity does not allow the minimum power principle to be used, so a dynamic model was created. Dynamics, however, do not describe the instantaneous change in velocity when rigid objects come into contact, thus impulsive forces are used.

## 4.1 The Grasp Stiffness

The concept of a stiffness of a grasp of an object is attractive and has been analyzed in detail for compliant multifingered hands by Cutkosky and Kao[Cutkosky 89:1]. Consider a *equilibrium configuration* of the bolt described by the vector $\mathbf{c}_e \in \Re^3 \times SO(3)$. Now consider the functions $\mathbf{f}_s(\mathbf{c}, \mathbf{c}_e)$ and $\tau_s(\mathbf{c}, \mathbf{c}_e)$. $\mathbf{f}_s$ is the *restoring force* in inertial coordinates that pushes the bolt towards the equilibrium configuration. $\tau_s$ is the *restoring torque* in inertial coordinates that rotates the bolt toward the equilibrium configuration. If these functions are linear then the restoring force and restoring torque are given by:

$$\begin{bmatrix} \mathbf{f}_s(\mathbf{c}, \mathbf{c}_e) \\ \boldsymbol{\tau}_s(\mathbf{c}, \mathbf{c}_e) \end{bmatrix} = \mathbf{K}(\mathbf{c}_e - \mathbf{c}) \tag{4.1}$$

where $\mathbf{K} \in \Re^{6\times 6}$. Thus when the bolt is in the equilibrium configuration, the grasp stiffness applies no force or torque.

Implicitly defined by this equation is the *center of stiffness frame* which is defined by [Loncaric 87] to be the coordinate frame which "maximally decouples the rotational and translational aspects of the stiffness." If we assume that the stiffness matrix is positive definite, which it must be for stability [Salisbury 82], and that it is symmetric, as neither direction along an axis is favored, then the stiffness matrix will be *diagonal* in the center of stiffness frame. The basis vectors for this frame need not be standard translations and rotations. The presence of terms coupling translations to rotations in the nondiagonal stiffness will cause the principle directions in the center of stiffness frame to be combinations of rotations and translations.

The search space for a linear stiffness may be reduced by noting that the terms affecting torques about the $x$ and $y$ axes must be the same. Additionally the terms affecting forces along the $x$ and $y$ axes must be the same. The reason for this is that we are assuming no knowledge of the initial configuration of the bolt with respect to the nut, hence motions in the plane perpendicular to the axis of the nut must be treated equally. The equality would not have to hold if, for instance, we knew relative rotation about the z-axis of the bolt with respect to the z-axis of the nut. In such a case a special trajectory would be developed to cause the threads to line up. Alternatively, if we knew there was orientation error about the x-axis and not the y-axis, then also the equality would not hold. In addition to the above restrictions, the stiffness matrix should be positive definite and symmetric for stability. So, for equal errors in each configuration variable, the stiffness matrix must have the form:

$$\mathbf{K} = \begin{bmatrix} k_{xx} & k_{xy} & k_{xz} & k_{x\theta_x} & k_{x\theta_y} & k_{x\theta_z} \\ k_{xy} & k_{xx} & k_{xz} & k_{x\theta_y} & k_{x\theta_x} & k_{x\theta_z} \\ k_{xz} & k_{xz} & k_{zz} & k_{x\theta_z} & k_{x\theta_z} & k_{z\theta_z} \\ k_{x\theta_x} & k_{x\theta_y} & k_{x\theta_z} & k_{\theta_x\theta_x} & k_{\theta_x\theta_y} & k_{\theta_x\theta_z} \\ k_{x\theta_y} & k_{x\theta_x} & k_{x\theta_z} & k_{\theta_x\theta_y} & k_{\theta_x\theta_x} & k_{\theta_x\theta_z} \\ k_{x\theta_z} & k_{x\theta_z} & k_{z\theta_z} & k_{\theta_x\theta_z} & k_{\theta_x\theta_z} & k_{\theta_z\theta_z} \end{bmatrix}$$

Thus the problem of finding the optimal linear stiffness has been reduced to finding 12 matrix parameters.

Before leaving this topic, the relation between the stiffness matrix and the rotation convention must be stressed. The variables $\theta_x, \theta_y$, and $\theta_z$ correspond to some rotation convention. Thus the torques given by equation 4.1 are expressed relative to that rotation convention. In the case of the *zyx* fixed axis rotation convention, the resulting torques are about axes fixed in the inertial frame. If a body rotation convention had been chosen the torques would have been in the body frame. It may be useful to think of the rotation convention as characterizing the mechanism holding the bolt. The order of rotations in the body axis scheme would determine the order of the joints in the arm holding the bolt.

## 4.2 Contact Forces

At each contact point the reaction force applied to the bolt will be denoted by $\mathbf{f}_i$ and the reaction torque by $\boldsymbol{\tau}_i = (\mathbf{R}(\boldsymbol{\theta})\mathbf{x}_b(\xi_{b_i})) \times \mathbf{f}_i$. We should note that forces may not be applied outside of the friction cone. This constraint can be expressed by setting the normal force, $\mathbf{f}_{n_i} = \mathbf{f}_i \cdot \hat{\mathbf{n}}_g(\xi_{g_i})$ and the tangential force $\mathbf{f}_{t_i} = \mathbf{f}_i - \mathbf{f}_{n_i}$. Thus $\|\mathbf{f}_{t_i}\| \leq \mu\|\mathbf{f}_{n_i}\|$. $\mu$ is the coefficient of dry coulomb friction.

Contact forces are reaction forces only. They are not created by the nut, but arise solely because the nut is fixed in space. The constraint equations discussed in the previous chapter provide the necessary equations to solve for the contact forces for one, two, and three point contact for both rolling and sliding.

## 4.3 Quasistatic Analysis

Originally we thought that quasistatic analysis would be sufficient for this problem as we anticipated the dissipative forces to be large compared to the inertial forces. Many other sliding-contact problems have been analyzed using quasistatics. However, when the principles were applied to the screw threading problem only certain limited cases could be analyzed using the minimum power principle. The following sections summarize the quasistatic analysis and show the limitations.

### 4.3.1 The Quasistatic and Minimum Power Principles

If one assumes the mass and the accelerations of the bolt are small then the motion will obey the "Quasistatic Assumption" [Peshkin 88]. We can then write the relations:

$$\sum \mathbf{f} = 0 \Rightarrow \mathbf{f}_s(\mathbf{c}, \mathbf{c}_e) + \sum_i \mathbf{f}_i = 0$$
$$\sum \boldsymbol{\tau} = 0 \Rightarrow \boldsymbol{\tau}_s(\mathbf{c}, \mathbf{c}_e) + \sum_i \boldsymbol{\tau}_i = 0$$

Peshkin and Sanderson [Peshkin 88] show that when coulomb friction is the only dissipative or velocity dependent force acting on the system the following minimum power principle holds:

> *A quasistatic system chooses that motion, from among all motions satisfying the constraints, which minimizes the instantaneous power.*

When using this principle one must be careful to include only the components of force which do work, and not the forces of constraint. Power, $P$ is:

$$P = \sum_i \mathbf{f}_{t_i} \cdot \mathbf{v}_{c_i} + \mathbf{f}_{s_{xc}}(\mathbf{c}, \mathbf{c}_e) \cdot \dot{\mathbf{x}} + \boldsymbol{\tau}_{s_{xc}}(\mathbf{c}, \mathbf{c}_e) \cdot \boldsymbol{\theta}$$

where $\mathbf{f}_{s_{xc}}$ and $\boldsymbol{\tau}_{s_{xc}}$ represent the forces and torques applied by the grasp which are not in the constraint directions.

As a simple example consider $\xi_{b_i}$ to be fixed, as in the case of a point on the bolt sliding on a flank of the nut, then $\mathbf{v}_{c_i}$, the contact velocity, is related to $\dot{\mathbf{c}}$ linearly by the contact constraint equations. Hence $P$ is linear in velocity. To determine the direction of sliding of the contact point we must consider constraints on the velocities. $\mathbf{v}_{c_i}$ must stay on the surface. So:

$$\mathbf{v}_{c_i} \cdot \hat{\mathbf{n}}_g = 0$$

To find the direction of $\mathbf{v}_{c_i}$ we must make the assumption that $\|\mathbf{v}_{c_i}\|$ is small and constant during sliding. The $\mathbf{v}_{c_i}$ which minimize $P$ over these constraints constitutes the sliding direction.

### 4.3.2 One Point Contact

Consider the case of a point on the bolt rolling on a flank region of the nut. The constraint of one point rolling contact leaves 3 degrees of freedom for motion of the bolt.

These directions are the *free directions*. In quasistatics the equilibrium values for these directions are those that provide the correct force and torques via the stiffness function to provide force and torque equilibrium. The equilibrium configuration $\mathbf{c}$ is given by the solution to:

$$\mathbf{x} = \mathbf{x}_g(\xi_{g_1}) - \mathbf{R}(\theta)\mathbf{x}_b(\xi_{b_1})$$
$$\mathbf{R}(\theta)\mathbf{x}_b(\xi_{b_1}) \times \mathbf{f}_s(\mathbf{c}, \mathbf{c}_e) = -\tau_s(\mathbf{c}, \mathbf{c}_e)$$

keeping $\mathbf{c}_e, \mathbf{x}_b$ and $\mathbf{x}_g$ fixed.

The motion of a point sliding on a plane has been studied by Cutkosky and Kao [Cutkosky 89:2]. They derive a closed form expression for the motion of the point under planar stiffness control. We cannot hope to obtain a closed form relation for our sliding analysis, so our methods are numerical. This is similar to Cole's [Cole 89] case for the analysis of contacts sliding on curved surfaces. In the case of a one point sliding the minimum power principle shows that the direction of sliding is same as the direction of the tangential force. Hence

$$\frac{\mathbf{v}_{slip}}{\|\mathbf{v}_{slip}\|} = \frac{\mathbf{f}_t}{\|\mathbf{f}_t\|}$$

### 4.3.3 Two Point Contact

A similar method for solving for equilibrium configurations and sliding can be used for two point contact. First consider the case of both contacts rolling, leaving one degree of freedom for motion.

The solution for contact forces can be found by using the concepts of contact wrenches and grasp matrices discussed by Kerr and Roth [Kerr 86]. As we are considering quasistatic equilibrium we may write the relationship between the applied stiffness force and the contact forces by:

$$\begin{bmatrix} f_{b_x} \\ f_{b_y} \\ f_{b_z} \\ \tau_{b_x} \\ \tau_{b_y} \\ \tau_{b_z} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & r_{1_z} & -r_{1_y} & 0 & r_{2_z} & -r_{2_y} \\ -r_{1_z} & 0 & r_{1_x} & -r_{2_z} & 0 & r_{2_x} \\ r_{1_y} & -r_{1_x} & 0 & r_{1_y} & -r_{1_x} & 0 \end{bmatrix} \begin{bmatrix} f_{1_x} \\ f_{1_y} \\ f_{1_z} \\ f_{2_x} \\ f_{2_y} \\ f_{2_z} \end{bmatrix}$$

where $\mathbf{r}_i = \mathbf{R}(\theta)\mathbf{x}_{b_i}$. This can be written more compactly as:

$$\begin{bmatrix} \mathbf{f}_s(\mathbf{c}, \mathbf{c}_e) \\ \boldsymbol{\tau}_s(\mathbf{c}, \mathbf{c}_e) \end{bmatrix} = \mathbf{W} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}$$

where $\mathbf{W} \in \Re^{6\times 6}$ . $\mathbf{W}$ is *not* full rank because we know that there is one axis about which $\mathbf{f}_1$ and $\mathbf{f}_2$ cannot exert a torque. Point contacts cannot exert a torque about the line between them. In this case $\mathbf{r}_i \times \mathbf{f}_i$ has no component of torque about $(\mathbf{x}_{g_1} - \mathbf{x}_{g_2})$.

As we know $\mathbf{f}_s$ and $\boldsymbol{\tau}_s$, we wish to solve for $\mathbf{f}_1$ and $\mathbf{f}_2$. $\mathbf{W}$ is not invertible so we cannot immediately do this. This problem is rectified by realizing that the quasistatic system will be in equilibrium when the amount of rotation, call it $\omega$, about the axis $(\mathbf{x}_{g_1} - \mathbf{x}_{g_2})$ is such that the forces and torques are in equilibrium. In other words, with the addition of the parameter $\omega$ on the right hand side of the previous equation we can make the map onto.

This still does not solve the inversion problem. Although the map is onto, it is not one-to-one. There are infinitely many combinations of $\mathbf{f}_1$ and $\mathbf{f}_2$ that have components summing to zero along $(\mathbf{x}_{g_1} - \mathbf{x}_{g_2})$ that will solve the quasistatic equilibrium equations. These forces are the internal forces of the "grasp" of the bolt by the nut along the line between the contact points. We can make the map invertible by realizing that the nut applies no internal force in its "grasp" of the bolt. This is due to the assumption that the nut is rigid and does not deform. Mathematically this means that the components of $\mathbf{f}_1$ and $\mathbf{f}_2$ along this line are equal.

$$\mathbf{f}_1 \cdot (\mathbf{x}_{g_1} - \mathbf{x}_{g_2}) - \mathbf{f}_2 \cdot (\mathbf{x}_{g_1} - \mathbf{x}_{g_2}) = 0$$

After adding this constraint we are now able to invert the map and solve for $\mathbf{f}_1$ and $\mathbf{f}_2$ given $\mathbf{c}$ and $\mathbf{c}_e$.

The above analysis can be used to determine constraints on the constants of the stiffness matrix. First recall the analysis in section 3.5.3 regarding the "slop angle" indicating the amount of free motion once the depth of thread engagement is greater than one complete thread. Suppose we have some initial error ball in orientation and translation. This could be described by two parameters $\phi_e$ and $r_e$ which indicate the orientation error from vertical and the displacement distance from the center of the hole. Consider the equilibrium configuration to be described by $\phi_e$ and $r_e$. That is, the zero force location lies at some unreachable point in the configuration space if $\phi_e > \phi_{max}$ or if $r_e$ is greater than the

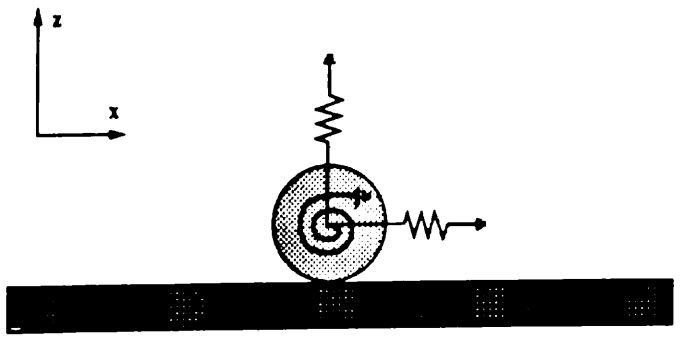


Figure 4.1: A disk rotating on a line

allowance. This will result in the two point contact described by the "slop angle." The actual configuration will be given by $\phi_{max}$, thus there will be restoring forces and torques. The stiffness matrix values should be chosen so that these forces and torques cause the contact forces as determined by the quasistatic analysis to lie outside the friction cone.

### 4.3.4 The Limitations of Quasistatic Analysis

A simple planar problem is enough to show that quasistatics are not sufficient for this problem.

Consider a disk on a line as shown in Figure 4.1. Consider three springs attached to the center of the disk. Two springs are translational and lie in the x and z axes, one spring is rotational and creates a torque about the y-axis, into the page. Now suppose the endpoint of the rotational spring is moved at constant angular velocity, causing the disk, when not in contact, to rotate at constant angular velocity. Next suppose the endpoint of the x axis spring is held fixed. Finally, apply a sinusoidal motion to the endpoint of the vertical spring such that the normal force cycles between being almost zero to being very large, and such that the disk continuously contacts the line. This situation is analogous to that of the bolt being inserted into the nut with one point of contact between the base of the bolt and the top of the nut. The million dollar question is: What is the angular velocity and the translational velocity of the disk?

One can quickly see that quasistatics does not apply here. There is no constraint that involves both the rotational and translational velocity that will give the equilibrium rotational velocity. For example suppose the disk is moving to the right and rotating counterclockwise quasistatically (i.e. in such a way that the forces and torques sum to zero). Now suppose the normal force is decreased. The question is how much of the

increased rotational velocity goes into decreasing translational velocity. The point is that although the accelerations are small, the system is rarely in quasistatic equilibrium.

This simple example illustrates that we cannot use the minimum power principle to determine the sliding direction. This is unfortunate. In [Schimmels 90] an optimum damping matrix for the peg-in-hole problem was found based on the idea that a generalized damper gives a *linear* relation between force and velocity. We had hoped to find a similar linear relationship for stiffness control and thus be able to find the optimum stiffness by a similar method. Instead we have simulated the dynamic motion under various stiffness controls to determine qualitatively what a good stiffness would be.

## 4.4 Dynamic Equations of Motion

Creating a full dynamic simulation of the motion of the bolt requires numerical integration of a function that gives the time derivative of the state variables as a function of the current state. The current state, $\mathbf{s}$ is:

$$\mathbf{s} = \begin{bmatrix} \mathbf{c} & \dot{\mathbf{c}} & \xi_{b_1} & \xi_{g_1} & \xi_{b_2} & \xi_{g_2} & \xi_{b_3} & \xi_{g_3} \end{bmatrix}^T \tag{4.2}$$

The derivative of the contact parameters were derived in the previous chapter, so in the following sections $\ddot{\mathbf{x}}$ and $\ddot{\boldsymbol{\theta}}$ are derived as functions of $\mathbf{x}, \dot{\mathbf{x}}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}$.

### 4.4.1 Useful Differential Relations

In the derivation of the dynamic equations it is useful to consider $\xi_{b_i}$ fixed and to decompose $\dot{\mathbf{R}}(\boldsymbol{\theta})\mathbf{x}_b(\xi_{b_i})$ into a matrix times $\dot{\boldsymbol{\theta}}$. The matrix $\mathbf{P}(\boldsymbol{\theta}, \xi_{b_i})$ and the vector $\mathbf{p}_l(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \xi_{b_i}, \dot{\xi_{b_i}})$ are defined for this purpose.

$$\dot{\mathbf{R}}(\boldsymbol{\theta})\mathbf{x}_b(\xi_{b_i}) = \mathbf{P}\dot{\boldsymbol{\theta}} \tag{4.3}$$

$$\frac{d}{dt}(\dot{\mathbf{R}}(\boldsymbol{\theta})\mathbf{x}_b(\xi_{b_i})) = \mathbf{P}\ddot{\boldsymbol{\theta}} + \mathbf{p}_l \tag{4.4}$$

It is also necessary to find the relation between $\dot{\boldsymbol{\theta}}$ and $\boldsymbol{\omega}_o$, the angular velocity of the bolt in the inertial frame. The following property of rotation matrices is used:

$$\dot{\mathbf{R}}(\boldsymbol{\theta}) = \boldsymbol{\omega}_o \times \mathbf{R}(\boldsymbol{\theta}) = \boldsymbol{\Omega}_o \mathbf{R}(\boldsymbol{\theta}) \tag{4.5}$$

where $\boldsymbol{\Omega}_o$ is skew symmetric. By differentiating the rotation matrix with respect to time, one derives a set of three matrices such that

$$\dot{\mathbf{R}}(\boldsymbol{\theta}) = \mathbf{R}_1\dot{\theta_x} + \mathbf{R}_2\dot{\theta_y} + \mathbf{R}_3\dot{\theta_z}. \tag{4.6}$$

By combining equations 4.5 and 4.6 one derives:

$$\boldsymbol{\omega}_o = S(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{4.7}$$

One must be careful to watch for singularities in this matrix to avoid problems inverting it. In the case of a zyx fixed axis rotation, its determinant is $\cos\theta_y$.

In addition note that the angular velocity of the bolt in the body frame $\boldsymbol{\omega}_b$ is related to $\boldsymbol{\omega}_o$ by:

$$\boldsymbol{\omega}_b = \mathbf{R}^T(\boldsymbol{\theta})\boldsymbol{\omega}_o. \tag{4.8}$$

### 4.4.2 Inertia Tensor

For the purposes of this thesis, the inertia tensor for the bolt is approximated as that for a cylinder with radius $r_s$ equal to half of the external thread basic minor diameter. As the screw is assumed to be solid and a small fraction of the mass is in the threads, this assumption seems reasonable. Hence define the rotational inertia tensor $\mathbf{I}_n$ as:

$$\mathbf{I}_n = \begin{bmatrix} \frac{1}{12}ml^2 + \frac{1}{4}mr_s^2 & 0 & 0 \\ 0 & \frac{1}{12}ml^2 + \frac{1}{4}mr_s^2 & 0 \\ 0 & 0 & \frac{1}{2}mr_s^2 \end{bmatrix} \tag{4.9}$$

The length, $l$, in the simulation is four thread pitches. $m$ is the mass. As $\mathbf{I}_n$ is given about the center of mass, it is constant in the body frame.

### 4.4.3 Euler's Equations

The derivations for the accelerations are based on the following two equations. The first is Euler's equation of motion for a rotating rigid body. It is written in the body frame. The second expresses that the sum of forces in the inertial frame is equal to the derivative of the translational momentum. These two equations are decoupled about the center of mass of the moving object, hence $\mathbf{x}$ has the subscript $cm$. $\mathbf{x}_{cm}$ is the location of

the center of mass in the inertial frame. Additionally the forces on the right hand side of the conservation of linear momentum equation are applied at the center of mass, so they also have the subscript *cm* and are given in the inertial frame [Goldstein 80].

$$\mathbf{I}_n\dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times \mathbf{I}_n\boldsymbol{\omega}_b = \sum \boldsymbol{\tau}_b \tag{4.10}$$

$$m\ddot{\mathbf{x}}_{cm} = \sum \mathbf{f}_{cm} \tag{4.11}$$

To put these equations in terms of the state variable $\mathbf{c}$ we make use of equation 4.4 to obtain:

$$\begin{aligned}
\dot{\mathbf{x}}_{cm} &= \dot{\mathbf{x}} + \dot{\mathbf{R}}\mathbf{r}_{cm} = \dot{\mathbf{x}} + \mathbf{P}_{cm}\dot{\boldsymbol{\theta}} \\
\ddot{\mathbf{x}}_{cm} &= \ddot{\mathbf{x}} + \ddot{\mathbf{R}}\mathbf{r}_{cm} \\
&= \ddot{\mathbf{x}} + \mathbf{P}_{cm}\ddot{\boldsymbol{\theta}} + \mathbf{p}_{l_{cm}}
\end{aligned}$$

where $\mathbf{r}_{cm}$ is the *constant* vector from the bolt origin to the center of mass in the bolt body frame. The sum of the forces applied to the bolt is:

$$\sum \mathbf{f}_{cm} = \mathbf{f}_s + \sum \mathbf{f}_i$$

where $\mathbf{f}_s$ is the stiffness force and $\mathbf{f}_i$ are the contact forces. As these forces are not applied at the center of mass, but at points in the body frame, they contribute to the net torque. Thus the net torque in the body frame is:

$$\sum \boldsymbol{\tau}_b = \mathbf{R}^T\boldsymbol{\tau}_s - \mathbf{r}_{cm} \times (\mathbf{R}^T\mathbf{f}_s) + \sum_i [\mathbf{x}_b(\xi_{b_i}) - \mathbf{r}_{cm}] \times (\mathbf{R}^T\mathbf{f}_i) \tag{4.12}$$

where $\boldsymbol{\tau}_s$ is the stiffness torque in the inertial frame.

Using the relations 4.8 and 4.7 the following is derived:

$$\begin{aligned}
\boldsymbol{\omega}_b &= \mathbf{R}^T\mathbf{S}\dot{\boldsymbol{\theta}} \\
\dot{\boldsymbol{\omega}}_b &= \mathbf{R}^T\mathbf{S}\ddot{\boldsymbol{\theta}} + \dot{\mathbf{R}}^T\mathbf{S}\dot{\boldsymbol{\theta}} + \mathbf{R}^T\dot{\mathbf{S}}\dot{\boldsymbol{\theta}}
\end{aligned}$$

To simplify future expressions, $\mathbf{n}_l(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ will be defined as:

$$\mathbf{n}_l(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = (\mathbf{R}^T\mathbf{S}\dot{\boldsymbol{\theta}}) \times (\mathbf{I}_n\mathbf{R}^T\mathbf{S}\dot{\boldsymbol{\theta}}) + (\mathbf{I}_n\dot{\mathbf{R}}^T\mathbf{S} + \mathbf{I}_n\mathbf{R}^T\dot{\mathbf{S}})\dot{\boldsymbol{\theta}}. \tag{4.13}$$

After substituting these results into Euler's equations we have:

$$\mathbf{I}_n\mathbf{R}^T\mathbf{S}\ddot{\boldsymbol{\theta}} = \mathbf{R}^T\boldsymbol{\tau}_s - \mathbf{r}_{cm} \times (\mathbf{R}^T\mathbf{f}_s) - \mathbf{n}_l + \sum_i [\mathbf{x}_b(\xi_{b_i}) - \mathbf{r}_{cm}] \times (\mathbf{R}^T\mathbf{f}_i) \tag{4.14}$$

$$m\ddot{\mathbf{x}} = \mathbf{f}_s - m(\mathbf{P}_{cm}\ddot{\boldsymbol{\theta}} + \mathbf{p}_{l_{cm}}) + \sum_i \mathbf{f}_i \tag{4.15}$$

### 4.4.4 No Contact

When there are no contact forces, equations 4.14 and 4.15 reduce to:

$$\mathbf{I}_n\mathbf{R}^T\mathbf{S}\ddot{\boldsymbol{\theta}} = \mathbf{R}^T\boldsymbol{\tau}_s - \mathbf{r}_{cm} \times (\mathbf{R}^T\mathbf{f}_s) - \mathbf{n}_l$$

$$m\ddot{\mathbf{x}} = \mathbf{f}_s - m(\mathbf{P}_{cm}\ddot{\boldsymbol{\theta}} + \mathbf{p}_{l_{cm}})$$

Thus when the bolt is not in contact, the derivative of the state $\mathbf{c}$ can be found by inverting a 3x3 matrix.

### 4.4.5 One Point Contact

For one point contact, the constraint force, $\mathbf{f}_1$ must be eliminated from the equations of motion. In the case of a rolling contact this is fairly straightforward. For a sliding contact the derivation is more involved. In both cases the equations are abbreviated by not explicitly stating the dependence of $\mathbf{x}_g$ and $\hat{\mathbf{n}}_g$ on $\xi_{g_1}$ and $\mathbf{x}_b$ and $\hat{\mathbf{n}}_b$ on $\xi_{b_1}$.

#### Rolling

In the case of rolling, the motion constraint equation given in section 3.5.2 can be used to eliminate $\mathbf{f}_1$. Differentiating the constraint $\mathbf{v}_{c_1} = \mathbf{0}$ one obtains:

$$\ddot{\mathbf{x}} + \frac{d}{dt}(\dot{\mathbf{R}}\mathbf{x}_b) = \mathbf{0}$$

Substituting this result into equation 4.4 gives:

$$\ddot{\mathbf{x}} = -\mathbf{P}_{xb}\ddot{\boldsymbol{\theta}} - \mathbf{p}_{l_{xb}} \tag{4.16}$$

Combining this with equation 4.15 gives:

$$m\left[(\mathbf{P}_{cm} - \mathbf{P}_{xb})\ddot{\theta} + (\mathbf{p}_{l_{cm}} - \mathbf{p}_{l_{xb}})\right] = \mathbf{f}_s + \mathbf{f}_l$$

Define $\mathbf{P}$ and $\mathbf{p}_l$ as:

$$\mathbf{P} = \mathbf{P}_{xb} - \mathbf{P}_{cm} \quad , \quad \mathbf{p}_l = {}_{l_{xb}} - \mathbf{p}_{l_{cm}} \tag{4.17}$$

so the constraint force is:

$$\mathbf{f}_l = -m\mathbf{P}\ddot{\theta} - m\mathbf{p}_l - \mathbf{f}_s.$$

Substituting this into equation 4.14 gives:

$$\begin{aligned}\left[\mathbf{I}_n\mathbf{R}^T\mathbf{S} + m(\mathbf{x}_b - \mathbf{r}_{cm}) \times (\mathbf{R}^T\mathbf{P})\right]\ddot{\theta} &= \mathbf{R}^T\boldsymbol{\tau}_s - \mathbf{r}_{cm} \times (\mathbf{R}^T\mathbf{f}_s) + \\ &\quad (\mathbf{x}_b - \mathbf{r}_{cm}) \times (\mathbf{R}^T(-m\mathbf{p}_l - \mathbf{f}_s)) - \mathbf{n}_l \\ &= \mathbf{R}^T\boldsymbol{\tau}_s - \mathbf{x}_b \times (\mathbf{R}^T\mathbf{f}_s) - \\ &\quad m(\mathbf{x}_b - \mathbf{r}_{cm}) \times (\mathbf{R}^T\mathbf{p}_l) - \mathbf{n}_l\end{aligned}$$

By recognizing that $(\mathbf{x}_b - \mathbf{r}_{cm})\times$ can be represented as a skew symmetric matrix, the last equation above can be solved for $\ddot{\theta}$ by inverting the matrix on the left hand side. $\ddot{\mathbf{x}}$ follows directly from equation 4.16. At this point $\mathbf{f}_l$ can be checked to see if it lies inside the friction cone. If it does not, then the following equations for motion during sliding are used.

### Sliding

During sliding the bolt has 5 degrees of freedom of motion. There are 6 equations for the dynamics given by Euler's equations. There is one more equation describing the motion constraint from section 3.5.1. The direction of the constraint force is known, yet the magnitude must be determined. This is expressed by:

$$\mathbf{f}_l = f_n\left[\hat{\mathbf{n}}_g - \mu\frac{\mathbf{v}_{c_l}}{\|\mathbf{v}_{c_l}\|}\right] \tag{4.18}$$

where $f_n$ is the magnitude of the normal force. The following derivation eliminates $f_n$ from the Euler's equations, giving a solution for $\ddot{\theta}$.

First differentiate the constraint equation $\mathbf{v}_{c_1} \cdot \hat{\mathbf{n}}_g = 0$ to get the acceleration constraint:

$$\dot{\mathbf{v}}_{c_1} \cdot \hat{\mathbf{n}}_{c_1} + \mathbf{v}_{c_1} \cdot \dot{\hat{\mathbf{n}}}_{c_1} = 0. \tag{4.19}$$

Now

$$\begin{aligned} \dot{\mathbf{v}}_{c_1} &= \ddot{\mathbf{x}} + \frac{d}{dt}(\dot{\mathbf{R}}\mathbf{x}_b) \\ &= \ddot{\mathbf{x}} + \mathbf{P}_{xb}\ddot{\boldsymbol{\theta}} + \mathbf{p}_{l_{xb}} \end{aligned}$$

but

$$m\ddot{\mathbf{x}} = \mathbf{f}_s + \mathbf{f}_1 - m(\mathbf{P}_{cm}\ddot{\boldsymbol{\theta}} + \mathbf{p}_{l_{cm}})$$

so

$$\begin{aligned} \ddot{\mathbf{x}} &= \frac{1}{m}(\mathbf{f}_s + \mathbf{f}_1) - (\mathbf{P}_{cm}\ddot{\boldsymbol{\theta}} + \mathbf{p}_{l_{cm}}) \\ \dot{\mathbf{v}}_{c_1} &= \frac{1}{m}(\mathbf{f}_s + \mathbf{f}_1) + (\mathbf{P}_{xb} - \mathbf{P}_{cm})\ddot{\boldsymbol{\theta}} + (\mathbf{p}_{l_{xb}} - \mathbf{p}_{l_{cm}}). \end{aligned}$$

By defining $\mathbf{P}$ and $\mathbf{p}_l$ as in equation 4.17 then:

$$\dot{\mathbf{v}}_{c_1} = \frac{1}{m}(\mathbf{f}_s + \mathbf{f}_1) + \mathbf{P}\ddot{\boldsymbol{\theta}} + \mathbf{p}_l.$$

The acceleration constraint equation 4.19 now becomes:

$$(\frac{1}{m}\mathbf{f}_s + \mathbf{P}\ddot{\boldsymbol{\theta}} + \mathbf{p}_l) \cdot \hat{\mathbf{n}}_{c_1} + \frac{1}{m}\mathbf{f}_1 \cdot \hat{\mathbf{n}}_{c_1} + \mathbf{v}_{c_1} \cdot \dot{\hat{\mathbf{n}}}_{c_1} = 0. \tag{4.20}$$

Now recognizing that $\mathbf{f}_1 \cdot \hat{\mathbf{n}}_{c_1} = f_n$ then:

$$\begin{aligned} f_n &= -((\mathbf{f}_s + m\mathbf{P}\ddot{\boldsymbol{\theta}} + m\mathbf{p}_l) \cdot \hat{\mathbf{n}}_{c_1} + m\mathbf{v}_{c_1} \cdot \dot{\hat{\mathbf{n}}}_{c_1} \\ &= -m\mathbf{P}\ddot{\boldsymbol{\theta}} \cdot \hat{\mathbf{n}}_{c_1} + f_l \qquad (4.21) \\ f_l &= -((\mathbf{f}_s + m\mathbf{p}_l) \cdot \hat{\mathbf{n}}_{c_1} + m\mathbf{v}_{c_1} \cdot \dot{\hat{\mathbf{n}}}_{c_1}). \qquad (4.22) \end{aligned}$$

Now if the vector $\mathbf{b}$ is defined as:

$$\mathbf{b} = -m\mathbf{P}^T\hat{\mathbf{n}}_g$$

then:

$$-m\mathbf{P}\ddot{\boldsymbol{\theta}} \cdot \hat{\mathbf{n}}_g = -m\ddot{\boldsymbol{\theta}} \cdot \mathbf{P}^T\hat{\mathbf{n}}_g = \ddot{\boldsymbol{\theta}} \cdot \mathbf{b}$$

Also define the vector $\mathbf{a}$ as:

$$\mathbf{a} = (\mathbf{x}_b - \mathbf{r}_{cm}) \times \left[\mathbf{R}^T \left[\hat{\mathbf{n}}_{c_1} - \mu \frac{\mathbf{v}_{c_1}}{\|\mathbf{v}_{c_1}\|}\right]\right]$$

so $(\mathbf{x}_b - \mathbf{r}_{cm}) \times (\mathbf{R}^T \mathbf{f}_1) = f_n \mathbf{a}$. But $f_n = \ddot{\boldsymbol{\theta}} \cdot \mathbf{b} + f_l$ so $f_n \mathbf{a} = \mathbf{a}(\ddot{\boldsymbol{\theta}} \cdot \mathbf{b}) + f_l \mathbf{a}$. This defines the second order tensor $\mathbf{A}$ with:

$$\mathbf{A} = \begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix}$$

so that $\mathbf{a}(\ddot{\boldsymbol{\theta}} \cdot \mathbf{b}) = \mathbf{A}\ddot{\boldsymbol{\theta}}$. Substituting these results into equation 4.14 produces:

$$(\mathbf{I}_n \mathbf{R}^T \mathbf{S} - \mathbf{A})\ddot{\boldsymbol{\theta}} = \mathbf{R}^T \boldsymbol{\tau}_s - \mathbf{r}_{cm} \times (\mathbf{R}^T \mathbf{f}_s) + f_l \mathbf{a} - \mathbf{n}_l \tag{4.23}$$

and the desired equation for $\ddot{\boldsymbol{\theta}}$ is derived. $\ddot{\mathbf{x}}$ is given by substituting equation 4.21 into equation 4.18 and substituting this result into equation 4.14 to get:

$$\ddot{\mathbf{x}} = \frac{1}{m}\mathbf{f}_s + \frac{1}{m}(\ddot{\boldsymbol{\theta}} \cdot \mathbf{b} + f_l)\left[\hat{\mathbf{n}}_g - \mu \frac{\mathbf{v}_{c_1}}{\|\mathbf{v}_{c_1}\|}\right] - m(\mathbf{P}_{cm}\ddot{\boldsymbol{\theta}} + \mathbf{p}_{l_{cm}}). \tag{4.24}$$

### 4.4.6 Multiple Point Contact

As was stated earlier, we only will consider the case of motion for all multiple points of contact sliding. We show a method for deriving the equations of motion for $n$ contacts with $n$ velocity constraints. For two points of contact this method requires the inversion of an 8x8 matrix. For $n$ contacts a square matrix of dimension $6 + n$ must be inverted.

It is convenient to define the vector $\hat{\mathbf{f}}_i$ to denote the direction of the reaction force applied by the nut to the bolt. This is *not* a unit vector, but is defined as:

$$\hat{\mathbf{f}}_i = \hat{\mathbf{n}}_{c_i} - \mu \frac{\mathbf{v}_{c_i}}{\|\mathbf{v}_{c_i}\|} \tag{4.25}$$

so that

$$\mathbf{f}_i = f_{n_i} \hat{\mathbf{f}}_i$$

where $f_{n_i}$ is the magnitude of the force applied in the direction of the constraint at contact $i$.

The velocity and acceleration constraints are given by:

$$\mathbf{v}_{c_i} \cdot \hat{\mathbf{n}}_{c_i} = 0 \quad , \quad \dot{\mathbf{v}}_{c_i} \cdot \hat{\mathbf{n}}_{c_i} + \mathbf{v}_{c_i} \cdot \dot{\hat{\mathbf{n}}}_{c_i} = 0. \tag{4.26}$$

Recall:

$$\dot{\mathbf{v}}_{c_i} = \ddot{\mathbf{x}} + \mathbf{P}_{rb_i}\ddot{\theta} + \mathbf{p}_{l_{rb_i}}.$$

Now equations 4.14 and 4.15 and the constraint equations 4.26 for a two point contact can be described by the 8x8 matrix equation:

$$\begin{bmatrix} \mathbf{I}_n\mathbf{R}^T\mathbf{S} & \mathbf{0} & -[\mathbf{x}_b(\xi_{b_1}) - \mathbf{r}_{cm}] \times \mathbf{R}^T\hat{\mathbf{f}}_1 & -[\mathbf{x}_b(\xi_{b_2}) - \mathbf{r}_{cm}] \times \mathbf{R}^T\hat{\mathbf{f}}_2 \\ m\mathbf{I}_{3\times 3} & m\mathbf{P}_{cm} & -\hat{\mathbf{f}}_1 & -\hat{\mathbf{f}}_2 \\ (\mathbf{P}_{rb_1}^T\hat{\mathbf{n}}_{c_1})^T & \hat{\mathbf{n}}_{c_1} & 0 & 0 \\ (\mathbf{P}_{rb_2}^T\hat{\mathbf{n}}_{c_2})^T & \hat{\mathbf{n}}_{c_2} & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\mathbf{x}} \\ f_{n_1} \\ f_{n_2} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{R}^T\boldsymbol{\tau}_s - \mathbf{r}_{cm} \times (\mathbf{R}^T\mathbf{f}_s) - \mathbf{n}_l \\ \mathbf{f}_s - m\mathbf{p}_{l_{cm}} \\ -\mathbf{p}_{l_{rb_1}} \cdot \hat{\mathbf{n}}_{c_1} - \mathbf{v}_{c_1} \cdot \dot{\hat{\mathbf{n}}}_{c_1} \\ -\mathbf{p}_{l_{rb_2}} \cdot \hat{\mathbf{n}}_{c_2} - \mathbf{v}_{c_2} \cdot \dot{\hat{\mathbf{n}}}_{c_2} \end{bmatrix} \tag{4.27}$$

## 4.5 Impulses: Instantaneous Change in Momentum

One of the most interesting physical problems in simulating the motion of an object under stiffness control is what happens to the velocity of the object at the instant the motion constraint equation changes. The problem is that the velocity of the point of contact initially has a velocity into the surface, but after contact is made it does not. The question is, where does the momentum go? In this problem the inertia is assumed to be small so the amount of energy dissipated at the time contact is made should be small.

For example, suppose the bolt is rotating in free space with some velocity $\dot{\mathbf{c}}$. Suppose then that the bolt makes a one point contact with the nut. Before the contact $\mathbf{v}_{c_1} \cdot \hat{\mathbf{n}}_g \neq 0$. After the contact $\mathbf{v}_{c_1} \cdot \hat{\mathbf{n}}_g = 0$. In a method analogous to that presented for the dynamics of multiple point contact, the solution for the magnitude of $n$ impulses applied by an $n$ point contact can be described as the solution to a $3 + n$ matrix equation.

Each contact $i$ applies an impulse of magnitude $\lambda_i$ at the contact point $\mathbf{x}_b(\xi_{b_i})$ in the direction $\hat{\mathbf{n}}_{c_i}$. We wish to determine the magnitude of the impulses that cause a change of momentum which results in the motion of the bolt satisfying the constraint equations. Let $\dot{\bar{\mathbf{x}}}_{cm}$ and $\dot{\bar{\theta}}$ be the linear and rotational velocities at the center of mass after the impulses are applied. Then the initial and final momenta are related by:

$$m(\dot{\bar{\mathbf{x}}}_{cm} - \dot{\mathbf{x}}_{cm}) = \sum_i \lambda_i \hat{\mathbf{n}}_{c_i} \tag{4.28}$$

$$\mathbf{I}_n \mathbf{R}^T \mathbf{S}(\dot{\bar{\theta}} - \dot{\theta}) = \sum_i [\mathbf{x}_b(\xi_{b_i}) - \mathbf{r}_{cm}] \times \lambda_i \mathbf{R}^T \hat{\mathbf{n}}_{c_i} \tag{4.29}$$

To reduce the order of the system of equations to be solved, $\dot{\bar{\mathbf{x}}}$ is eliminated by the use of:

$$\dot{\bar{\mathbf{x}}} = \dot{\mathbf{x}} + \mathbf{P}_{cm}(\dot{\theta} - \dot{\bar{\theta}}) + \frac{1}{m} \sum_i \lambda_i \hat{\mathbf{n}}_{c_i}$$

and the velocity constraints:

$$\bar{\mathbf{v}}_{c_i} \cdot \hat{\mathbf{n}}_{c_i} = (\dot{\bar{\mathbf{x}}} + \mathbf{P}_{xb_i} \dot{\bar{\theta}}) \cdot \hat{\mathbf{n}}_{c_i} = 0.$$

Thus there are $n$ equations of which the $j^{th}$ is:

$$\left[ \dot{\mathbf{x}} + \mathbf{P}_{cm} \dot{\theta} + (\mathbf{P}_{xb_j} - \mathbf{P}_{cm}) \dot{\bar{\theta}} + \frac{1}{m} \sum_i \lambda_i \hat{\mathbf{n}}_{c_i} \right] \cdot \hat{\mathbf{n}}_{c_j} = 0. \tag{4.30}$$

So for a two point contact, the impulses and final velocities are given by the solution to:

$$\begin{bmatrix} \mathbf{I}_n \mathbf{R}^T \mathbf{S} & -(\mathbf{x}_b(\xi_{b_1}) - \mathbf{r}_{cm}) \times \mathbf{R}^T \hat{\mathbf{n}}_{c_1} & -(\mathbf{x}_b(\xi_{b_2}) - \mathbf{r}_{cm}) \times \mathbf{R}^T \hat{\mathbf{n}}_{c_2} \\ \left[(\mathbf{P}_{xb_1} - \mathbf{P}_{cm})^T \hat{\mathbf{n}}_{c_1}\right]^T & \frac{1}{m} & \frac{1}{m} \hat{\mathbf{n}}_{c_1} \cdot \hat{\mathbf{n}}_{c_2} \\ \left[(\mathbf{P}_{xb_2} - \mathbf{P}_{cm})^T \hat{\mathbf{n}}_{c_2}\right]^T & \frac{1}{m} \hat{\mathbf{n}}_{c_1} \cdot \hat{\mathbf{n}}_{c_2} & \frac{1}{m} \end{bmatrix} \begin{bmatrix} \dot{\bar{\theta}} \\ \lambda_1 \\ \lambda_2 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{I}_n \mathbf{R}^T \mathbf{S} \dot{\theta} \\ -(\dot{\mathbf{x}} + \mathbf{P}_{cm} \dot{\theta}) \cdot \hat{\mathbf{n}}_{c_1} \\ -(\dot{\mathbf{x}} + \mathbf{P}_{cm} \dot{\theta}) \cdot \hat{\mathbf{n}}_{c_2} \end{bmatrix} \tag{4.31}$$

# Chapter 5

# Simulation of Motion Under Stiffness Control

The results developed in the previous two chapters were used to create a program producing an animated simulation of a bolt threading into a nut. Figure 5.1 shows a sequence of frames from the simulation. This simulation is used to test various grasp stiffnesses to determine their performance. Performance is defined by the size of the initial error that allows the insertion to proceed correctly.

## 5.1 Methodology

The program, which is listed in Appendix C, continually loops through two procedures: a state update routine and a contact configuration checking routine. This is shown in Figure 5.2. The reason for decoupling the configuration analysis from the updating, which is essentially integration of the dynamics, is to keep the speed of the simulation reasonable. The sacrifice in accuracy is small as the configuration update routine backtracks along the trajectory to find the point of time at which the configuration changed. The appropriate dynamic equation is then integrated from that point on.

Besides these two main procedures, there is one procedure, *set_stiffness_position*, that returns the equilibrium position of the bolt at time $t$. To run a simulation the user modifies the file *stiff.c* which contains the equilibrium position update routine, as well as the function that returns the stiffness force as a function of the current configuration and the equilibrium configuration. This allows a great deal of flexibility in the choice of force

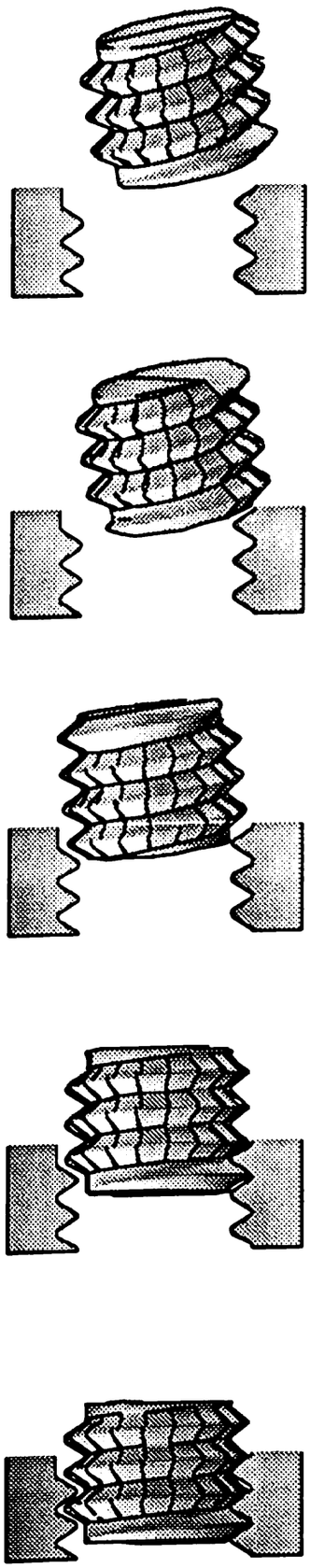

Figure 5.1: Frames from motion simulation.

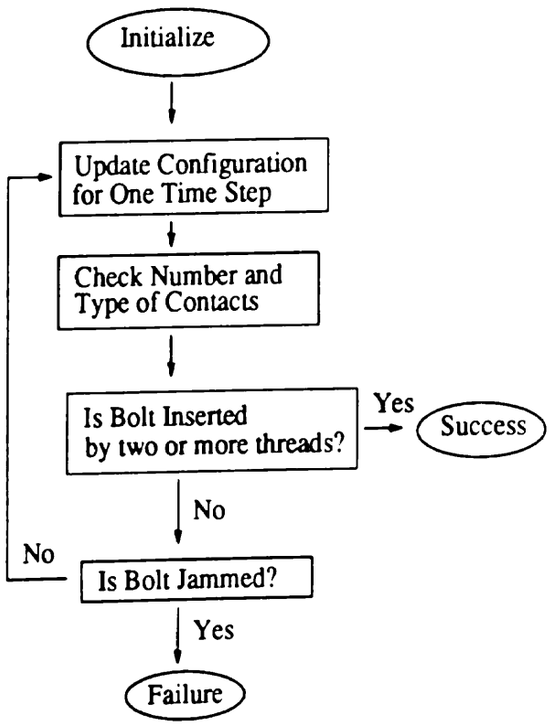


Figure 5.2: Simulation program structure

controls to simulate as both the path of the equilibrium position and the stiffness function may be specified.

### 5.1.1 The Equilibrium Configuration

The equilibrium configuration at time t is determined by the C function, *set_stiffness_position*. This procedure computes the equilibrium configuration based on the current time of the simulation. Time is altered by the two main routines, the update routine and the contact configuration analysis routine.

For all of the simulations discussed in this thesis, the equilibrium configuration was given as:

$$\mathbf{c}_e(t) = \mathbf{c}_e(0) + \dot{\mathbf{c}}_e t \quad , \quad \dot{\mathbf{c}}_e = \begin{bmatrix} 0 \\ 0 \\ \dot{x}_{z_e} \\ 0 \\ 0 \\ 2\pi \frac{\dot{x}_{z_e}}{p} \end{bmatrix} \tag{5.1}$$

So the control objective is to find the stiffness function that guarantees successful insertion for the largest range of $\mathbf{c}_e(0)$.

### 5.1.2 Switching Contact Configurations

This routine tests the current configuration of the bolt to determine the number and type of contacts. If the configuration has changed the bolt will have most likely gone too far and the contact points have ended up inside, instead of on the surface of the nut. Thus the trajectory is followed backwards to find the point of time during the last update at which the configuration changed. Figure 5.3 gives a flow outline of this routine.

#### Determining the number of contacts

Using the helical approximation discussed in section 3.4.2 with a resolution of 400 points per revolution of the helix, we check to see how many of these points lie inside either the surface of the nut or the surface of the bolt. If none do, then there are no *new* contacts, however the helical approximation is only a sufficient but not a necessary condition for contact. Hence if there are no intersections detected, the contact configuration is unaffected by the intersection detection routine.

If intersections are detected, then they are grouped into clusters of points. A cluster is determined by a metric on the surface parameter space. For this simulation points were determined to be in the same cluster if they differed in their angle by less than 50 degrees and differed in their z height by less than one half of a pitch. The angle may seem high, but recall that the two surfaces have close to, but not exactly, the same curvature, hence one actual contact point may be evidenced by many intersection points on the helix.

Once the number of clusters has been determined for both the bolt and nut helixes, the values for z and theta in the cluster are averaged to find an approximate location for

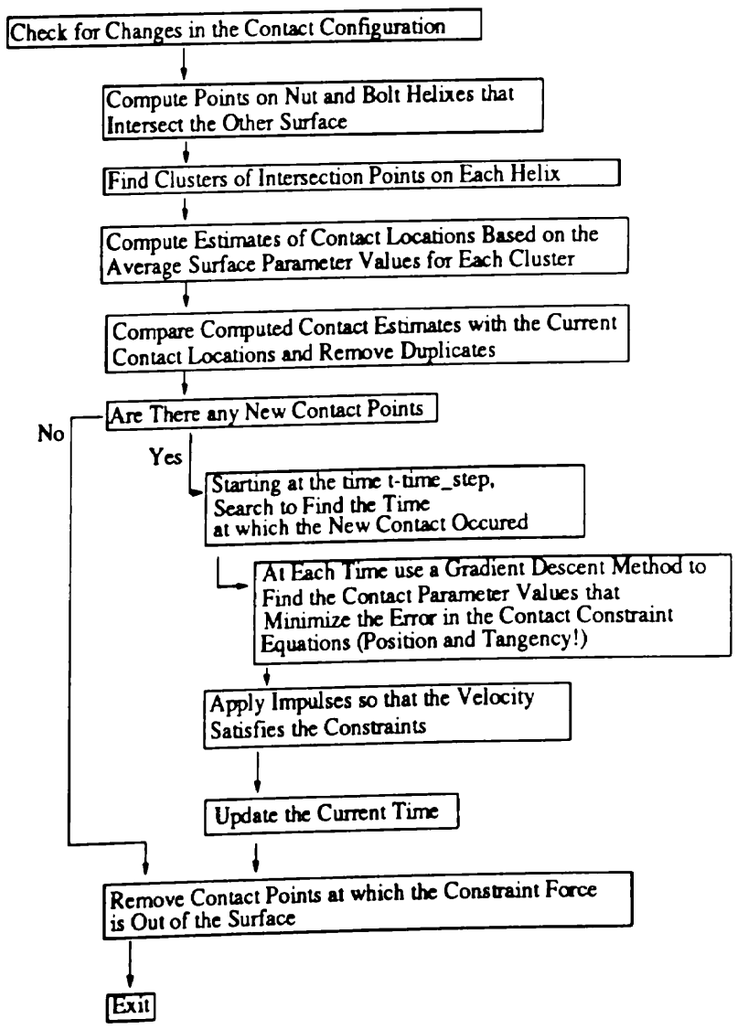


Figure 5.3: Configuration Analysis Procedure

the contact point. The derived contact points for the bolt are then mapped to inertial space to be compared with the derived contact points for the nut. Again using the same metric for determining if two points correspond to the same contact point, the derived nut contact points that correspond to the same derived contact points found for the bolt are deleted. Finally the bolt contact points that correspond to contact points already known to the simulation are deleted. The remaining derived contact points are determined to be new and are added to the list of known contact points.

### Backing up

If there are any new contact points then we must find the point of time during the last update at which the new contact occurred. This was one of the more difficult parts of the program to get right. The problem is to find the point in time at which the error to the contact constraint equations, 3.3 and 3.2, are minimized. But how, for a particular choice of the configuration, **c**, do we find the contact parameters $\xi_{b_i}$ and $\xi_{g_i}$ that minimize the error? The method that worked best was to use a nonlinear equation solver at each of 100 points evenly divided over the last time step. The nonlinear solver would find the best fit for the surface parameters. The point which gave the minimum error was then chosen as a starting point for the nonlinear solver to fit the time parameter as well as the contact parameters. Due to local minima and saddle points, however, the extra degree of freedom in solving for the solution often led to highly erroneous results which were discarded.

The key point is that the surfaces under consideration are quite complex, with both convex and concave parts. This makes it hard for a standard gradient descent equation solver to get the correct solution and avoid the local minima. Certainly one point for future research is a better method for finding the contact points exactly.

In the simulation implemented, the time step between contact configuration checks was made small enough so that backtracking was held to one half of a time step. Typical rms errors in the solution ranged from a low of 1e-8 to a high of 0.1 thread pitches.

Once the minimum error point has been found, the velocity of the bolt must be changed to reflect the new constraint on motion given by the *new* contact point. Section 4.5 discusses how this is done.

Any change in time is passed back to the calling function, allowing the equilibrium configuration to be updated accordingly.

### Removing Contacts

As a last task, the contact configuration update routine removes from its list of known contact points those at which the constraint force has been determined to point out of the surface. The constraint force is set by the integration routine.

### 5.1.3 Integrating the State

Updating the state is based on integrating the equations of motion over a designated time period assuming that the contact configuration does not change over this period. Numerical integration of the dynamic equations is accomplished by using a variable step fourth order runge-kutta algorithm. The variable step size is desirable due to the changing configurations and hence equations. The evaluation of the state derivative involves solving various square and rectangular matrix equations. The Linpack fortran libraries are used for this. A library of matrix and vector functions was created to compute the equations.

Figure 5.4 shows an outline of the procedure calls used to integrate the state. The rectangular contact evolution matrices discussed in section 3.6 are solved by the generalized inverse technique. A fortran library routine was used for this. The square matrix equations are solved by using the lower and upper triangular factorization of the matrices. Again a fortran Linpack library routine is used.

There is one file for each contact configuration state: no contact, one point contact, two point contact, and three point contact. Each file contains a function that calls the runge-kutta integration routine with the appropriate parameters, and a function that evaluates the derivative of the state given the current state and time.

All the matrix and vector computations to create the matrices that are passed to the matrix inversion routines are accomplished with a library of matrix and vector function routines. The file *mat_l.c* contains these. The file *screw.c* contains all the routines necessary to describe the geometry of the screw threads.

## 5.2 Simulations

At this point we can begin to investigate the relations between stiffness matrices and motion of threaded fasteners under their control. Our investigation will focus on RCC type devices. A suggestion by [Loncaric 87] will be considered as an addition to the RCC

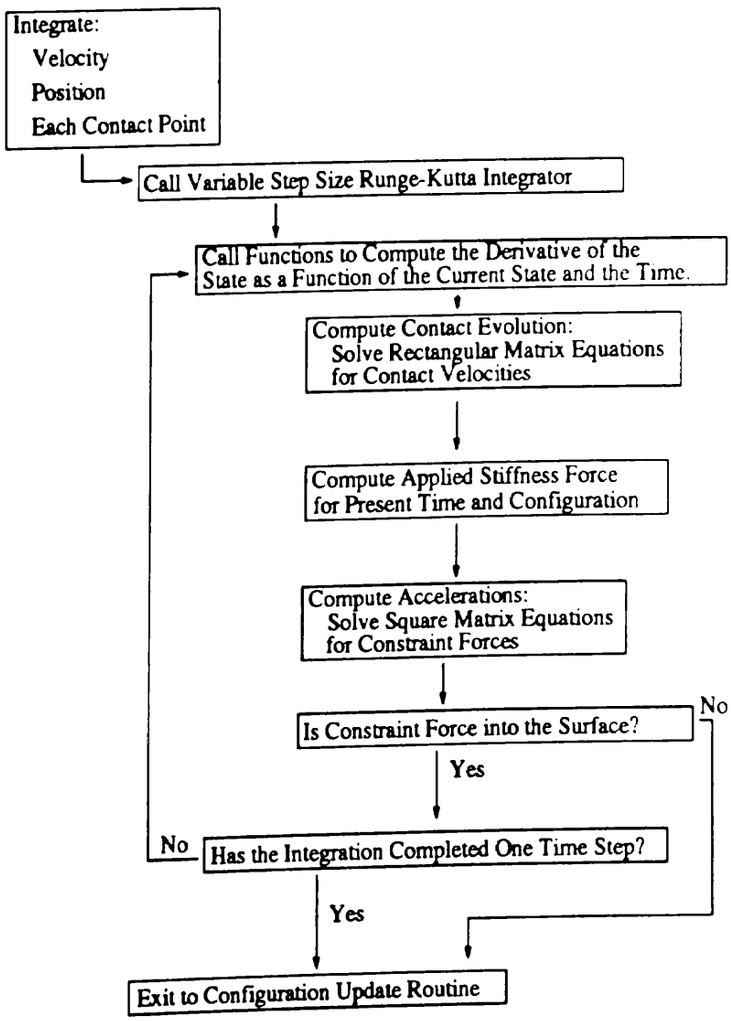


Figure 5.4: Integration Procedure

control.

### 5.2.1 Whitney's RCC

The remote center of compliance device proposed by Whitney may be modeled by a diagonal stiffness matrix defined at some center of compliance on the bolt. The center of compliance is defined as the point about which translations without rotations from the center result only in forces and rotations without translations result only in torques. If $\mathbf{K}$ is diagonal, then the center of compliance lies at the origin of the bolt. In this simple case one can think of the object as held by springs along each translational and rotational axis such that displacement from the equilibrium configuration results in linear restoring forces and torques. If $\mathbf{K}$ is positive definite then the grasp is stable [Salisbury 82].

The location of the center of compliance is an important parameter to choose for a grasp stiffness. Whitney [Nevins 80] showed how the peg-in-hole insertion failed for compliance centers at one end of the peg, but success was achieved for centers at the peg and hole interface. Thus we now derive the functions $\mathbf{f}_s$ and $\boldsymbol{\tau}_s$, which give the force and torque in inertial coordinates about the origin of the bolt, to give the correct forces for a compliance center located at an arbitrary point in the bolt body coordinates.

Let the vector $\mathbf{r}_c$ denote the center of compliance in the bolt body frame. Let $\mathbf{K}$ be the stiffness matrix at this point. Now the equilibrium position for the center of compliance will be given by $\hat{\mathbf{x}}_e$ with:

$$\hat{\mathbf{x}}_e = \mathbf{x}_e + \mathbf{R}(\boldsymbol{\theta}_e)\mathbf{r}_c.$$

Similarly the current location of the center of compliance is given by:

$$\hat{\mathbf{x}} = \mathbf{x} + \mathbf{R}(\boldsymbol{\theta})\mathbf{r}_c.$$

Now the force applied at $\hat{\mathbf{x}}$ is:

$$\mathbf{f}_s(\mathbf{c}, \mathbf{c}_e) = \mathbf{K}_x(\hat{\mathbf{x}}_e - \hat{\mathbf{x}}) \tag{5.2}$$

where $\mathbf{K}_x$ is the upper left 3x3 matrix of $\mathbf{K}$. It is important to note the dependence of $\hat{\mathbf{x}}_e$ on $\boldsymbol{\theta}_e$. This causes cross terms in the stiffness. That is, a pure rotation about the bolt origin now results in a restoring force as well as a torque.

The restoring torque has a similar cross term due to the fact that $\mathbf{f}_s$ acts at $b\hat{f}x$ and not at $\mathbf{x}$. Thus $\boldsymbol{\tau}_s$ is composed of two parts: a torque, or couple, due to rotational

displacement from the equilibrium orientation and a torque, or couple, due to the force applied at $\hat{\mathbf{x}}$. Note that as the moment of a couple is independent of the choice of the point of rotation and that the change in stiffness is purely a translation, the torque is not transformed by the change in location of the center of compliance. Thus

$$\tau(\mathbf{c}, \mathbf{c}_e) = \mathbf{K}_\theta(\boldsymbol{\theta}_e - \boldsymbol{\theta}) + (\mathbf{R}(\boldsymbol{\theta})\mathbf{r}_c) \times \mathbf{f}_s(\mathbf{c}, \mathbf{c}_e) \tag{5.3}$$

It is interesting to note that now the restoring force and torque are nonlinear functions of the present configuration and the equilibrium configuration.

Due to the axial symmetry of the bolt and the assumed lack of knowledge about the starting point of the thread, the desired center of compliance must lie along the axis of the bolt. Hence the the desired $\mathbf{r}_c$ must be of the form:

$$\mathbf{r}_c = \begin{bmatrix} 0 \\ 0 \\ z_c \end{bmatrix}. \tag{5.4}$$

If we restrict our search for stiffness matrix solutions to compliances, then search space for linear stiffnesses is reduced further as a compliance matrix is diagonal at the center of compliance. So by symmetry arguments,

$$\mathbf{K} = \begin{bmatrix} k_t & 0 & 0 & 0 & 0 & 0 \\ 0 & k_t & 0 & 0 & 0 & 0 \\ 0 & 0 & k_z & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{\theta_t} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{\theta_t} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{\theta_z} \end{bmatrix}.$$

Thus the problem of finding the optimal RCC type device has been reduced to finding 5 parameters, $(k_t, k_z, k_{\theta_t}, k_{\theta_z}, z_c)$.

Table 5.1 lists simulation parameters that were tried. In reference to the number of each simulation, the results were:

1. RCC with center of compliance beneath bolt, 180 degree phase offset: the run-up of the bolt's thread ran up against that of the nut and caused the bolt to twist out of the nut.

| | Initial Position, angles in degrees | | | | | | Stiffness | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $c_{e_x}$ | $c_{e_y}$ | $c_{e_z}$ | $c_{e_{\theta_x}}$ | $c_{e_{\theta_y}}$ | $c_{e_{\theta_z}}$ | $z_c$ | $k_t$ | $k_z$ | $k_{\theta_t}$ | $k_{\theta_z}$ |
| 1 | 0.5 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | -3.0 | 0.1 | 0.1 | 10.0 | 10.0 |
| 2 | 0.5 | 0.0 | 0.0 | 0.0 | 1.0 | 90.0 | -3.0 | 0.1 | 0.1 | 10.0 | 10.0 |
| 3 | 0.5 | 0.0 | 0.0 | 0.0 | 1.0 | 180.0 | -3.0 | 0.1 | 0.1 | 10.0 | 10.0 |
| 4 | 0.0 | 1.0 | 0.5 | 1.0 | 5.0 | 180.0 | 3.0 | 1.0 | 1.0 | 0.01 | 1.0 |

Table 5.1: Bolt behavior under RCC Control

2. Center of compliance beneath bolt: 90 phase degree offset: Again the bolt did not start threading.

3. Center of compliance beneath bolt: no phase offset: The bolt tipped out of the hole after its run-up contacted the first full thread of the nut and caused a large orientational error.

4. This last simulation was designed to correspond to the grasp stiffness and initial conditions for the screwing operation implemented with a robotic hand with two soft fingers. The bolt tipped over and failed to be inserted, much like the case with the robotic hand. Pictures of the robotic hand and the simulation are given in Appendix A.

*Phase*, as referred to above, indiactes the number of degrees the start of the thread on the nut and bolt are apart. No phase offset means that with no errors in translation or aligment, the bolt should be inserted without contacting the nut. This phase parameter seems to be one of the most critical in determining if propoer insertion will occur. With 180 degrees of offset, then the bolt will tilt when contact is first made, causing a bad insertion. Even if the bolt is in phase, due to the simple trajectory used, it will tilt, if the orientaional stiffness is low enough, before the threads start to mate. The tilting occurs as the thread run-up on the bolt touches the beginnnings of the first complete thread on the nut. An improved trajectory migh be to drop the bolt into the hole first and use the human "backing off" heuristic discussed in section 2.1.2 to algin the bolt.

In addition to the simulations listed above, many others were carried out with orientation errors of 5 degrees and varying stiffnesses. The results of these simulations were inconclusive as the simulation terminated due to large errors in contact locations before it could be determined if cross threading had occured. In many cases it appeared that the final configuration was one from which a succesful threading operation could not occur.

At this point no firm conclusions can be made about the merits of RCC stiffness control for screw threading. Not enough simulations have been run to say conclusively that a certain compliance center or stiffness matrix will guarantee insertion. However the simulations did indicate that a center of compliance below the peg was desirable. They also showed that screw threading is highly sensitive to orientational errors. With almost perfect alignment, matched starting threads, and small position displacements the bolt could be made to insert with a large orientational stiffness and a small translational stiffness, with the center of compliance beneath the bolt.

### 5.2.2 Loncaric's Solution

In [Loncaric 87] a solution for a screw threading stiffness matrix is given as:

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_z & 0 & 0 & k_{z\theta_z} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{z\theta_z} & 0 & 0 & k_{\theta_z} \end{bmatrix}.$$

The idea of this stiffness is to couple displacements in the free direction, $z$, to displacements in the free direction $\theta_z$. So if the bolt is, say, 180 degrees out of phase with the nut, then the stiffness will cause an appropriate torque to cause the stiffness center to "catch up." In detail, that is, suppose the start of the thread on the bolt is 180 degrees away from the start of the thread of the nut and that the bolt is otherwise *perfectly* aligned with the nut. In this case there will always be a bias force when the threads start to mate as the equilibrium $z$ position will be one half of a thread pitch from the correct position for the nut and bolt not to touch. By adding the cross term $k_{z\theta_z}$ a displacement in $z$ causes an additional torque about the $z$ axis to get the bolt to "catch up."

It is not clear why Loncaric did not specify stiffnesses for the other directions. The first simulation in table 5.2 was of just the stiffness he specifies. It failed. Hence an RCC device was modeled with the addition of the cross term $k_{z\theta_z}$ proposed by Loncaric. The results are shown in table 5.2.

1. Lonacaric's stiffness alone: the bolt flops over.

| | Initial Position, angles in degrees | | | | | | Stiffness | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\mathbf{c}_{e_x}$ | $\mathbf{c}_{e_y}$ | $\mathbf{c}_{e_z}$ | $\mathbf{c}_{e_{\theta_x}}$ | $\mathbf{c}_{e_{\theta_y}}$ | $\mathbf{c}_{e_{\theta_z}}$ | $z_c$ | $k_t$ | $k_z$ | $k_{\theta_t}$ | $k_{\theta_z}$ | $k_{z\theta_z}$ |
| 1 | 0.5 | 0.0 | 0.0 | 0.0 | 1.0 | 180.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 1.0 |
| 2 | 0.5 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | -3.0 | 0.1 | 0.1 | 10.0 | 10.1 | 1.0 |
| 3 | 0.5 | 0.0 | 0.0 | 1.0 | 0.0 | 90.0 | -3.0 | 0.1 | 0.1 | 10.0 | 10.1 | 1.0 |
| 4 | 0.5 | 0.0 | 0.0 | 1.0 | 0.0 | 180.0 | -3.0 | 0.1 | 0.1 | 10.0 | 10.1 | 1.0 |

Table 5.2: RCC with Loncaric's $k_{z\theta_z}$ term.

2. Loncaric's stiffness with RCC, 180 degrees out of phase with nut thread: The bolt aligns to the nut.

3. Loncaric's stiffness with RCC, 90 degrees out of phase with nut thread: the simulation failed to give a conclusive result due to early termination.

4. Loncaric's stiffness with RCC, in phase with nut thread: after some initial jostling the bolt threads correctly.

The addition of Lonaric's stiffness seems to help the bolt get in phase with the nut. The above simulations, with 1 degree orientational errors, started with varying offsets from the correct "phasing" of the threads. When the bolt was started such that there was no phase offset, the insertion proceeded much better. Loncaric's $k_{z\theta_z}$ term helped to reduce this phase lag. Again, more simulations, with a larger range of intial conditions, must be made before any firm conclusion can be made.

# Chapter 6

# Extensions and Conclusions

In this thesis a comprehensive method for analyzing the motion of a bolt threading into a nut has been presented. At this point we cannot say we have a guaranteed solution, but the simulations have pointed to important features of stiffness matrices for consideration.

The analysis began with an overview of current threaded fastener technology. We next developed a mathematical model for threaded parts and analyzed the configuration space. Next quasistatics, dynamics, and impulses were used to determine the equations of motion. These equations were compiled into a simulation program and combined with the geometric model to simulate the motion under contact. No approximations were made in the theory, however numerical inaccuracies in the calculations might cause disagreement between the simulation and actual threading operations.

There are many areas for further research. In the field of threaded fasteners more control schemes could be considered. Also redesign of threaded fasteners, especially the chamfer section, could improve insertion performance. The nut modeled in this thesis did not have a chamfer, but it was observed in the simulations that it would have certainly helped align the bolt.

In the area of simulations, the method used here could be generalized to handle arbitrary objects and thus allow the modeling of other part mating problems. In doing this, research into contact localization and tracking would be important. In this simulation the contact points were integrated, which works fine if the surfaces are smooth and the contact velocity is low. It would be advantageous to be able to solve exactly for the contact points. In the simulation described here, the gradient descent approach was tried at each point in

the integration with catastrophic results as the contact points varied wildly, causing all the other variables, including the constraint direction, to vary wildly as well.

We hope this marks the beginning of a more in depth analysis of threaded fastening by the academic community. Threaded fasteners are one of the most important parts of assembly. Without understanding of how to mate them automatically we cannot expect to develop a fully autonomous assembly system.

for number sequence only

# Appendix A

# Experimental Work

The RobotWorld system manufactured by Automatix was used to demonstrate the problems with open loop threading with dextrous hands. A soft rubber fingertip was attached to each of the two manipulator modules of the RobotWorld system. The finger tips were position controlled using a simple discrete time PD controller.

A simple task was created in which the two fingers picked up a nut and placed it on the bolt. They then attempted to screw the nut onto the bolt using a simple position trajectory. Grasp forces were generated by setting the desired fingertip positions inside the nut and using the torque generated by the error in the PD controller. This was a rather crude implementation and worked only when the initial positions were exactly right. The following sequence of images shows a case in which the nut failed to be screwed onto the bolt.

In addition illustration A.6 shows images taken from the simulation program. The RobotWorld grasp was simulated with a weak orientational stiffness. As in the case with the nut, the initial translational displacement was sufficient to start the bolt off in the peg-like-contact phase. The rotational motion of the bolt then caused it to twist out. The simulation parameters used are in line 4 of Table 5.1.

Figure A.1: The start of the threading operation.

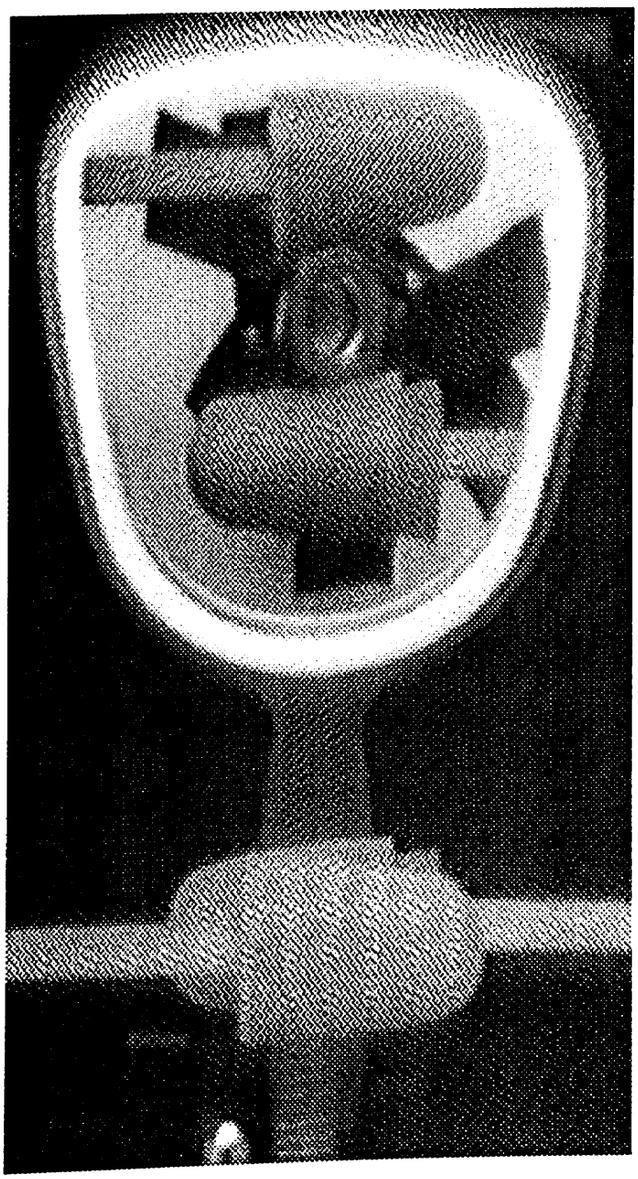

Figure A.2: The nut starts to tip.

Figure A.3: The nut is rotated about the head of the screw.

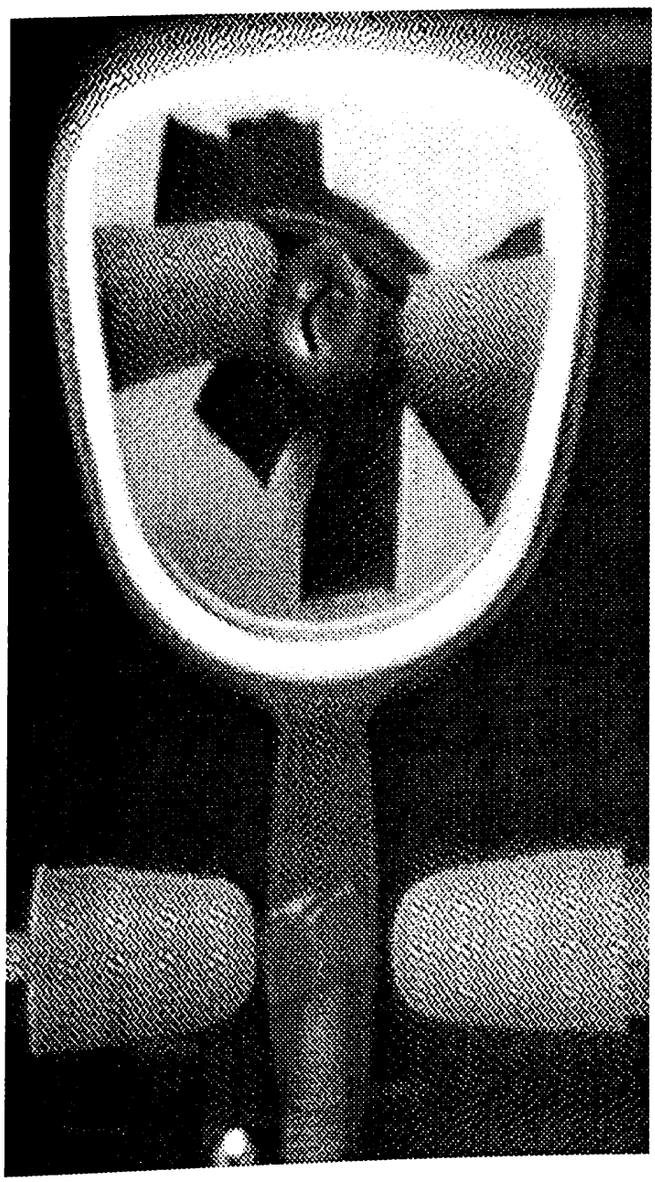

Figure A.4: The nut is precariously balanced.

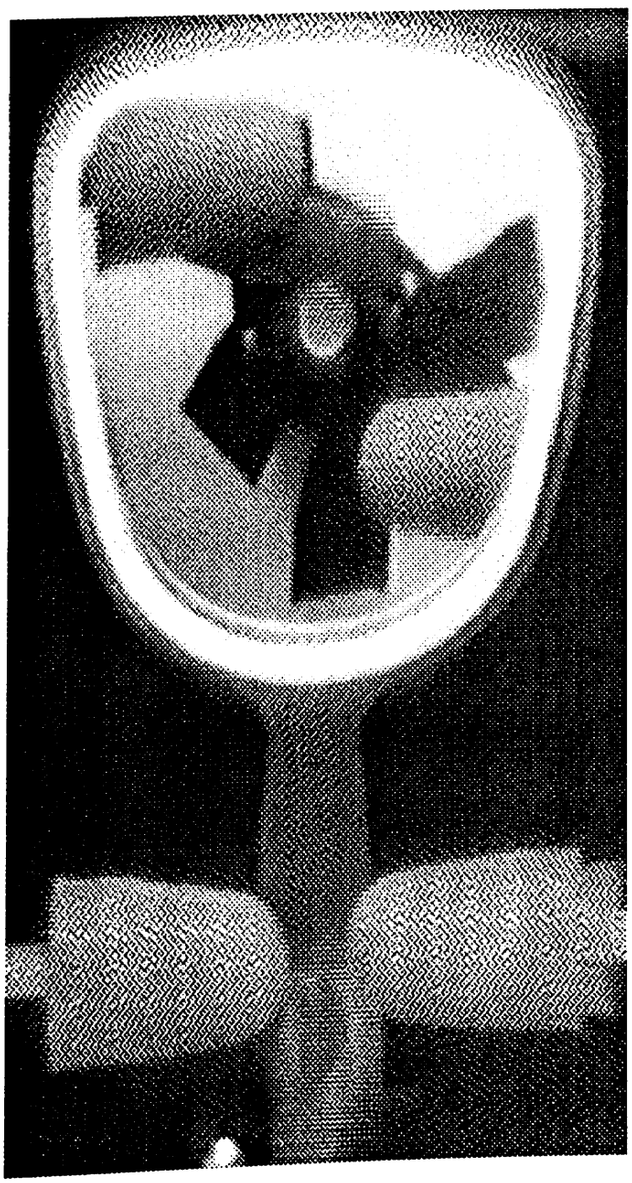

Figure A.5: The nut falls.

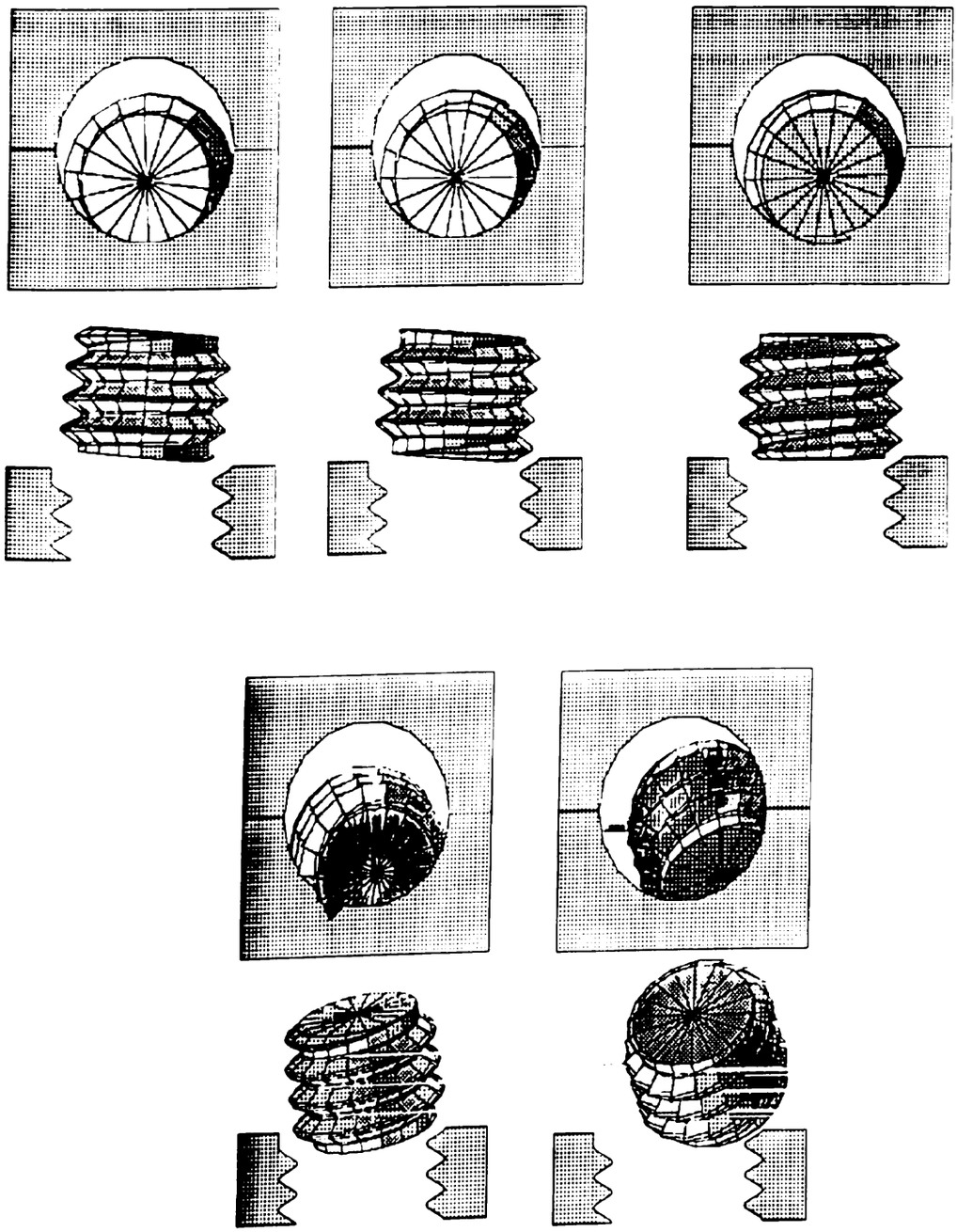

Figure A.6: Frames from the simulation.

# Appendix B

# Screw Thread Terminology

The following terms are useful when discussing threaded fasteners. They are summarized from [Blake 86] and [Bickford 81].

**screw thread:** a ridge of uniform section in the form of a helix.

**external screw thread:** the screw thread on a bolt or screw.

**internal screw thread:** the screw thread on a nut.

**thread profile:** the configuration of the thread in the axial plane.

**root:** the part of the profile at the smallest diameter.

**crest:** the part of the profile at the largest diameter.

**flank:** the straight part of the thread joining the roots and crests.

**incomplete thread:** the thread runout at the end of the nut or bolt where the threads run into the shaft.

**external thread major diameter:** the diameter at the crest of the external thread.

**external thread minor diameter:** the diameter at the root of the external thread.

**internal thread minor diameter:** the diameter at the crest of the internal thread.

**internal thread major diameter:** the diameter at the root of the internal thread.

**flank angle:** the angle between the flank and the axis perpendicular to the thread.

**symmetrical thread:** a thread for which both flank angles are the same. In most cases both flank angles are 30 degrees, then the thread profile is similar to a sequence of rounded equilateral triangles.

**fundamental triangle height:** the height of the thread if it were extended to a full V form. i.e. when it is not rounded or flattened at the crests and roots. Thus for a 60 degree symmetrical thread the height would be $\frac{\sqrt{3}}{2}P$ where $P$ is the thread pitch.

**pitch:** the distance measured parallel to the thread axis between corresponding points of the thread. The pitch may be either coarse or fine.

**interference:** if the external thread is larger or of a different profile or pitch from the internal member than interference will occur.

**allowance:** the amount by which the external thread diameter is decreased as compared to the internal thread. In this paper we will express the allowance as a fraction of the internal thread basic (minimum) major diameter.

**clearance fit:** a fit which permits free running assembly.

**interference fit:** a fit ensuring interference between the threads.

**length of thread engagement:** the distance through which complete threads are in contact.

**depth of thread engagement:** the amount of thread overlap.

**crossed-thread:** this occurs when the first external thread crosses the the internal thread in such a way that the thread engaged on one side of the internal thread is not on the same revolution as the thread engaged on the opposite side.

# Appendix C

# Source Code from Motion Simulation

The following is a list of the source code that was used to create the simulation. The files are: screw.h (header file), simul.c (the main loop), update.c (calls appropriate integration routine), context.c (set contact configuration), stiff.c (functions to do with control stiffness), screw.c (screw thread description functions, including normals and partial derivatives), contact.c (contact evolution, type checking), nocontact.c (integrate with no contact), one.c (integrate with one point contact), two.c (integrate with two point contact), three.c (integrate with three point contact), inter.c (nut and bolt intersection detection), mat_l.c (math library), grafix.c (commands for UNIGRAFIX), and runge_kutta.c (variable step runge-kutta integration).

In addition a *profile* of a run of the program is given at the end detailing the number of calls and cumulative time spent for each function.