Dynamic Modeling of a Part Mating Problem: Threaded Fastener Insertion*

Edward J. Nicolson and **Ronald S. Fearing**
Department of EE&CS
University of California
Berkeley, CA 94720

Abstract

A dynamic simulation of threaded insertion is developed based on Euler's equations, impulsive forces, and a geometric description for threaded parts. Points of contact between the threaded parts are determined and tracked during the simulation. Reaction forces are computed based on the contact locations and the velocity constraints given by the kinematic description. The simulation will be used to investigate the behavior of the bolt during insertion into the nut under grasp stiffness and damper control.

1 Introduction

According to a study by Nevins and Whitney [Nevins 80] the insertion and tightening of threaded fasteners is one of the twelve most common assembly tasks, yet little published work has considered the type of control best suited for the assembly of threaded parts. There are many situations in which an automatic method for fastener insertion would be desirable. Hazardous environments and assembly are prime candidates.

Although manufacturers would like to replace them with something easier to assemble [Allen 88], threaded fasteners are unique in their ability to be reassembled many times and to develop a variable preload, or force of assembly. These two attributes make it unlikely that threaded fasteners will be eliminated from assembly operations.

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. 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] 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. [Schimmels 90] presents a method for constructing a damping matrix for an insertion problem in which canonical configurations have been identified. This method can not be used until such canonical configurations are identified for the threaded fastener problem. In [Loncaric 87] a stiffness solution for threaded fasteners is suggested, without experimental verification, as an example of a stiffness control scheme.

[Blaer 62] provides guidelines for the rotational speed of a nut being fitted on to a bolt with a given axial stiffness when there are no orientation errors. In this restricted case, when errors are allowed only for the position along the axis of the bolt, the geometry may be analyzed in a planar fashion and the control problem is one dimensional. The main result showed that if the bolt rotates too fast for a given axial spring constant and position for the spring equilibrium point, then the nut will not begin to thread. This result is backed up by 1,000 trials with an experimental system.

[Smith 80] provides a good overview of automatic screwdriver technology, unfortunately it is now ten years old. The most interesting method described requires monitoring of the torque and angle about the axis of rotation as the bolt is inserted. The plot of torque versus angle, called the "fastening signature," is compared against signatures for proper and failed assemblies to determine if the insertion proceeded correctly. In this manner Smith claims the ability to discriminate proper fastening from thread stripping and thread crossing as well as detecting faulty fasteners.

Current automatic insertion methods do not guarantee successful insertion. Hence bolts are often started by hand and then tightened with a machine. Manual threading uses heuristics, such as rotating the fastener the wrong way for half a turn and then rotating in the correct direction, to ensure proper assembly.

This paper investigates a model for the motion of threaded fasteners during mating that allows force control schemes to be tested. In section 2 we give mathematical equations to describe the surfaces of threaded parts. With these equations the free directions of motion under contact, or kinematics, are determined by the set of possible contact types and the corresponding constraint directions. Given the motion constraints, the dynamic equations of motion for the bolt are derived in section 3. With the kinematic and dynamic models, a C language program was written to simulate the motion. Section 4 discusses the general outline of this program along with results of simulations of motion under grasp stiffness control.

*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.

Figure 1: A cross-threaded bolt and nut configuration.

2 Threaded Parts

2.1 Terminology

The discussion of threaded fasteners is facilitated with the introduction of some terms from [Blake 86] and [Bickford 81]. Figure 2 illustrates the most important ones.

A *screw thread* is a ridge of constant section, called the *thread profile*, wrapped in a helical fashion about a cylinder. The *pitch* is the spatial period of the thread profile. The *external screw thread* is the thread on a bolt and the *internal screw thread* is that on a nut. 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. If the thread were extended to a full V the *fundamental triangle height* would be reached. Instead it is rounded off or flattened at the roots and crests.

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 reduced as compared to the internal thread. This paper discusses the *allowance ratio* which expresses allowance as a fraction of the internal thread major diameter.

Threads do not start immediately on a nut 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 contacted on one side of the internal thread is not on the same revolution as the thread contacted on the opposite side. Figure 1 shows a bolt in a crossed thread configuration.

2.2 Functional Description

2.2.1 Thread Profile

Based on the definitions given in section 2.1, 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. The simulations in this paper use $r = \frac{p}{10}$.

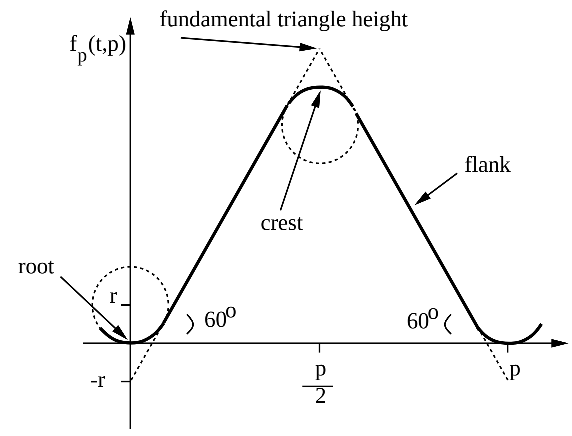

Figure 2: Thread profile.

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

The thread profile can be parameterized by the following function of position, t and thread pitch p. Figure 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.

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

2.2.2 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 θ is the amount of rotation.

A surface in $\Re^3$ may be parameterized by 2 variables. The parameters z and θ 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 1 was created using these functions. (In the figure $\frac{d}{p} = 6.0, \frac{r}{p} = 0.1, a = 0.05$.)

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

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

The descriptions include a linear thread run-up over 180 degrees. 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 $\mathbf{x}_b(z,\theta)$. 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.

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

2.2.3 Surface Normals

We denote the unit normals on the nut and bolt by the functions $\hat{\mathbf{n}}_g(z,\theta)$ and $\hat{\mathbf{n}}_b(z,\theta)$ respectively. Note that the surface normals vary continuously with the parameters z and θ, except, of course, at $z=0$. 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 $\hat{\mathbf{n}}_t(\theta_g,\theta_b)$ is defined to indicate the constraint direction. It is of unit magnitude and is in the direction of the cross product of the tangents to the circles at the base of the bolt and the top of the nut.

2.3 Contacts under Rigid Body Motion

Given models for the nut and bolt, we must next determine when and where the parts come into contact and what the constraints on motion are during contact.

2.3.1 Rigid Body Motion Notation

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 $\mathbf{c}$ as:

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

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:

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

Figure 3 shows the two frames and the coordinate axes. For contact locations, the following notation is used:

$$\mathbf{u}_{g_i} = \begin{bmatrix} z_{g_i} & \theta_{g_i} \end{bmatrix}^T \quad , \quad \mathbf{u}_{b_i} = \begin{bmatrix} z_{b_i} & \theta_{b_i} \end{bmatrix}^T .$$

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

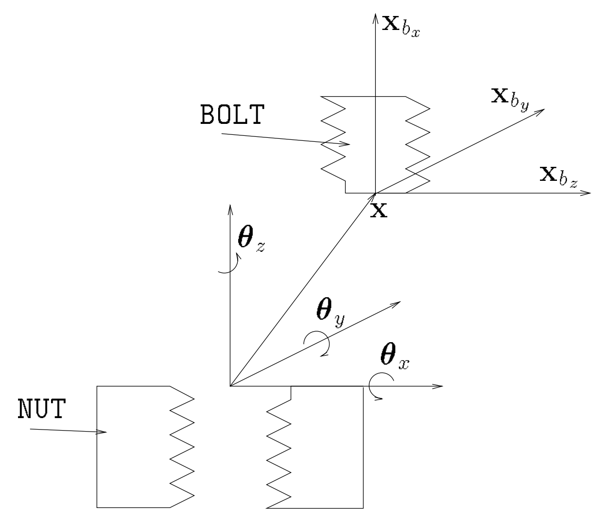

Figure 3: Rigid body notation.

$$\mathbf{x}_g(\mathbf{u}_{g_i}) = \mathbf{x} + \mathbf{R}(\boldsymbol{\theta})\mathbf{x}_b(\mathbf{u}_{b_i}) \tag{1}$$

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

$$\hat{\mathbf{n}}_g(\mathbf{u}_{g_i}) = -\mathbf{R}(\boldsymbol{\theta})\hat{\mathbf{n}}_b(\mathbf{u}_{b_i}) \tag{2}$$

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

2.3.2 Finding Contact Points

To determine if the nut and bolt are in contact when in a particular configuration $\mathbf{c}$, 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]. Unfortunately we do not have an implicit description for the bolt. In [Ponce 87] 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 argue that the contact location on at least one of the surfaces must be on a crest. From equation (2), we see that when contact occurs 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.

This observation leads to a simple method for surface intersection detection. Consider approximating the bolt by a helix with radius the same as that at the crest of the bolt thread. Given $\mathbf{c}$ we can determine if any points on this helix lie inside the surface of the nut by discretizing the helix. By using a few helixes spanning the rounded crest of the profile we can get a coarse estimate for the intersection points.

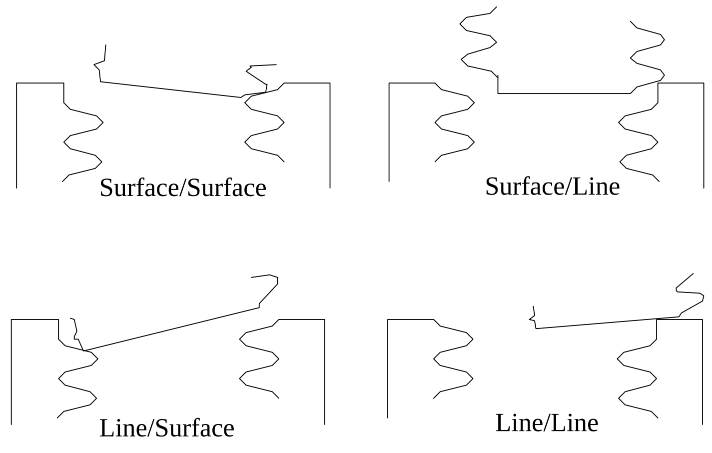

Figure 4: Illustrations of contact types.

These intersection estimates are improved by using a nonlinear algebraic equation solver.

2.3.3 Types of Contact

There are two different contact features that give rise to four possible contact types. 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 when this is not the case occurs when the nut and bolt are perfectly aligned. 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. The four contact types are shown in Figure 4.

2.4 Motion Constraint During Sliding

By differentiating the contact constraint equation (1) with respect to time we can define the quantity $\mathbf{v}_{c_i}$ [Cole 89]:

$$\mathbf{v}_{c_i} = \frac{\partial \mathbf{x}_g}{\partial \mathbf{u}_g}\dot{\mathbf{u}}_{g_i} - \mathbf{R}(\boldsymbol{\theta})\frac{\partial \mathbf{x}_b}{\partial \mathbf{u}_b}\dot{\mathbf{u}}_{b_i} = \dot{\mathbf{x}} + \dot{\mathbf{R}}(\boldsymbol{\theta})\mathbf{x}_b(\mathbf{u}_{b_i}) \tag{3}$$

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

$$\mathbf{v}_{c_i} \cdot \hat{\mathbf{n}}_{c_i}(\mathbf{u}_{g_i}, \mathbf{u}_{b_i}) = 0 \tag{4}$$

where the constraint direction $\hat{\mathbf{n}}_{c_i}$ is given in Table 1.

3 Dynamics Under Contact

3.1 The Force Controller

Assume a controller applies a force which is a function of the displacement from an equilibrium configuration and its instantaneous time derivatives. We denote the force applied by the controller in the inertial frame by $\mathbf{f}_a$. It is applied at the

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

Table 1: Constraint direction as determined by contact type.

point $\mathbf{r}_a$ in the bolt body frame. The inertial torque applied by the controller about $\mathbf{r}_a$ is denoted by $\boldsymbol{\tau}_a$. We denote the equilibrium configuration by $\mathbf{c}_e$. Thus a linear grasp stiffness and damper may be described by:

$$\mathbf{c}_e(t) = \mathbf{c}_e(0) + \dot{\mathbf{c}}_e t \tag{5}$$

$$\begin{bmatrix} \mathbf{f}_a(\mathbf{c}, \mathbf{c}_e, \dot{\mathbf{c}}, \dot{\mathbf{c}}_e) \\ \boldsymbol{\tau}_a(\mathbf{c}, \mathbf{c}_e, \dot{\mathbf{c}}, \dot{\mathbf{c}}_e)) \end{bmatrix} = \mathbf{K}(\mathbf{c}_e - \mathbf{c}) + \mathbf{B}(\dot{\mathbf{c}}_e - \dot{\mathbf{c}}) \tag{6}$$

where $\dot{\mathbf{c}}_e$ is constant, $\mathbf{K}$ is positive definite, and $\mathbf{B} \in \Re^{6\times6}$. Thus when the bolt is in the equilibrium configuration, and its velocity is the same as the nominal velocity, $\dot{\mathbf{c}}_e$, the force controller applies no force or torque.

3.2 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 $\boldsymbol{\tau}_i = (\mathbf{R}(\boldsymbol{\theta})\mathbf{x}_b(\mathbf{u}_{b_i})) \times \mathbf{f}_i$. For sliding contacts 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:

$$\hat{\mathbf{f}}_i = \hat{\mathbf{n}}_{c_i} - \mu\frac{\mathbf{v}_{c_i}}{\|\mathbf{v}_{c_i}\|} \tag{7}$$

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.

3.3 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:

$$\mathbf{s} = \begin{bmatrix} \mathbf{c} & \dot{\mathbf{c}} \end{bmatrix}^T. \tag{8}$$

[Montana 88] derives the equations for the derivative of the contact parameters given the velocity of the rigid body, so in the following sections $\ddot{\mathbf{x}}$ and $\ddot{\boldsymbol{\theta}}$ are derived as functions of $\mathbf{s}$, $\dot{\mathbf{u}}_{b_i}$ and $\dot{\mathbf{u}}_{g_i}$.

3.3.1 Useful Differential Relations

In the derivation of the dynamic equations it is useful to decompose $\dot{\mathbf{R}}(\boldsymbol{\theta})\mathbf{x}_b(\mathbf{u}_{b_i})$ into a matrix times $\dot{\boldsymbol{\theta}}$. The matrix $\mathbf{P}(\boldsymbol{\theta}, \mathbf{r})$ and the vector $\mathbf{p}_l(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \mathbf{r}, \dot{\mathbf{r}})$, where $\mathbf{r}$ is a vector in bolt body coordinates, are defined for this purpose.

$$\dot{\mathbf{R}}(\boldsymbol{\theta})\mathbf{r} = \mathbf{P}(\boldsymbol{\theta},\mathbf{r})\dot{\boldsymbol{\theta}} \tag{9}$$

$$\frac{d}{dt}(\dot{\mathbf{R}}(\boldsymbol{\theta})\mathbf{r}) = \mathbf{P}(\boldsymbol{\theta},\mathbf{r})\ddot{\boldsymbol{\theta}} + \mathbf{p}_l(\boldsymbol{\theta},\dot{\boldsymbol{\theta}},\mathbf{r},\dot{\mathbf{r}}) \tag{10}$$

The relation between $\dot{\boldsymbol{\theta}}$ and $\boldsymbol{\omega}_o$, the angular velocity of the bolt in the inertial frame, is given by:

$$\boldsymbol{\omega}_o = \mathbf{S}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}. \tag{11}$$

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

3.3.2 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 written in the body frame, the second equates the sum of forces in the inertial frame to the derivative of the translational momentum. These two equations are decoupled about the center of mass of the moving object [Goldstein 80].

$$\mathbf{I}_n\dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times \mathbf{I}_n\boldsymbol{\omega}_b = \sum \boldsymbol{\tau}_b \tag{12}$$

$$m\ddot{\mathbf{x}}_{cm} = \sum \mathbf{f}_{cm} \tag{13}$$

Here $\mathbf{I}_n$ is the inertia tensor about the center of mass in the body frame and $\mathbf{x}_{cm}$ is the location of the center of mass of the bolt in the inertial frame. To put these equations in terms of the state variable $\mathbf{c}$ we make use of equation (10) and the shorthand:

$$\mathbf{P}_{cm} = \mathbf{P}(\boldsymbol{\theta},\mathbf{x}_{cm}) \quad , \quad \mathbf{p}_{l_{cm}} = \mathbf{p}_l(\boldsymbol{\theta},\dot{\boldsymbol{\theta}},\mathbf{x}_{cm},\mathbf{0})$$

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:

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

As the 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:

$$\begin{aligned}
\sum \boldsymbol{\tau}_b = {} & \mathbf{R}^T\boldsymbol{\tau}_a + (\mathbf{r}_a - \mathbf{r}_{cm}) \times (\mathbf{R}^T\mathbf{f}_a) + \\
& \sum_i [\mathbf{x}_b(\mathbf{u}_{b_i}) - \mathbf{r}_{cm}] \times (\mathbf{R}^T\mathbf{f}_i)
\end{aligned}$$

where $\boldsymbol{\tau}_a$ is the controller torque in the inertial frame. To simplify future expressions, $\mathbf{n}_l(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})$ will be defined as:

$$\mathbf{n}_l(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) = (\mathbf{R}^T\mathbf{S}\dot{\boldsymbol{\theta}}) \times (\mathbf{I}_n\mathbf{R}^T\mathbf{S}\dot{\boldsymbol{\theta}}) + (\mathbf{I}_n\dot{\mathbf{R}}^T\mathbf{S} + \mathbf{I}_n\mathbf{R}^T\dot{\mathbf{S}})\dot{\boldsymbol{\theta}}. \tag{14}$$

After substituting into Euler's equations we have:

$$\begin{aligned}
\mathbf{I}_n\mathbf{R}^T\mathbf{S}\ddot{\boldsymbol{\theta}} = {} & \mathbf{R}^T\boldsymbol{\tau}_a + (\mathbf{r}_a - \mathbf{r}_{cm}) \times (\mathbf{R}^T\mathbf{f}_a) - \mathbf{n}_l + \\
& \sum_i [\mathbf{x}_b(\mathbf{u}_{b_i}) - \mathbf{r}_{cm}] \times (\mathbf{R}^T\mathbf{f}_i)
\end{aligned} \tag{15}$$

$$m\ddot{\mathbf{x}} = \mathbf{f}_a - m(\mathbf{P}_{cm}\ddot{\boldsymbol{\theta}} + \mathbf{p}_{l_{cm}}) + \sum_i \mathbf{f}_i. \tag{16}$$

3.3.3 Solving for the Accelerations

As was stated earlier, we consider the case of all contacts sliding. We show a method for deriving the equations of motion for n contacts with n velocity constraints. The resulting $6+n$ equations are linear in the 6 accelerations and n force magnitudes.

The velocity and acceleration constraints are given by:

$$\mathbf{v}_{c_i} \cdot \hat{\mathbf{n}}_{c_i} = 0 \quad , \quad \dot{\mathbf{v}}_{c_i} \cdot \hat{\mathbf{n}}_{c_i} + \mathbf{v}_{c_i} \cdot \dot{\hat{\mathbf{n}}}_{c_i} = 0. \tag{17}$$

Using the shorthand:

$$\mathbf{P}_{xb_i} = \mathbf{P}(\boldsymbol{\theta},\mathbf{x}_b(\mathbf{u}_{b_i})) \quad , \quad \mathbf{p}_{l_{xb_i}} = \mathbf{p}_l(\boldsymbol{\theta},\dot{\boldsymbol{\theta}},\mathbf{x}_b(\mathbf{u}_{b_i}),\tfrac{d}{dt}\mathbf{x}_b(\mathbf{u}_{b_i}))$$

combine equations (3) and (10) to get:

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

Now equations (15) and (16) and the constraint equations (17) may be combined in a $6+n$ square matrix equation which can be solved for the accelerations and force magnitudes. For example, for 2 points of contact we solve the 8×8 matrix equation:

$$\begin{bmatrix}
\mathbf{I}_n\mathbf{R}^T\mathbf{S} & \mathbf{0} & \begin{matrix}\mathbf{R}^T\hat{\mathbf{f}}_1\times \\ [\mathbf{x}_b(\mathbf{u}_{b_1}) - \mathbf{r}_{cm}]\end{matrix} & \begin{matrix}\mathbf{R}^T\hat{\mathbf{f}}_2\times \\ [\mathbf{x}_b(\mathbf{u}_{b_2}) - \mathbf{r}_{cm}]\end{matrix} \\
m\mathbf{I}_{3\times 3} & m\mathbf{P}_{cm} & -\hat{\mathbf{f}}_1 & -\hat{\mathbf{f}}_2 \\
(\mathbf{P}_{xb_1}^T\hat{\mathbf{n}}_{c_1})^T & \hat{\mathbf{n}}_{c_1} & 0 & 0 \\
(\mathbf{P}_{xb_2}^T\hat{\mathbf{n}}_{c_2})^T & \hat{\mathbf{n}}_{c_2} & 0 & 0
\end{bmatrix} \cdot$$

$$\begin{bmatrix} \ddot{\boldsymbol{\theta}} \\ \ddot{\mathbf{x}} \\ f_{n_1} \\ f_{n_2} \end{bmatrix} = \begin{bmatrix} \mathbf{R}^T\boldsymbol{\tau}_a + (\mathbf{r}_a - \mathbf{r}_{cm}) \times (\mathbf{R}^T\mathbf{f}_a) - \mathbf{n}_l \\ \mathbf{f}_a - m\mathbf{p}_{l_{cm}} \\ -\mathbf{p}_{l_{xb_1}} \cdot \hat{\mathbf{n}}_{c_1} - \mathbf{v}_{c_1} \cdot \dot{\hat{\mathbf{n}}}_{c_1} \\ -\mathbf{p}_{l_{xb_2}} \cdot \hat{\mathbf{n}}_{c_2} - \mathbf{v}_{c_2} \cdot \dot{\hat{\mathbf{n}}}_{c_2} \end{bmatrix}$$

3.4 Impulses

This phenomena of impact is modeled by applying impulses at the edge of the friction cone at the onset of contact. A fraction of the impulse, given by the coefficient of restitution [Lankarani 90], is added to the impulse that cancels the velocities so that the collision is modeled as being partially elastic.

Each contact i applies an impulse of magnitude λ_i at the contact point $\mathbf{x}_b(\mathbf{u}_{b_i})$ in the direction $\hat{\mathbf{f}}_i$. (In computing $\hat{\mathbf{f}}_i$ from equation (7) we take the projection of $\mathbf{v}_{c_i}$ onto the tangent plane at the contact.) We wish to determine the magnitudes 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 $\dot{\bar{\boldsymbol{\theta}}}$ 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:

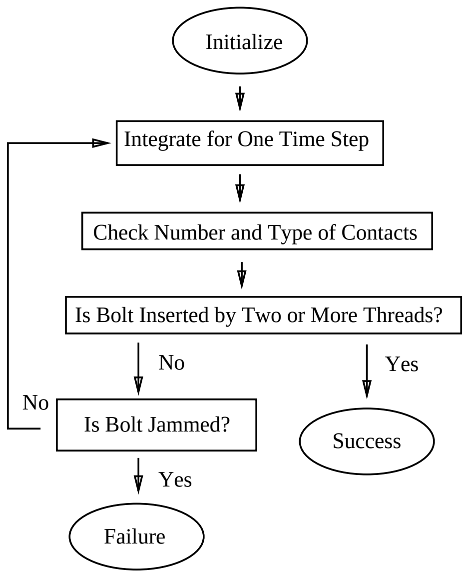

Figure 5: Simulation program structure

$$m(\dot{\tilde{\mathbf{x}}}_{cm} - \dot{\mathbf{x}}_{cm}) = \sum_i \lambda_i \hat{\mathbf{f}}_i \tag{18}$$

$$\mathbf{I}_n \mathbf{R}^T \mathbf{S}(\dot{\tilde{\boldsymbol{\theta}}} - \dot{\boldsymbol{\theta}}) = \sum_i [\mathbf{x}_b(\mathbf{u}_{b_i}) - \mathbf{r}_{cm}] \times \lambda_i \mathbf{R}^T \hat{\mathbf{f}}_i \tag{19}$$

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

$$\dot{\tilde{\mathbf{x}}} = \dot{\mathbf{x}} + \mathbf{P}_{cm}(\dot{\boldsymbol{\theta}} - \dot{\tilde{\boldsymbol{\theta}}}) + \frac{1}{m}\sum_i \lambda_i \hat{\mathbf{f}}_i$$

and the velocity constraints:

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

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

$$\left[\dot{\mathbf{x}} + \mathbf{P}_{cm}\dot{\boldsymbol{\theta}} + (\mathbf{P}_{xb_j} - \mathbf{P}_{cm})\dot{\tilde{\boldsymbol{\theta}}} + \frac{1}{m}\sum_i \lambda_i \hat{\mathbf{f}}_i\right] \cdot \hat{\mathbf{n}}_{c_j} = 0. \tag{20}$$

Equations (19) and (20) give $3+n$ equations linear in the 3 rotational velocities and the n impulse magnitudes. The λ_i are then multiplied by $(1.0+\gamma)$ where γ is the coefficient of restitution. The resulting impulses are applied to the bolt.

4 Simulation of Motion

The results developed in the previous two sections were used to create a program producing an animated simulation of a bolt threading into a nut. The program loops through two procedures: a state update routine and a contact configuration checking routine. This is shown in Figure 5.

4.1 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 simulation has probably proceeded through the intersection of the nut and bolt. Thus the trajectory is followed backwards to find the point of time during the last update at which the configuration changed. This method is similar to [Baraff 90]. Figure 6 gives a flow outline of this routine.

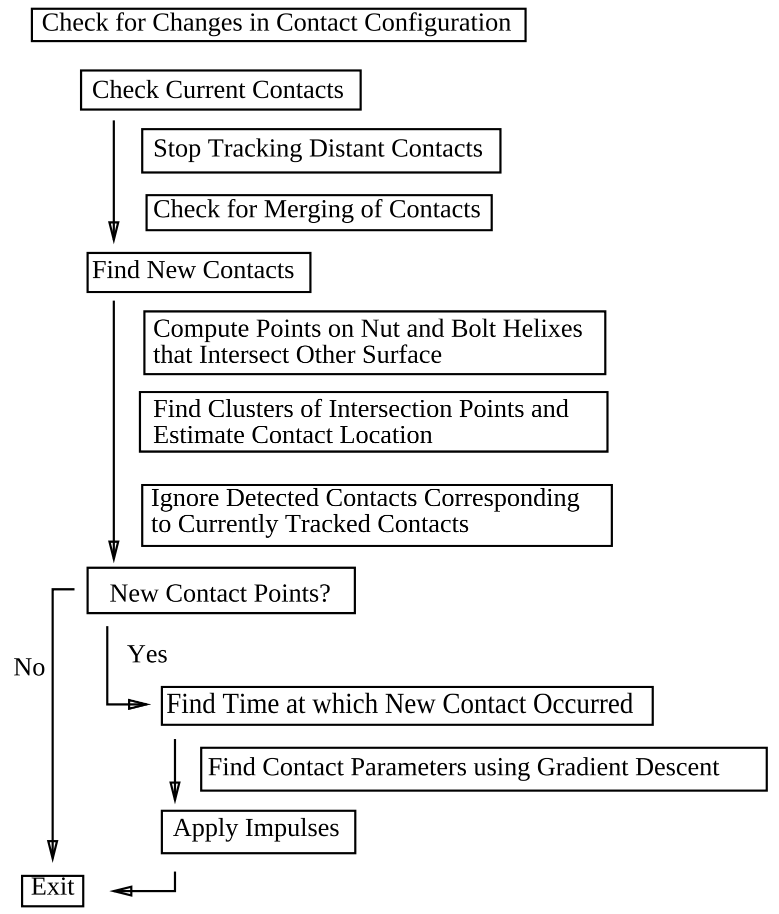

Figure 6: Configuration Analysis Procedure

4.2 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 square and rectangular matrix equations. The Linpack Fortran libraries are used for this.

4.3 Simulations

We simulated the motion of a 10 gram bolt with a 1.0mm thread pitch, an allowance ratio of 0.05, and a radius of 3.0mm under underdamped grasp stiffness control. Referring to equations (5) and (6), we chose:

$$\begin{aligned}
\dot{\mathbf{c}}_e &= \begin{bmatrix} 0 & 0 & -1.0\frac{\text{mm}}{\text{s}} & 0 & 0 & -360\frac{\text{degrees}}{\text{s}} \end{bmatrix}^T \\
\mathbf{c}_e(0) &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1^\circ & 0 \end{bmatrix}^T \\
\mathbf{K} &= \text{diag}\begin{bmatrix} \mathbf{K}_x & \mathbf{K}_\theta \end{bmatrix} \\
\mathbf{K}_x &= \begin{bmatrix} 0.1 & 0.1 & 0.01 \end{bmatrix}\frac{\text{N}}{\text{mm}} \\
\mathbf{K}_\theta &= \begin{bmatrix} 0.35 & 0.35 & 0.35 \end{bmatrix}\frac{\text{N mm}}{\text{degree}}.
\end{aligned}$$

$\mathbf{B}$ was chosen based on the inertia to give a damping ratio of 0.1 for each degree of freedom near the configuration $\mathbf{c} = \mathbf{0}$. The control force is applied at $\mathbf{r}_a$, which was chosen to be 3.0mm beneath the base of the bolt along its axis.

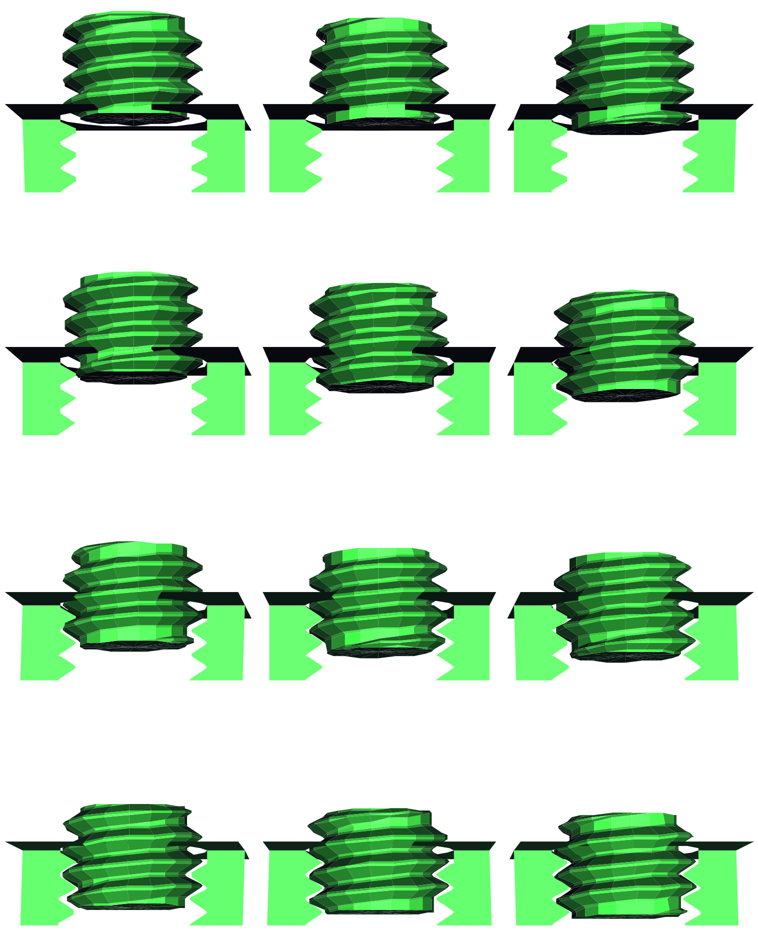

Figure 7: Snapshots from the simulation.

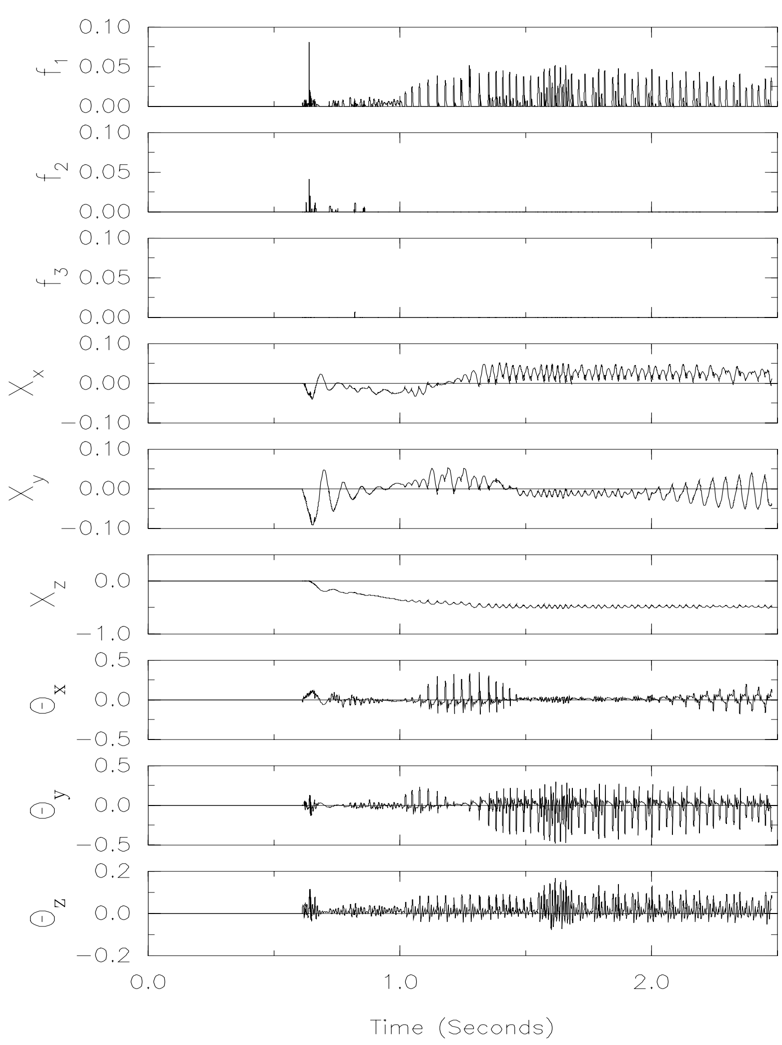

Figure 8: Contact force magnitudes in Newtons and displacement from equilibrium position, $\mathbf{c}(t) - \mathbf{c}_e(t)$, in mm and degrees

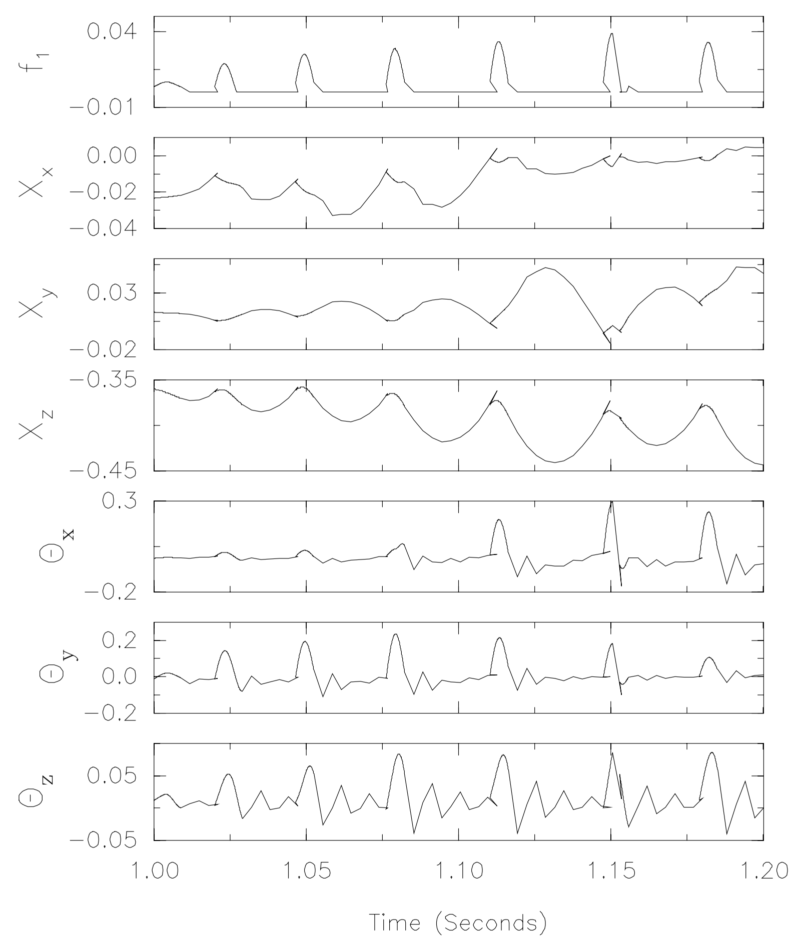

Figure 9: Magnified portion of the previous figure showing the underdamped response and backtracking.

Figure 7 shows snapshots from the simulation. Contact forces and displacements from the equilibrium position are shown in Figure 8. For $\mathbf{c}_e$ as given above, the bolt and nut make their first contact at the crests of their first thread. This is a singular configuration in that the bolt can be displaced vertically in either direction by a half pitch in order to mate correctly. For the stiffness chosen the bolt is displaced by -0.5mm, as shown by $\mathbf{x}_z$ between times $t = .6$ and $t = 1.2$ in Figure 8, resulting in a net upward force applied along the z-axis by the controller. As $\mathbf{K}_\theta$ is large compared to $\mathbf{K}_x$, there is little change in the orientation. This is also due to the fact that the initial orientation error was small, only 1 degree, and the allowance ratio was large, 5%, so the parts could be mated with a final error of 1 degree. This "slop angle" is discussed further in [Nicolson 90].

We should also explain the chattering behavior shown in Figure 8. This is due to the coefficient of restitution, chosen to be 0.5, and the damping ratio, chosen to be 0.1. As half of the force of contact goes into accelerating the bolt off the surface and the motion is underdamped, the bolt bounces along during insertion, similar to a ball bearing bouncing on an anvil. The ball bearing would bounce more, in fact, as a typical coefficient of restitution would be 0.8 [Lankarani 90] and there would be no damping. The undamped behavior of $\mathbf{x}$ and $\boldsymbol{\theta}$ is clear in Figure 9 which shows a time-magnified portion of the trajectory. In particular $\mathbf{x}_z$ shows how the bolt impacts the nut during vertical upward motion.

The effect of the backtracking discuused in section 4.1 may also be noticed in Figure 9. For example, note that at

$t = 1.11$ the trajectory branches. The trajectory segments that terminate indicate the motion that would have occurred had a contact not occurred. As the contact was detected by the intersection algorithm after the actual contact occurred, the simulation backtracked along the trajectory until the constraint equations were best satisfied.

5 Extensions and Conclusions

In this paper a method for simulating the motion of a bolt threading into a nut has been presented. The method is based on a generalized cylindrical description of a screw thread, impulsive forces, and computation of contact forces from dynamic motion equations. This method was implemented and sample simulation results are shown. We will plan to use the simulation to analyze control strategies and investigate the effect of thread design parameters on assembly. The method could be adapted for mating problems with other generalized cylindrical parts.

6 Acknowledgements

We thank Matthew Berkemeier for his help on the integration routine and Matthew Mason for providing the reference to [Blaer 62].

References

[Allen 88] L.G. Allen, "Fastener Failure: Who Dunnit? What's to be done about it?," *Automation,* November 1988.

[Baraff 90] D. Baraff, "Curved Surfaces and Coherence for Non-penetrating Rigid Body Simulation," *Computer Graphics*, Vol. 24, No. 4, August 1990.

[Bickford 81] J.H. Bickford, *An Introduction to the Design and Behavior of Bolted Joints,* Marcel Dekker, Inc., New York, 1981.

[Blaer 62] I.L. Blaer, "Reliable Starting of Threads," *Russian Engineering Journal,* v.42, n. 12, 1962.

[Blake 86] A. Blake, *Threaded Fasteners,* Marcel Dekker Inc., New York, 1986.

[Cole 89] A. Cole, "Dextrous Manipulation for Multifingered Hands: Planning and Control," UC Berkeley Ph.D. Thesis, July 1989.

[Donald 89:2] B.R. Donald and D.K. Pai, "On the Motion of Compliantly Connected Rigid Bodies in Contact, Part II: A System for Analyzing Designs for Assembly," *Proc. 1990 IEEE Conf. on Robotics and Automation,* May 1990.

[Goldstein 80] H. Goldstein, *Classical Mechanics*, 2nd. ed., Addison-Wesley Publishing Company Inc., Philippines, 1980.

[Hoffman 89] C.M. Hoffman, *Geometric and Solid Modeling: An Introduction,* Morgan Kaufmann Publishers, Inc., 1989, San Mateo, CA.

[Lankarani 90] H.M. Lankarani and P.E. Nikravesh, "A Contact Force Model with Hysteresis Damping for Impact Analysis of Multibody Systems," *Journal of Mechanical Design,* Vol 112, September 1990.

[Loncaric 87] J. Loncaric, "Normal Forms of Stiffness and Compliance Matrices," *IEEE Journal of Robotics and Automation,* Vol. RA-3, No. 6, December 1987.

[Lozano-Perez 84] T. Lozano-Perez, M.T. Mason and R.H. Taylor, "Automatic Synthesis of Fine-Motion Strategies for Robots," *The Int. Journal of Robotics Research,* Vol. 3, No. 1, Spring 1984.

[Montana 88] D.J. Montana, "The Kinematics of Contact and Grasp," *The Int. Journal of Robotics Research,* Vol. 7, No. 3, June 1988.

[Nevins 80] J.L. Nevins and D.E. Whitney, "Assembly Research," *Factory Automation* v. 2, 1980, Maidenhead, England.

[Nicolson 90] E.J. Nicolson, "Grasp Stiffness Solutions for Threaded Insertion," U.C. Berkeley M.S. Thesis, December 1990.

[Patrikalakis 90] N.M. Patrikalakis and P.V. Prakash, "Surface Intersections for Geometric Modeling," *Trans. of the ASME,* Vol. 112, March 1990.

[Ponce 87] J. Ponce and D. Chelberg, "Localized Intersections Computation for Solid Modelling with Straight Homogeneous Generalized Cylinders," *Proc. 1987 IEEE Conf. on Robotics and Automation,* Raleigh, NC, April 1987.

[Schimmels 90] J.M. Schimmels and M.A. Peshkin, "Synthesis and Validation of Non-Diagonal Accommodation Matrices for Error-Corrective Assembly," *Proc. 1990 IEEE Conf. on Robotics and Automation,* Cincinatti, OH, May 1990.

[Smith 80] S.K. Smith, "Use of a Microprocessor in the control and Monitoring of Air Tools while Tightening Threaded Fasteners," Eaton Corporation, *Autofact West,* Proc. Vol. 2, Society of Manufacturing Engineers, Dearborn, MI, 1980.

[Strip 88] D.R. Strip, "Insertions Using Geometric Analysis and Hybrid Force-Position Control: Method and Analysis," *Proc. 1988 IEEE Int. Conf. on Robotics and Automation,* Philidelphia, PA April 1988.

[Tao 90] J.M. Tao, J.Y.S. Luh and Y.F. Zheng, "Compliant Coordination Control of Two Moving Industrial Robots," *IEEE Trans. on Robotics and Automation* Vol. 6, No. 3, June 1990.

[Whitney 82] D.E. Whitney, "Quasi-Static Assembly of Compliantly Supported Rigid Parts," *Journal of Dynamic Systems, Measurement, and Control,* March 1982, Vol. 104.