SIMULATION OF CONTACT TEXTURES AND FORCES
WITH APPLICATION TO HAPTIC INTERFACES

By
Juhani Siira
B.A.Sc, University of British Columbia, 1989
M.B.A, McGill University, 1992

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in
THE FACULTY OF GRADUATE STUDIES
ELECTRICAL ENGINEERING

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
December, 1995
© Juhani Siira, 1995
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia
Vancouver, Canada

Date January 02, 1996
Abstract

This thesis will consider the simulation and display of contact forces as currently present in much of the robotics literature. In particular, the following two issues are addressed.

Texture forces are not currently routinely included in haptic display algorithms, nor are there comprehensive frameworks for using physical parameters to generate realistic texture forces. Algorithms are required which combine implementation ease with minimal computation requirements. This thesis will present one such algorithm, which has been shown to produce realistic feeling texture forces by generating force pulses at high bandwidth. The algorithm has the potential to use physical measures of surface roughness to produce accurate texture forces.

Much of the contact simulation currently presented uses the spring/damper as a surface model. Although the spring/damper can be an accurate model for impact, it is not generally nearly as applicable to low-speed or quasi-static load cycling, and it requires careful consideration when applied to multi-point contact situations (when naive application can cause incorrect results). The shortcomings of the spring/damper system are exposed, and an alternative model is presented which corrects the deficiencies of the spring/damper, without greatly increasing implementation complexity. Some of the features of the new model include energy dissipation during quasi-static cycling, correct multi-point contact behaviour without modification, and a clearly established equilibrium transition time, which can be beneficial in maintaining subsets of active objects in a complicated simulation.
# Table of Contents

Abstract ii

List of Figures vii

Acknowledgement ix

Dedication x

1 Introduction 1
   1.1 Rigid Body Contact ........................................ 2
   1.2 Haptic Texture During Contact ............................. 3
   1.3 Distinctions Between Simulation and Display ............... 4
   1.4 Thesis Organization ...................................... 5

2 The "P2 Pantograph" Haptic Device 6
   2.1 Pantograph Description ................................... 6
   2.2 Pantograph Kinematics .................................... 6
      2.2.1 Forward Kinematics .................................. 7
      2.2.2 Distal Link Velocities from Base Link Velocities .... 10
      2.2.3 Distal Link Accelerations from Base Link Accelerations 10
   2.3 Jacobians ............................................. 11
      2.3.1 Handle to Base Joints for Individual Chains ........... 12
      2.3.2 Handle to Base Joints for the Closed Chain ............ 13
   2.4 Pantograph Dynamics ..................................... 14
3 A Stochastic Approach to Haptic Texturing

3.1 Introduction ............................................. 15
3.2 Related Work ........................................... 16
  3.2.1 Textures in Graphics ............................... 16
  3.2.2 Textures in Haptics ............................... 17
3.3 Haptic Textures ........................................ 17
  3.3.1 Surface Roughness Model ....................... 17
  3.3.2 Haptic Texture Definition ....................... 20
3.4 Force Decomposition .................................. 21
3.5 Generating Gaussian Texture ....................... 23
  3.5.1 Generating Texture Forces in World-Space ........ 23
  3.5.2 Generating World-Space Forces in Joint Space .... 24
  3.5.3 Input Parameters to the Gaussian Texture Functions .... 26
3.6 Implementation ....................................... 27
  3.6.1 Testbed ........................................... 27
  3.6.2 Results ........................................... 27

4 Internal Friction Contact Model ..................... 30

4.1 Introduction ......................................... 30
4.2 Dynamic Simulation - Euler-Lagrange Formulation .... 31
  4.2.1 Constrained Dynamics .......................... 31
  4.2.2 Planar Implementation of Contact ............. 32
4.3 Coefficient of Restitution ........................... 35
  4.3.1 Newton’s Hypothesis ............................ 35
  4.3.2 Poisson’s Hypothesis ............................ 36
  4.3.3 Energy Hypothesis .............................. 36
4.3.4 Computation for Objects of Different Material Properties 37
4.4 Existing Methods for Contact Simulation 37
   4.4.1 Finite Element Modelling 37
   4.4.2 Soft Contact 38
   4.4.3 Hard Contact 39
4.5 Surface Contact 41
4.6 Internal Friction 43
   4.6.1 Simplified Surface Contact Model 45
   4.6.2 Choosing the form of $F(g)$ 48
4.7 Determining the Effective Stiffness $k$: Measuring Impact Duration 49
4.8 Special Case Example: Bar Landing on a Planar Surface 51
   4.8.1 Internal Friction Algorithm 52
   4.8.2 Dropping the Bar in a Vertical Orientation 52
   4.8.3 Dropping the Bar in a Horizontal Orientation 55
   4.8.4 Numerical Stability 61
   4.8.5 Identifying Equilibrium Transitions 61
4.9 Using the Internal Friction Model For Haptic Display 62

5 Conclusions 63

6 Future Work 65

Bibliography 67

A Pantograph Dynamics 72
   A.1 Determining Mass Center for Links 72
   A.2 Link Inertia 72
      A.2.1 Energy Losses 74
# List of Figures

2.1 Pantograph Haptic Interface ........................................... 7  
2.2 Implementation Apparatus ............................................. 8  
2.3 Definition of Parameters for Kinematics ............................. 9  

3.4 A "wavy" Surface Profile .............................................. 18  
3.5 A "wavy" Surface Profile with Guassian Deviation Added ............ 19  
3.6 Gaussian Surface Profiles ........................................... 20  
3.7 Force Decomposition ................................................... 22  
3.8 Lateral Texture Force Experiment .................................... 28  

4.9 Planar Objects Contacting Planar Obstacles ......................... 33  
4.10 Rate-Dependent Hysteretic Loop .................................... 43  
4.11 Rate-Independent Hysteretic Loop .................................. 45  
4.12 Quasi-static Loading for Internal Friction and Spring/Damper Based Models 50  
4.13 Contact Duration Experiment ....................................... 50  
4.14 Equilibrium Transition for Bar on Planar Surface ................... 53  
4.15 Internal Friction Algorithm for Planar Bar .......................... 53  
4.16 Straight Vertical Bar Landing on a Flat Surface. The curves for the Spring/Damper and Internal Friction models overlap. (IF = Internal Friction model, SD = Spring/Damper model). Note, all units SI (mks) . . . 54  
4.17 Straight Vertical Bar Landing on a Flat Surface - Single Bounce (IF = Internal Friction model, SD = Spring/Damper model). Note, all units SI (mks) ......................................................... 56
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.18</td>
<td>Straight Horizontal Bar Landing on a Flat Surface - Single Bounce (IF = Internal Friction Model, SD = Spring/Damper Model). Note, all units SI (mks).</td>
<td>58</td>
</tr>
<tr>
<td>4.19</td>
<td>Straight Horizontal Bar Landing on a Flat Surface (IF = Internal Friction Model, SD = Spring/Damper Model). Note, all units SI (mks).</td>
<td>59</td>
</tr>
<tr>
<td>4.20</td>
<td>Arbitrary Geometry with Spring/Damper Contact Forces</td>
<td>60</td>
</tr>
<tr>
<td>A.21</td>
<td>Measuring Link Mass Center</td>
<td>73</td>
</tr>
<tr>
<td>A.22</td>
<td>Pendulum Inertia versus Oscillation Period</td>
<td>77</td>
</tr>
<tr>
<td>D.23</td>
<td>Internal Friction Model with $C^0$ Continuity</td>
<td>87</td>
</tr>
<tr>
<td>E.24</td>
<td>Equilibrium Lists for Objects</td>
<td>90</td>
</tr>
<tr>
<td>E.25</td>
<td>System Scenario</td>
<td>91</td>
</tr>
</tbody>
</table>
Acknowledgement

The author wishes to thank the thesis committee, Peter Lawrence, Tim Salcudean and Dinesh Pai, for constructive suggestions to improve the final draft. The author is grateful to Tim Salcudean for providing the opportunity to straddle the traditional, if not often artificial, line between computer science and electrical engineering. Also, the author is particularly grateful to Dinesh Pai for providing an environment allowing the opportunity to pursue interesting research directions, and for providing valuable suggestions throughout the research and writing of this thesis. Finally, the author thanks Rod Barman and Stewart Kingdon for providing the control hardware and much of the software for the haptic interface used during the research.
To my family,

for their support and encouragement.
Chapter 1

Introduction

This thesis presents algorithms for the simulation and display of dynamic systems, in particular rigid objects in contact. The simulation treats objects contacting under impact and low-frequency load cycling, while the display will focus on haptic devices, which have received less attention than graphical displays. Issues of accurate simulation and display of rigid bodies in contact are important in developing realistic virtual environment computer systems. Virtual environments can be useful in areas such as testing and prototyping of robotic systems, telerobotics with substantial communication delay, and worker training for hazardous or complex tasks. For example, in telerobotics, a remote manipulator must often come in contact with various environments. In order to design and verify hardware and control algorithms for such tasks, it is useful to use simulations in place of actual prototyping, for reasons of safety and efficiency. Furthermore, if there are communication buffers or switches (or extreme physical separation) between the manipulator and the user (as is typically the case of a space-bound manipulator controlled by an earth-bound user), then substantial communication delays can occur between control requests from the user, manipulator actuation and feedback to the user. In such an application, a simulation of the manipulated world could provide the user with immediate feedback (or consequences) of a given action. If the manipulation task involves hazardous materials or environments, then allowing the user to practice and plan using a simulation is of clear benefit. The simulation should, therefore, provide an accurate depiction of the actual forces applied to the manipulator, and those forces should be conveyed to the user.
Chapter 1. Introduction

In a meaningful way.

In this thesis, the simulation and display of contact forces will be treated as separate, but related, pursuits. Due to the physical limitations of haptic devices, algorithms which produce accurate simulation results may not produce accurate displays. In particular, algorithms which produce accurate simulation trajectories for constraint forces of stiff objects in contact, may not produce accurate trajectories when implemented on a haptic device. Conversely, algorithms which produce realistic texture forces in haptic devices may require substantially different parameters to produce accurate simulated texture forces. The recognition that a simulation trajectory may be substantially different from the haptic display trajectory (which provides the input to the simulation) raises many interesting questions. The true system state is unclear when the input and simulation trajectories diverge. These issues are beyond the scope of the present thesis, but the reader should be aware that both accurate simulation and realistic display are necessary - how the two are optimally merged is an open question.

1.1 Rigid Body Contact

We consider the general nature of contact and, although a thorough treatment is beyond the scope of this work, we isolate some of the important aspects of impact and non-impact contact. From this, we develop a model of contact which is based on internal friction, which is an energy absorption phenomenon of solid materials. In choosing the form and parameters, we seek to create a simple model which can provide accurate macroscopic behaviour, as measured by energy dissipation during impacts, as well as microscopic behaviour, as measured by impact duration. Furthermore, the model provides the correct form for the force/displacement curve during non-impact load cycling, and is simple to implement. To differentiate between impact and non-impact contact, the most useful
definition is to consider the relative size of the contact force with respect to the net force applied to the object. If the contact force is largely dominant, then the contact can be considered impact (this is characteristic of "fast" relative closing speed of the contact points of the bodies).

The internal friction model is compared to a spring/damper, which is one of the most prevalent existing models of rigid body contact. Most often, objects are modelled as polyhedra, and the spring/damper is applied to the intrusion distance from a polygon vertex of one object to the surface of the other object. Although the spring/damper is shown to provide good results in impact situations, additional complexity is necessary to deal effectively with non-impact, multiple-point contact. We expose the major deficiencies of the spring/damper by noting that it is energy-conservative in quasi-static loading, and that the choice of damping is dependent on the number of simultaneous contact points. Choosing the correct damping for multiple contact points under arbitrary geometries is difficult. We show that the internal friction model overcomes both of these deficiencies without greatly complicating the implementation.

1.2 Haptic Texture During Contact

Haptic interfaces, such as force reflecting hand masters, are mechanical devices that can provide kinesthetic feedback to human operators. The impression of haptic roughness can be effectively produced by applying force pulses to the hand, which can enrich user interactions in simulation or teleoperation. Texture has been studied extensively for graphical displays, and some deterministic models have been studied for haptic displays. However, texture is an important component of contact force, and should be routinely included in haptic displays. All objects have a surface roughness which manifests itself as small forces when objects slide under load against each other. Simulating this roughness haptically
enriches the interaction between a user and a virtual world, just as creating graphical textures enhances the depiction of a scene. As with graphical textures, a major design constraint for haptic textures is the generation of a sufficiently "realistic" texture given hard constraints on computational costs. This thesis presents a simple, fast algorithm to synthesize haptic textures from statistical properties of surfaces. The synthesized texture can be overlaid on other contact models, such as hard contact with Coulomb friction. The algorithm requires minimal hardware support, and can be implemented on a variety of force-feedback mechanisms. It produces high bandwidth texture forces with model parameters potentially definable from material and physical surface characteristics. To enable high bandwidth force generation, methods are presented to allow the texture forces to be implemented in low-level, embedded control layers. The algorithm was successfully implemented on a two-degree-of-freedom, pantograph-style haptic device.

1.3 Distinctions Between Simulation and Display

In this thesis, the simulation and display of contact forces will be treated as separate, but related, pursuits. Due to the physical limitations of haptic devices, algorithms which produce accurate simulation results may not transfer to produce accurate displays. In particular, algorithms which produce accurate simulation trajectories for constraint forces of stiff objects in contact, may not produce accurate trajectories when implemented on a haptic device. Also, algorithms which produce realistic simulated texture forces may require substantially different parameters to produce realistic texture forces in haptic devices. The recognition that a simulation trajectory may be substantially different from the haptic display trajectory (which acts to input states to the simulation) raises many interesting questions. The true system state is unclear when the input and simulation trajectories diverge. These issues are beyond the scope of the present thesis, but the
reader should be aware that both accurate simulation and realistic display are necessary - how the two are optimally merged is an open question.

1.4 Thesis Organization

Chapter 2 describes the pantograph haptic device used, and derives the closed-loop kinematics and necessary dynamics. It is shown that the inertial properties of the pantograph are such that its dynamics can be safely ignored for the applications described in this thesis.

Chapter 3 describes a new algorithm to generate haptic texture based on stochastic modelling using physical parameters, as opposed to the deterministic models which have previously been reported.

Chapter 4 surveys contact constraint force simulation in general, and based on this analysis, proposes a new model of contact constraint force. The new model is compared to a spring/damper model using a straight bar landing on a flat surface. This illustrative example will expose some of the weaknesses of the spring/damper model, as well as show that the new internal friction model overcomes the problems associated with the spring/damper.

Chapter 5 provides concluding remarks, and Chapter 6 discusses future research directions.
Chapter 2

The "P2 Pantograph" Haptic Device

2.1 Pantograph Description

The device is comprised of a five bar linkage with rotational actuators on each of the two base links, and a workspace of approximately 16cm wide x 10cm long, as shown in Figure 2.1. The P2 linkage was designed by Vincent Hayward [RH94], but uses different control architecture. Optical encoders provided position feedback for the two ground-fixed links with 500 pulses/revolution, counted in quadrature to give 2000 counts/revolution resolution. This gave worst-case handle position resolution of about 1 mm. Actuation was achieved with pulse-width-modulated (PWM) amplifiers through 12 bit D/A conversion. A block diagram of the apparatus is shown in Figure 2.2. The MC68332 and T805 are located on an Embedded Module for Distributed Control (EMDC)\(^1\), which interfaces to a workstation, allowing control code to be downloaded.

2.2 Pantograph Kinematics

Since the encoders are mounted on the base links, kinematic relations are needed between the base links and the handle position, velocity and acceleration. This section will follow a procedure proposed by Wang [Wan91]. The notation (as shown in Figure 2.3) in this section is consistent with Wang; it will not be adopted outside of this section.

\(^1\)Designed in the Laboratory for Computational Intelligence, UBC Department of Computer Science, by Rod Barman, Stewart Kingdom, Alan Mackworth and Dinesh Pai
2.2.1 Forward Kinematics

The forward kinematics are solved using a root finding procedure. Given the base link angles \( q_1 \) and \( q_5 \), the distal link angles \( q_2 \) and \( q_4 \) are found using a gradient descent approach. The five bar linkage is treated as being comprised of two-link, ground-fixed chains connected at the handle, as shown in Figure 2.3. The gradient descent is continued until the ends of the two chains are within some tolerance. Let the location of each chain end be defined by:

\[
x_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = F_1(q_1) = \begin{bmatrix} d_1 \cos q_1 + d_2 \cos q_{12} \\ d_1 \sin q_1 + d_2 \sin q_{12} \end{bmatrix} \tag{2.1}
\]

\[
x_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = F_2(q_2) = \begin{bmatrix} d_5 + d_4 \cos q_5 + d_3 \cos q_{45} \\ d_4 \sin q_5 + d_3 \sin q_{45} \end{bmatrix} \tag{2.2}
\]

where \( q_{ij} = q_i + q_j \), \( q_i \) is the set of all possible joint angles for chain \( i \), and

\[
q_1 = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad q_2 = \begin{bmatrix} q_5 \\ q_4 \end{bmatrix} \tag{2.3}
\]
Figure 2.2: Implementation Apparatus
The loop-closure equation is

\[ x_1 = x_2 \]  \hspace{1cm} (2.4)

Let

\[ q_c = \begin{bmatrix} q_2 \\ q_4 \end{bmatrix} \]  \hspace{1cm} (2.5)

The gradient descent equations are

\[
\begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix} = x_1 - x_2
\]  \hspace{1cm} (2.6)

\[
J_c = \begin{bmatrix}
\frac{\partial (\delta x_1)}{\partial q_2} & \frac{\partial (\delta x_1)}{\partial q_4} \\
\frac{\partial (\delta x_2)}{\partial q_2} & \frac{\partial (\delta x_2)}{\partial q_4}
\end{bmatrix} = \begin{bmatrix}
-d_2 \sin q_{12} & d_3 \sin q_{45} \\
d_2 \cos q_{12} & -d_3 \cos q_{45}
\end{bmatrix}
\]  \hspace{1cm} (2.7)
where \( q_c' \) is the previous \( q_c \). For this linkage, numerical solution by numerical methods is sufficiently fast since it is only a two degree of freedom system, the sampling rate is high (greater than 1 kHz), and components of the Jacobian are required for later computations.

### 2.2.2 Distal Link Velocities from Base Link Velocities

If the base link velocities \( \dot{q}_1 \) and \( \dot{q}_5 \) are given, the distal link velocities can be computed from

\[
\dot{q}_2 = \left( \frac{d_1 \sin (q_1 - q_{45})}{d_2 \sin (q_{45} - q_{12})} - 1 \right) \dot{q}_1 + \left( \frac{d_1 \sin (q_4)}{d_2 \sin (q_{45} - q_{12})} \right) \dot{q}_5
\]

\[
\dot{q}_4 = \left( \frac{d_4 \sin (q_{12} - q_5)}{d_3 \sin (q_{45} - q_{12})} - 1 \right) \dot{q}_5 - \left( \frac{d_4 \sin (q_2)}{d_3 \sin (q_{45} - q_{12})} \right) \dot{q}_1
\]

### 2.2.3 Distal Link Accelerations from Base Link Accelerations

If the base link accelerations \( \ddot{q}_1 \) and \( \ddot{q}_5 \) are given, the distal link velocities can be computed from

\[
\ddot{q}_2 = \frac{d_1}{d_2} \left[ \frac{\cos (q_1 - q_{45})(\dot{q}_1 - \dot{q}_{45}) - \sin (q_1 - q_{45}) \cos (q_{45} - q_{12})(\dot{q}_{45} - \dot{q}_{12})}{\sin (q_{45} - q_{12}) \sin^2 (q_{45} - q_{12})} \right] \ddot{q}_1 +
\left( \frac{d_1 \sin (q_1 - q_{45})}{d_2 \sin (q_{45} - q_{12})} - 1 \right) \ddot{q}_1 +
\frac{d_1}{d_2} \left[ \left( \frac{\cos (q_4) \dot{q}_4}{\sin (q_{45} - q_{12})} - \frac{\sin (q_4) \cos (q_{45} - q_{12})(\dot{q}_{45} - \dot{q}_{12})}{\sin^2 (q_{45} - q_{12})} \right) \ddot{q}_5 +
\left( \frac{\sin (q_4)}{\sin (q_{45} - q_{12})} \right) \dddot{q}_5 \right]
\]
\[ \dot{q}_4 = \frac{d^4}{d\theta^3} \left[ \frac{\cos(q_{12} - q_5)(\dot{q}_{12} - \dot{q}_5)}{\sin(q_{45} - q_{12})} - \frac{\sin(q_{12} - q_5)\cos(q_{45} - q_{12})(\dot{q}_{45} - \dot{q}_{12})}{\sin^2(q_{45} - q_{12})} \right] \dot{q}_5 + \\
\left( \frac{d^4 \sin(q_{12} - q_5)}{d^3 \sin(q_{45} - q_{12})} - 1 \right) \ddot{q}_5 - \\
\frac{d^4}{d^3} \left[ \frac{\cos(q_2)\dot{q}_2}{\sin(q_{45} - q_{12})} - \frac{\sin(q_2)\cos(q_{45} - q_{12})(\dot{q}_{45} - \dot{q}_{12})}{\sin^2(q_{45} - q_{12})} \right] \ddot{q}_1 + \\
\left( \frac{\sin(q_2)}{\sin(q_{45} - q_{12})} \right) \dddot{q}_1 \] (2.12)

### 2.3 Jacobians

The Jacobian is a linear mapping which relates forces and velocities between the joint space and some working space, usually a global coordinate system. If we let \( \mathbf{q} \) be a vector of joint values, and \( \mathbf{x} \) be a vector of world coordinates, the following relations hold

\[ \mathbf{x} = \mathbf{F}(\mathbf{q}), \quad \dot{\mathbf{x}} = \mathbf{J} \dot{\mathbf{q}}, \quad \mathbf{f} = \mathbf{J}^{-T} \mathbf{\tau}, \quad \mathbf{J} = \frac{\partial \mathbf{F}}{\partial \mathbf{q}} \] (2.13)

where

- \( \mathbf{F} \) = the forward mapping from joint space to world space,
- \( \mathbf{J} \) = the Jacobian matrix,
- \( \mathbf{f} \) = forces vector in world space,
- \( \mathbf{\tau} \) = torques vector in joint space,

One method of computing \( \mathbf{J} \) by-passes the necessity of differentiating complicated forward mappings by making use of the fact that the system is composed of connected rigid bodies, and is given by (see for example [SV89])

\[ \mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 & \cdots & \mathbf{J}_n \end{bmatrix} \] (2.15)

where

\[ \mathbf{J}_i = \begin{bmatrix} \mathbf{z}_{i-1} \times (\mathbf{o}_n - \mathbf{o}_{i-1}) \\ \mathbf{z}_{i-1} \end{bmatrix} \] (2.16)
Chapter 2. The “P2 Pantograph” Haptic Device

\[ z_i = \text{joint axis for } q_{i+1}, \text{ expressed in the world frame} \]  
\[ o_i = \text{location of } i'th \text{ origin, expressed in the world frame} \]  

This method uses the forward mappings directly to compute \( J \). However, due to the simplicity of the pantograph geometry, partial derivatives can be used in this case.

Since the pantograph is a two-degree of freedom device with actuation and position encoding on the two ground-fixed links, we require a Jacobian which relates the joint properties (forces and velocities) at these two joints to the handle properties. However, in order to derive this Jacobian (\( \in \mathbb{R}^{2 \times 2} \)), we will first compute the Jacobian (also \( \in \mathbb{R}^{2 \times 2} \)) for each of the two two-link, ground-fixed chains comprising the linkage, then show how these two Jacobians are related to the Jacobian between the two base joints and handle. The pantograph will be considered as two chains connected at the handle, as shown in Figure 2.3.

2.3.1 Handle to Base Joints for Individual Chains

We will now compute the Jacobians for the two ground-fixed chains, denoted by \( J_1 \) and \( J_2 \), as shown in Figure 2.3. By taking partial derivatives of the forward mapping, given in section 2.2.1, we get

\[
J_1 = \begin{bmatrix}
  d_1 \sin q_1 - d_2 \sin q_{12} & -d_2 \sin q_{12} \\
  d_1 \cos q_1 + d_2 \cos q_{12} & d_2 \cos q_{12}
\end{bmatrix} \tag{2.18}
\]

\[
J_2 = \begin{bmatrix}
  d_4 \sin q_5 - d_3 \sin q_{45} & -d_3 \sin q_{45} \\
  d_4 \cos q_5 + d_3 \cos q_{45} & d_3 \cos q_{45}
\end{bmatrix} \tag{2.19}
\]
2.3.2 Handle to Base Joints for the Closed Chain

We now derive the Jacobian which projects the base joint properties to the handle. Note that \( q \in \mathbb{R}^{2 \times 2} \) now refers to the two base joint angles. If \( J_1 \) and \( J_2 \) are the Jacobians for the two individual chains, then the following relations hold:

\[
\begin{align*}
  f_1 &= (J_1)^{-T} \tau_1, \quad f_2 = (J_2)^{-T} \tau_2 \\
  f &= f_1 + f_2,
\end{align*}
\]

The total force at the handle is \( f = f_1 + f_2 \), resulting in

\[
\begin{align*}
  f &= (J_1)^{-T} \tau_1 + (J_2)^{-T} \tau_2 \\
  \text{Let}
\end{align*}
\]

\[
\begin{align*}
  \tau_1 &= \begin{bmatrix} (\tau_1)_1 \\ (\tau_1)_2 \end{bmatrix}, \quad \tau_2 = \begin{bmatrix} (\tau_2)_1 \\ (\tau_2)_2 \end{bmatrix} \\
  (J_1)^{-T} &= \begin{bmatrix} (J_1^{-T})_1 & (J_1^{-T})_2 \end{bmatrix}, \quad (J_1)^{-1} = \begin{bmatrix} (J_1^{-1})_1 \\ (J_1^{-1})_2 \end{bmatrix}
\end{align*}
\]

where \((J^{-T})_i\) is the \( i \)th column of \((J)^{-T}\), and \((J^{-1})_i\) is the \( i \)th row of \((J)^{-1}\) (note that bold fonts imply row or column vectors).

If \((\tau_1)_1\) and \((\tau_2)_1\) are the base joints, then, since there is no actuation on the second joint of either chain \(((\tau_1)_2 = (\tau_2)_2 = 0)\), it can be seen that

\[
\begin{align*}
  f &= (J_1^{-T})_1(\tau_1)_1 + (J_2^{-T})_1(\tau_2)_1 = \begin{bmatrix} (J_1^{-T})_1 & (J_2^{-T})_1 \end{bmatrix} \begin{bmatrix} (\tau_1)_1 \\ (\tau_2)_1 \end{bmatrix}
\end{align*}
\]
Therefore,

\[
J^{-T} = \begin{bmatrix} (J_1^{-T})_1 & (J_2^{-T})_1 \end{bmatrix}
\]  
(2.25)

and

\[
J^{-1} = \begin{bmatrix} (J_1^{-1})_1 \\ (J_2^{-1})_1 \end{bmatrix}
\]  
(2.26)

and we need to compute the first row of the inverse Jacobian of each individual chain.

### 2.4 Pantograph Dynamics

The dynamic properties of the links were determined and an SD/Fast \(^2\) model was constructed (see appendix A for methodology). It was verified that the inertial properties were sufficiently small and constant to be ignored for the purposes of this thesis (which is consistent with the linkage design). The effective mass of the handle was determined by applying to it a simulated force in different directions and computing the acceleration magnitude. The mass was found to be about 120 to 260 grams at the workspace center, centripetal forces were ignored, and there are no Coriolis forces for the planar linkage. The inertia was not further considered in this thesis.

\(^2\)SDFast is a dynamic system modelling software package produced by Symbolic Dynamics Inc.
Chapter 3

A Stochastic Approach to Haptic Texturing

3.1 Introduction

We now consider the generation of texture contact forces for two sliding objects. The nature of these forces is determined by the physical characteristics of the contacting bodies, for example the “average” surface roughness of the pair. Various physical measures of surface roughness exist, and these can be used to produce the frequency and magnitude of the pulses necessary to depict the forces which would result from the actual sliding. However, the micro-modelling of actual collisions to create trajectories would be computationally prohibitive for real-time applications. It will suffice to produce haptic stimulation which gives the correct psychological illusion given the limitations on the user’s perceptual acuity and ability to spatially and temporally connect surface features to individual haptic events. Thus, although the haptic device can not generally accurately track the roughness-induced micro-motion of the actual system trajectory, the interaction is enriched by depicting the essence of the scene events, with the limitations of the user’s haptic input channels in mind. It is proposed that human visual acuity is unable to correlate the individual micro-surface details which cause the individual force impulses either spatially or temporally for typical surfaces. For example, consider some typical object sliding across a table where small textural impulse forces are haptically perceived, but due to the size, number and partial or complete visual obstruction of the
actual surface asperities causing the force impulses, there is no spatial or temporal correlation between them observed by the individual. Therefore, the sense of texture can be treated as characterized by some macroscopic properties.

Note that we treat the simulation of texture forces separately from display. We create a model of texture forces which is designed to produce the forces which an object would experience during sliding contact, thus generating simulated texture. A method is also presented which allows the simulated texture forces to be displayed on a haptic device. No attempt is made at tracking the state trajectory which the texture forces create.

3.2 Related Work

3.2.1 Textures in Graphics

Texture generation and mapping has received considerable attention in graphics. As described by Heckbert [Hec86], the traditional graphical texturing problem comprises mapping a defined texture from some convenient space (called the texture-space), to the screen-space. Lewis [Lew89] surveys methods based on noise, while Perlin [Per85, Per89] presents noise-based techniques which bypass texture space. In Perlin's technique, a simple noise function is defined as

\[ d = \text{noise}(x, y, z) \]

where the parameters \( x, y, z \) are object coordinates and \( d \) is applied at the time the object is rendered. By perturbing surface normals with the noise derivatives, various surface effects were presented, including "bumpy" and "stucco". Fournier et. al [FFC82] studied the use of stochastic models under the observation of self-similarity in many physical systems, such as terrain representation, where randomly distributed overall properties dominate macroscopic properties. Fournier et. al considered fractal Brownian processes, which have the property of self-similarity, but are computationally expensive. As
presented in this thesis, the idea of stochastic texture generation also has application in haptic texturing.

### 3.2.2 Textures in Haptics

Sakaguchi [Sak94] describes a system for automatically detecting and classifying surface texture, without duplicating the texture through a force display device. De Rossi [DR91] classified various sensory needs with respect to specific manipulation tasks, and described instrumentation technologies to measure surface properties in relatively unstructured teleoperation environments. Lederman et al. [LK87] studied the importance of hand motion in obtaining object information through haptic channels. Minsky et. al. [MOyS+90] studied force display by creating texture maps consisting of spring approximations to local gradients of "bumps". They also investigated the correlation between visual and haptic acuity and some of the kinesthetic parameters relevant to force display. Drucker [Dru88] presents a deterministic analysis of texture based on the elastic properties of materials and their effect on sensors. Hirota [HH93] describes a deterministic method to generate surface "pseudo-texture" which is similar to texture mapping in graphics, and a mechanism which uses oscillations of about 1mm at 30Hz to generate the textural feel.

### 3.3 Haptic Textures

#### 3.3.1 Surface Roughness Model

Early work in the area of friction and surface roughness used a form of overlay model to represent surface texture. For example, Schurig [Sch40] defined a surface as consisting of "waviness", "roughness" and "surface flaws" (see Figure 3.4 and Figure 3.5). Although Schurig described only roughness as being superimposed on waviness, the surface could
be described as surface flaws overlain on roughness overlain on waviness. Methods of measuring surface characteristics are reviewed in [WT80], and a statistical treatment is given in [Tho82]. The two prominent surface height characterization measures are root-mean-square deviation ($R_q$) and center-line average ($R_a$), which measure the variance of surface height asperities from a straight, average datum line. The measures are defined from

\[ R_q^2 = \frac{1}{L} \int_0^L z^2 dx \]  
\[ R_a^2 = \frac{1}{L} \int_0^L |z| dx \]  

where $x$ is the profile direction and $z$ is the surface height from the datum line, and the datum line is chosen to minimize these measures.

For our purposes, the most significant observation is that many surface preparation methods result in roughness profiles which are approximately Gaussian, as shown in Figure 3.6 (see [Tho82]). Although practical surfaces can deviate from the Gaussian
ideal, we will restrict our analysis to the Gaussian profile.

Consider a point moving laterally along the surface at speed $v$. If the surface height is sampled at points $v \times n \times dt$, where $n$ is a positive integer and $dt$ is the sampling period, a Gaussian histogram would result. Therefore, if our low-level control loop runs at time increments of $dt$, each control cycle should see a surface asperity with magnitude drawn from the Gaussian distribution. Note that this is independent of the speed at which the sampling occurs, since re-sampling a Gaussian distribution should result in a Gaussian distribution with the same mean and standard deviation (since we assume that the surface asperities are independent and identically distributed). Therefore, the same Gaussian distribution is used regardless of the sampling time. However, in order to generate more realistic textures, it may be beneficial to modulate the texture pulse frequency based on the relative sliding speed of the objects.
3.3.2 Haptic Texture Definition

The forces of interaction between sliding surfaces are a complex function of the material properties and the surface geometry of the contacting bodies. The surface contact force can in general be decomposed into orthogonal normal and lateral components, and results from the contact and deformation of micro-surface asperities as two objects slide over each other under load. In modelling an object surface, decisions must be made about the level of detail in the model. For example, if using polygonal approximations, the relative size and number of polygons to include must be decided. The haptic texture is defined to be all the effects which are not explicitly accounted for by traditional rigid body contact normal (constraint) and lateral (friction) forces. To determine the level of detail, it is useful to consider the size of the contact patch between the sliding bodies. Texture forces will be caused by those surface profile components with spatial period lower than the dimensions of the contact patch.

To facilitate the determination of appropriate texture generation parameters, we make
the following assumptions:

- During sliding between surfaces, normal texture forces are generated due to interactions between surface asperities, with magnitude proportional to the height of the asperities.

- Lateral texture forces are generated proportional to the above normal forces.

The first assumption is justified when considering any two contacting surface asperities sliding over each other. Force components are generated in normal and lateral directions as the asperities slide relative to each other. The second assumption follows from the first in that it is consistent with Coulomb friction. No attempt is made to model the individual surface asperity contacts which are produced during sliding. Such an approach is considered intractable not only because of the computational expense, but because conventional sensing and control technologies would have difficulty accurately reproducing any resulting trajectories to the small scales to which they would have to be computed. Instead, our approach will concentrate on isolating and using physically meaningful surface roughness parameters, with the result that the texture mapping will provide realistic feel to the user.

3.4 Force Decomposition

Figure 3.7 shows a decomposition of the contact force. The following model is used (see Appendix B for statistics notation):

$$F_{\text{contact}} = F_{\text{constraint}} + F_{\text{friction}} + F_{\text{texture}}$$

(3.29)

where

$$F_{\text{texture}} \sim u \text{Normal}(\mu_{\text{feature}}, \sigma_{\text{feature}}^2)$$

(3.30)
Chapter 3. A Stochastic Approach to Haptic Texturing

The generation of normal constraint forces is discussed in detail in Section 4.9, but can be generated using a PD control approach for haptic devices (for instance, see Salcudean and Vlaar [SV94]). This component creates the illusion of surface hardness by producing forces which oppose interpenetration of simulated contacting bodies. With only this component, a frictionless, smooth surface, such as ice, is generated. The tangential Coulomb friction opposes tangential motion along the surface. Adding this component creates a smooth frictional surface, such as glass. The texture forces are then added to give surface roughness characteristics. They can be related to the $R_q$ and $R_a$ parameters in that both are statistical measures of variance, and the variance of the statistical haptic texture ($F_{\text{texture}}$) is proportional to the $R_q$ and $R_a$ variances.

\[ u = \begin{cases} 
0 & \text{for } |v| < \epsilon \\
1 & \text{otherwise} 
\end{cases} \quad (3.31) \]

where $v$ is the lateral speed, and $\epsilon$ is a small positive number. This provides a small deadband around a stationary position, improving stability. The choice of $\mu$ and $\sigma$ will be determined by surface roughness characteristics and in the initial implementation, $\mu$ was taken as zero and $\sigma$ was varied to produce different textural feels. It was predictably found that higher $\sigma$ leads to a sensation of a rougher surfaces. The actual $\sigma$ would ultimately be taken from surface roughness literature for different surfaces.
3.5 Generating Gaussian Texture

A realistic surface has many contacts occurring during motion. It is therefore desirable to compute the texture forces at relatively high frequencies. In our initial implementations a frequency of 1 KHz was used to produce effective textural feel.

The contacting bodies are generally described in a Cartesian frame. The statistical properties of the texture forces will thus be described in a Cartesian frame called the surface-space, with one axis perpendicular to the contact surface. These properties will then be projected to the world-space, which is a convenient inertial reference frame. From the world frame, the force properties are projected to joint-space. Once the texture force statistical properties are known in joint-space, actuation forces can be generated with the mapped statistical properties. Therefore, the necessary texture forces can be generated directly in joint-space, and there is no need to map individual forces from the surface space to the joint space. The reader is referred to the appendices for relevant statistical concepts.

3.5.1 Generating Texture Forces in World-Space

The texture properties are defined relative to an object's surface. The object’s surface-space can thus be mapped to world-space. Let \( R \) be the orientation mapping from the surface-space to the world-space

\[
B_w = RB_s
\]  

(3.32)

where \( B_w \) and \( B_s \) are the matrices whose columns are the unit vectors of the world-space and surface-space, respectively.
Then forces in surface-space map to forces in world-space according to

\[ \mathbf{F}_w = R^T \mathbf{F}_s \]  

(3.33)

where \( \mathbf{F}_s \) is the force in surface-space, and \( \mathbf{F}_w \) is the same force mapped to world-space.

Then, the statistical properties of \( \mathbf{F}_w \) are as follows (see Appendix B).

\[
\begin{align*}
F_w \sigma^2 & = R^T (F_s \sigma^2) R \\
F_w \mu & = R^T (F_s \mu)
\end{align*}
\]  

(3.34) (3.35)

The properties of the texture can now be defined in the invariant surface-space and the stochastic behaviour mapped to the world-space. This removes the computational cost of mapping forces between spaces for each control cycle. If the lateral texture force variances are made functions of the normal constraint force, then, under the assumption that the change in normal force is low-bandwidth (since it is limited by the bandwidth of a human user), the computation of variance matrices is more efficient than producing and mapping individual forces - the normal force changes relatively slowly so the covariance matrix needs to be updated at low bandwidth.

### 3.5.2 Generating World-Space Forces in Joint Space

To implement the algorithm in low-level control, instead of generating the actual texture impulses in world-space one can generate only the necessary mean and covariance matrices for a given world-space region, and map those parameters to joint-space through the Jacobian. To achieve this, consider the following.

Let

\[ \mathbf{\Gamma} = J^T \mathbf{F} \]  

(3.36)
where \( \mathbf{F} \) is a world-space force, and \( \mathbf{T} \) is the corresponding joint-space torque vector. Restricting the analysis to three dimensional forces, and \( n \) dimensional joint-space,

\[
\Gamma = \begin{bmatrix}
\tau_1 \\
\tau_2 \\
\vdots \\
\tau_n
\end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix}
f_1 \\
f_2 \\
f_3
\end{bmatrix}
\]

Then, if \( \mathbf{F} \) has the form

\[
\mathbf{F} \sim \text{Normal}(\mu, \sigma)
\]

\[
(3.37)
\]

\( \mathbf{F} \) has the form

\[
\mathbf{F} \sim \text{Normal}(\mu, \sigma)
\]

\[
(3.38)
\]

To obtain the joint-space covariance matrix, \( \Gamma \sigma^2 \), begin by defining the covariance matrix for \( \mathbf{F} \) as

\[
\Gamma \sigma^2 = \begin{bmatrix}
\Gamma_{11} & \Gamma_{12} & \Gamma_{13} \\
\Gamma_{21} & \Gamma_{22} & \Gamma_{23} \\
\Gamma_{31} & \Gamma_{32} & \Gamma_{33}
\end{bmatrix}
\]

\[
(3.40)
\]

where \( \Gamma_{ij} \) is the covariance of the \( i \)th and \( j \)th components of the world force \( \mathbf{F} \), as defined on the object's surface.
Then, the joint-space covariance matrix is given by

\[ \Gamma \sigma^2 = \begin{bmatrix}
\tau \sigma_{11} & \tau \sigma_{12} & \cdots & \tau \sigma_{1n} \\
\tau \sigma_{21} & \tau \sigma_{22} & \cdots & \tau \sigma_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\tau \sigma_{n1} & \tau \sigma_{n2} & \cdots & \tau \sigma_{nn}
\end{bmatrix} = J^T (F \sigma^2) J \]  

(3.41)

To obtain \( \Gamma \mu \), define the vector of world-space force component averages

\[ F \mu = \begin{bmatrix}
J \mu_1 \\
J \mu_2 \\
J \mu_3
\end{bmatrix} \]  

(3.42)

Then,

\[ \Gamma \mu = J^T (F \mu) \]  

(3.43)

Gaussian deviates can be generated in joint-space with the mapped mean and covariance and added directly to the actuation. This will be valid in so far as the Jacobian remains the best linear mapping between the world and joint spaces. The system is BIBO stable, and the textures only add uncorrelated input noise.

### 3.5.3 Input Parameters to the Gaussian Texture Functions

The initial implementation used zero mean with different standard deviations, and the evaluation criterion was the subjective impression of a user. However, ultimately surface roughness parameters as quoted in surface roughness literature would be the appropriate values to use. However, the electro-mechanical limitations of haptic interfaces must be realized, and additional scaling of the parameters for any specific device may be required to achieve a realistic feel.
3.6 Implementation

3.6.1 Testbed

The algorithm was implemented on the two-degree-of-freedom P2 Pantograph device.

3.6.2 Results

Using this algorithm, it was possible to generate texture force pulses at approximately 1kHz using only the 16MHz, fixed-point MC68332 processor. The results are tested informally using a small number of subjects. Careful psycho-physical evaluation awaits further research.

Lateral Texture Forces Only

A model with only lateral texture forces (similar to that of Minsky et al [MOyS+90]) was implemented where the texture impulses were created with uncorrelated world-space components.

In order to generate lateral texture, the handle of the pantograph was placed in contact with various objects. To get a very lightly textured surface, the handle was placed in contact with the machined finish of the workspace. To get heavier textures, sheets of sandpaper were placed beneath the pantograph handle such that contact existed between the handle and paper (see Figure 3.8). Thus, the experimental set-up corresponded to a simulation of an identical device. Three subjects participated in these experiments. Representation of surfaces with very light coarseness was found to be reasonably good. Although a reasonable textural feel was produced to emulate very coarse texture, comparison with actual contact with sandpaper revealed that stick-slip phenomena form an important component for effective textural representation of very coarse surfaces.

Next, the workspace of the pantograph device was seperated into four quadrants,
with each quadrant given a different texture force variance. One quadrant was given no
texture, and acted as a control. Three subjects were asked to explore the workspace,
and subjectively grade the quadrants for roughness. It was found that subjects could
effectively determine that one quadrant was rougher than another, and could thus grade
the quadrants from no texture to heaviest texture.

A Straight Edge with Normal and Lateral Texture Forces

A straight edge was implemented combining normal constraint and texture impulses with
tangential texture impulses.

A blind experiment was then conducted where the experimenter randomly chose be­
tween a texture model with covariant or independent world texture forces. The subject
was asked to determine whether the texture forces were independent, or correlated.

The covariance between normal and lateral texture impulse components (non-zero
off-diagonal terms in the world-space covariance matrix) did not produce a detectably
different feel versus independent world-space texture impulses (diagonal world-space co­
variance matrix). This suggests that it is generally unnecessary to transform the texture
from the object-space to world-space, and that a diagonal covariance matrix can be maintained in world-space. This saves the computation steps of mapping the texture from object-space to world-space, as well as making the joint-space mapping less costly since the world-space covariance matrix remains diagonal.
Chapter 4

Internal Friction Contact Model

4.1 Introduction

In the previous chapter, we generally took the normal constraint force as given by a spring/damper model. We now investigate the simulation of contact constraint forces in more detail, and expose some of the deficits in the spring/damper model. A new model for constraint forces will emerge, which corrects the most serious deficiencies of the spring/damper model, but maintains its implementation simplicity.

We will begin with a general discussion of contact simulation using the Euler-Lagrange formulation. We then discuss contact forces in general, beginning with the coefficient of restitution, which is an empirical parameter associated with contact phenomena. This coefficient will be extensively used in the discussions to follow. We will then survey some of the existing techniques for contact simulation, and propose a new, internal friction model which will be inspired by surface physics, and endeavours to correct the most serious flaws of the spring/damper model. Implementations of the internal friction model will be compared with a spring/damper model using comparable model parameters, and it will be shown that both provide equally accurate impact results. However, the internal friction model will not suffer from an inability to absorb energy during low-speed, non-impact quasi-static cycling, and the internal friction model can better preserve the coefficient of restitution during multi-contact point conditions.
4.2 Dynamic Simulation - Euler-Lagrange Formulation

We now present the dynamic simulation method which will be used in the thesis.

4.2.1 Constrained Dynamics

In this technique, a dynamic system is formulated as

\[
\begin{align*}
\dot{q} &= v \\
M(q)\dot{v} &= f(q,v) - G^T(q)\lambda \\
g(q) &= 0
\end{align*}
\]

(4.44) \hspace{1cm} (4.45) \hspace{1cm} (4.46)

where

- \( q \in \mathbb{R}^{n_q} \) are the generalized coordinates,
- \( \lambda \in \mathbb{R}^{n_\lambda} \) is the Lagrange multiplier vector (also constraint forces),
- \( G = \frac{\partial g(q)}{\partial q} \),
- \( M \) = mass matrix.

To modify this traditional form for use in contact problems, we make the following observations. The constraint \( g(q) = 0 \) is replaced by \( g(q) \geq g_{eq} \) for some equilibrium constraint value \( g_{eq} \). The constraint will typically be the intrusion distance of a polyhedron vertex. As explained in section 4.4, for soft contact, where interpenetration is allowed, \( g_{eq} < 0 \). For hard contact, where interpenetration is not allowed, \( g_{eq} = 0 \). Also, since the number of contact points \( n_\lambda \) can vary, the number of columns in \( G^T \) varies accordingly. Implementation methodology is now presented for the planar case.
4.2.2 Planar Implementation of Contact

Canny [Can86] uses a quaternion approach to determine object kinematics in three dimensions. In particular, he identifies the following three types of collisions

- Type A: obstacle vertex against object face
- Type B: object vertex against obstacle face
- Type C: object edge against obstacle edge

In this analysis, we restrict ourselves to Type B contact, noting that Type A and Type C analysis follows in a similar manner. Type C is somewhat troublesome since there is no well defined direction for the contact constraint force. In the planar case, Type C contact consists of vertex against vertex, and the best choice for the contact tangent plane has a normal colinear with a line between the contacting vertices. In the general case, the tangent plane could be the unique plane with normal perpendicular to both contacting edges.

Assume the object under consideration has \( n_v \) vertices (labeled from 0 to \( n_v - 1 \)), and it can contact only planar surfaces with no tangential friction, as shown in Figure 4.9. If \( z \) represents the state vector, and \( q \) the generalized coordinates, then the system is described by

\[
\begin{bmatrix}
\dot{z}_0 \\
\dot{z}_1 \\
\dot{z}_2 \\
\dot{z}_3 \\
\dot{z}_4 \\
\dot{z}_5
\end{bmatrix} =
\begin{bmatrix}
q_0 \\
q_1 \\
q_2 \\
\dot{q}_0 \\
\dot{q}_1 \\
\dot{q}_2
\end{bmatrix}
\begin{bmatrix}
\text{orientation} \\
x \text{ component of mass center location} \\
y \text{ component of mass center location} \\
\text{angular velocity} \\
x \text{ component of velocity} \\
y \text{ component of velocity}
\end{bmatrix}
\]

(4.47)
Figure 4.9: Planar Objects Contacting Planar Obstacles

\[ \dot{z} = \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ M^{-1}(f - GT\lambda) \end{bmatrix} = \begin{bmatrix} z_3 \\ z_4 \\ z_5 \\ M^{-1}(f - GT\lambda) \end{bmatrix} \]

\[ g(q) \geq g_{eq} \text{ where } g_{eq} \leq 0 \] (4.49)

\( M \in \mathbb{R}^{3\times3} \) is the mass matrix given by

\[ M = \begin{bmatrix} I & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \] (4.50)

where

\[ I = \text{moment of inertia} \] (4.51)

\[ m = \text{mass} \]

Also, \( G \in \mathbb{R}^{n_\lambda \times 3} \) is the constraint Jacobian with respect to the generalized coordinates, \( g \in \mathbb{R}^{n_\lambda \times 1} \) are the constraints, and \( g_{eq} \in \mathbb{R}^{3\times1} \) is the vector of equilibrium constraints.
Note that in a dynamic simulation where contacts are made and broken, \( n \) is variable. To keep track of contacts, we define the notation \( g_{ij} \) to be the contact constraint between obstacle vertex \( i \) and obstacle plane \( j \).

Let each vertex \( i \) be represented in the world coordinate system by \( p_i \), let the unit normal of obstacle plane \( j \) be \( w_j \) and let the number of obstacle planes be \( n_o \). Let \( c \) be the object's mass center location, with

\[
    c = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}.
\]

Then, the columns of \( G^T \) (which are the gradient vectors at each contact point) are given by

\[
    v_{ij} = \begin{cases} 
        (p_i - c) \times (w_j) & \text{if } g_{ij} \leq 0 \\
        (w_j) & \text{otherwise}
    \end{cases}
\]

Then \( G^T \) is given by

\[
    G^T = \left\{ v_{ij}; i = 0, \ldots, n_v - 1, j = 0, \ldots, n_o - 1 \right\}
\]

In words, \( G^T \) is composed of the set of gradient vectors where constraint violation is known to exist. The reader should note that this method effectively computes the constraint gradient, as defined by

\[
    G = \frac{\partial g}{\partial q}
\]

without complicated partial differentiation. It is essentially the same trick presented in section 2.3 for computing the Jacobian of serial, rigid body chains without differentiating.
the forward mapping. However, in section 2.3 angular velocity terms were stacked below linear velocity terms in the velocity vector, whereas here angular velocity terms are above linear velocity terms. We have chosen this convention for similarity to Lilly's spatial notation [Lil93], which is likely to produce the cleanest formulation for multi-link manipulators undergoing contacts. This form for the gradient generalizes well for the six-degree-of-freedom, non-planar problem.

Note that up to this point, nothing has been said about how to compute the constraint force vector $\lambda$. We now turn our attention to this task.

### 4.3 Coefficient of Restitution

The coefficient of restitution, $r$, is an empirical property which characterizes the behaviour of objects during collisions. It is a function of many parameters, the most notable being material properties, although there is no definitive description for how it relates to the details of a contact event. Three different interpretations of the coefficient are presented in this section. An important property of the coefficient, noted as early as 1834 by Hodgkinson [Hod34], is that its value remains approximately constant for different, slow impact speeds.

#### 4.3.1 Newton's Hypothesis

Newton postulated that the relative normal velocity at the point of contact for an arbitrary collision is governed by the relation

$$v' = r v$$  \hspace{1cm} (4.56)  

where

$v'$ = post-impact relative normal velocity of the point which comes into contact.
\( v \) = pre-impact relative normal velocity of the point which comes into contact.

\( r \) = coefficient of restitution.

### 4.3.2 Poisson's Hypothesis

Poisson postulated that the ratio of the total impulse during restitution to the total impulse during compression obeys the equation

\[
r = \frac{\int_{t=T}^{T_{\text{final}}} F_{\text{restitution}} \, dt}{\int_{t=0}^{T} F_{\text{compression}} \, dt}
\]

Bhatt, et. al. [BK94] use this hypothesis.

### 4.3.3 Energy Hypothesis

The energy hypothesis proposes that the coefficient of restitution relates the ratio of the total energy absorbed from the impacting bodies (during the compression phase), to the total energy returned (during the restitution phase) through the equation

\[
r^2 = \frac{\text{energy returned}}{\text{energy absorbed}}
\]

Many authors [BJ84], [Str90] have stated that the three interpretations provide identical results under certain conditions, including: central or co-linear impact (when the centres of mass align with the impaction surface normal); frictionless impact of smooth bodies [BJ84]; impact of smooth bodies where a Coulomb tangential friction force exists, but the relative sliding at the point of contact does not reverse during the impact [Str90]. The energy interpretation is often seen as the most useful and fundamental [Bra84].
4.3.4 Computation for Objects of Different Material Properties

As stated in [Wan89], the effective coefficient of restitution for two contacting objects of different material properties is

\[ r = \frac{r_1 Y_1 + r_2 Y_2}{Y_1 + Y_2} \] (4.59)

where

\[ r_i = \text{coefficient of restitution of material } i \] (4.60)
\[ Y_i = \text{Young's Modulus of Elasticity for material } i \]

4.4 Existing Methods for Contact Simulation

Contact simulation is generally divided into soft and hard contact. Soft contact models allow interpenetration of bodies, and compute a contact force based on the state of interpenetration. The measure for the amount of interpenetration varies, but is often based on polygonal approximations to surfaces, with the depth of vertex interpenetration the most common measure.

Hard contact models do not allow interpenetration, and frequently are designed to deal exclusively with impact. This method treats objects as infinitely rigid, and does not permit interpenetration. As a result, contacts occur instantaneously, and impulse forces can be infinite, although the impulse itself remains bounded (the impulse force versus time is of Dirac delta form).

4.4.1 Finite Element Modelling

Thorough analysis of the contact problem has been conducted using finite element techniques [KO88]. These techniques can provide a very detailed and accurate model of the contact forces and displacements for arbitrary surface profiles, but suffer from extreme
complexity and computational cost. They are not currently applicable for real-time simulation applications for this reason.

4.4.2 Soft Contact

The major advantage to soft contact approaches is their ability to relate existing states to actual contact forces. This gives the methods the ability to propagate forces in realistic ways, in both impact and low-frequency cycling. The major disadvantage is that realistic stiffnesses for the contact forces lead to stiff systems of differential equations, which require more sophisticated integration techniques.

Spring-Damper

Perhaps the most ubiquitous contact model is the spring-damper system. Usually, the model uses polygonal approximations to surfaces, and attaches a simple spring-damper to the maximum interpenetration distance [Kot92, GPS94a, GPS94b]. The method has the advantages of simplicity (it allows a "local" solution) and correct prediction of some collision behaviour; as shown in [SV94] and [GPS94a, GPS94b], the coefficient of energy restitution can be made independent of collision speed with the appropriate choice of spring stiffness and damper viscosity. There is also support in the surface physics literature that a spring/damper is a reasonable model for high speed impact [MSB85]. A major deficit is its inability to correctly predict energy loss during quasi-static load cycling; a spring-damper system predicts no energy loss, while energy loss occurs even during quasi-static cycling [MSB85].
4.4.3 Hard Contact

The major advantage to these techniques is that stiff systems of differential equations can be avoided.

Index Reduction Techniques

The traditional solutions for the constraint forces \( \lambda \) of section 4.2 involve differentiating the constraints twice with respect to time, and solving for \( \lambda \) [Chi95, ACPR93]. The solution is

\[
\lambda = (G M^{-1} G^T)^{-1}(G M^{-1} f + h) \tag{4.61}
\]

\[
h = \frac{d^2}{dq^2}g(q)(v,v) \tag{4.62}
\]

An advantage to the technique is simplicity in modelling relatively complicated systems. A disadvantage is the loss of physical meaning of the Lagrange multipliers. It is more difficult to associate them with physically meaningful processes; for example, during an impact, this method gives no assurance that the correct amount of energy is absorbed. Note that Appendix C proposes a simple method for computing \( G \) and \( h \) without requiring partial differentiation of the constraints with respect to the generalized coordinates.

Quadratic Programming Solution

Chand, et. al. [CHR76] considered the material elastic properties of the contacting objects to formulate the contact problem as a quadratic program. As shown in [GPS94a, GPS94b, Lod84], and as studied extensively by Baraff [Bar89, Bar90, Bar93, Bar94] the objects can also be considered as completely rigid. The normal contact forces can be described by a linear complementarity solution, and following Baraff's notation, if \( g(q) \) and \( \lambda \) are the constraints and constraint forces, respectively, then
where $g_i$ and $\lambda_i$ are the $i$'th components of $g$ and $\lambda$, respectively. In words, only repulsive constraint forces are allowed, and either contact exists with some repulsive constraint force ($g_i = 0, \lambda_i > 0$), or no contact exists and no constraint force exists ($g_i > 0, \lambda_i = 0$). The condition $g_i \geq 0$ can be replaced with $\ddot{g}_i \geq 0$ since $\ddot{g}_i < 0$ implies that the $i$'th contact point is accelerating to violate $g_i \geq 0$.

This results in the system

$$
\begin{align*}
\lambda_i & \geq 0 \\
\ddot{g}(q) & \geq 0 \\
\dddot{g}_i,\lambda_i &= 0
\end{align*}
$$

for which the solution is

$$
\min_{\lambda} \lambda^T (A\lambda + b) \text{ subject to } \begin{cases} A\lambda + b \geq 0 \\ \lambda \geq 0 \end{cases}
$$

where $A$ and $b$ are linear transformations of $M$ and $f$.

**Impulse Force Computation**

Stronge [Str90] investigated planar impacts and computed momentum changes given the coefficient of restitution under different contact geometries. Stronge’s methodology does not extend easily to non-planar impacts, however. Mirtich [Mir95] has proposed a model of impact which reduces stiffness by parameterizing the contact with the surface normal.
force. Stationary objects are furthermore envisaged as constantly moving in “micro” collisions with the environmental constraints. Although the method of reducing contact stiffness during impacts by parameterizing with the normal force is useful, it is not clear that the micro-collision model of stationary objects is the most efficient.

4.5 Surface Contact

In the study of contact surface dynamics as related to robotic control, surfaces have predominantly been modelled as spring-damper systems. Salcudean and Vlaar [SV94] related the coefficient of restitution to the damper’s energy dissipation and noted that a “hard” surface contact could be effectively simulated with a haptic interface by applying a braking force at the instant of contact. Shoji and Inaba [SIFH90, SIF91] use a spring-damper system where the spring force and the energy dissipation are functions derived from Hertzian contact stress theory. Kotoku [Kot92] considers a remote manipulator where all environmental contact is modelled with spring/dampers.

To investigate the general nature of contact, we must identify the relationship between normal stress, denoted \( \sigma \), and force, denoted by \( F \). The net normal constraint force is the integral of the normal stress over the contact patch. Therefore, we can quite safely switch between stress and force, remembering that one is the integral of the other. Furthermore, the normal stress is variable over the contact patch, and any equation using \( \sigma \) denotes some average stress. Since we only seek average values in the following analysis, we can essentially treat normal force and stress as being interchangeable. We must also clarify the inter-relation of strain and penetration. Strain is the elasto-plastic deformation of a real material under contact loads, and is usually proportional to stress. In our polyhedral models, we will use penetration as a surrogate for strain. Strain and penetration will thus be used interchangeably.
In the more general study of interfacial contacts, the coefficient of restitution has been related to modifications of the spring damper system using power laws for the damping coefficient and the spring displacement [HC75]. This work states that the coefficient of restitution is a decreasing function of impact velocity (as long as the velocity does not exceed some maximum value), given approximately by

\[ r = 1 - a \, v_i \]  \hspace{1cm} (4.70)

where

\[ a = \text{constant} \ (0.08-0.32 \text{ m/sec for steel, bronze and ivory}) \]
\[ v_i = \text{impact speed} \]

Hunt and Crosley [HC75] attributed energy loss in collisions to heat generation, but attributed the primary transport mechanism to shock waves in the solid; of course the energy of the shock waves must ultimately be dissipated by some other phenomena.

The normal stress has been modelled with power laws for both the spring displacement and damping coefficient, [KO88]. The form of the power law for the normal stress is

\[ \sigma = c \, g^n + b \, g^l \dot{g} \]  \hspace{1cm} (4.71)

where

\[ c = \text{spring-like constant}, \]
\[ b = \text{viscosity coefficient}, \]
\[ n, l = \text{measured parameters}, \]
\[ g = \text{strain}. \]
Figure 4.10: Rate-Dependent Hysteretic Loop

This law produces quasi-elliptical hysteresis loops for dynamic loading, as shown in Figure 4.10 and does not predict hysteretic loss for quasi-static loading.

4.6 Internal Friction

A large body of research has concentrated on the mechanisms by which solids dissipate energy [ECI83, ECI87, ECI81, ECI85]. During contact, energy will be dissipated due to the stresses and strains produced within the solid, and will be transmitted from the contact point through the solid as ultrasonic shock waves or heat. The macroscopic shock waves ultimately attenuate, losing their energy to the microscopic motion of the molecules, a process generally termed heat dissipation. This process has been studied in detail, and the rate of energy loss has been linked to the degree of dislocations in crystalline structures, temperature, magnetism, etc. The main method of measuring the degree of internal friction is the attenuation rate of vibrations within the solid.

As was pointed out by Lazan, et. al. [Laz64], the constitutive equation describing
the damping loss may include terms to account for

"...anelastic and dislocation movements in crystalline materials, interactions among molecular forces in polymers, inhomogeneous plastic slip and flow phenomena, magnetomechanical effects, and general viscoelastic phenomena. This category includes structural metal and nonmetals, viscoelastic materials, elastomers and other rubber-like materials, foamed materials, and fluids."

It is clear that different terms may be required in the constitutive equations for different materials, since a general law is not available, nor would such a law be mathematically tractable. Lazan, et. al. also classified rate-dependent and rate-independent damping effects. Metals and polymers were both stated to have rate-dependent (also called dynamic hysteresis) and rate-independent (also called static hysteresis) effects. The rate-independent effects were attributed predominantly to plastic deformation, and the unique features of the hysteretic loops of the two classes were degenerate elliptical shapes for the rate-dependent effects, and sharp slope discontinuities for rate-independent effects [Laz64](see Figure 4.10 and Figure 4.11).

Thornley, et. al. [TCBK65] studied quasi-static loading characteristics and observed the form of the curves shown in Figure 4.11, if complete separation was allowed between loading cycles. Millet, et. al. [MSB85] considered damping in grey cast iron, and found that low frequency stress cycling of iron produces energy loss which is independent of the cycling frequency. This suggested that the low frequency losses were not dominated by viscous effects, but, rather, solid internal friction losses. A solid friction term, therefore, was added to the classical spring-damper model, giving the following equation

\[
\sigma = K \dot{g} + B \frac{\dot{g}}{|\dot{g}|} + F_{fr} \frac{\ddot{g}}{|\ddot{g}|}
\]  

(4.72)
Chapter 4. Internal Friction Contact Model

Figure 4.11: Rate-Independent Hysteretic Loop

where

\[ \sigma = \text{repulsive force} \]
\[ g = \text{measurement of the contact penetration} \]
\[ K = \text{a spring stiffness parameter} \]
\[ B = \text{a damper viscosity parameter} \]
\[ F_{fr} = \text{a friction stress or force} \]

When cycling frequency is small, the equation reduces to

\[ \sigma = K g + F_{fr} \frac{\dot{g}}{|\dot{g}|} \] (4.74)

4.6.1 Simplified Surface Contact Model

Internal Friction Model Overview

A method will be proposed which is a hybrid of soft and hard contact. The parameters are chosen to allow the same model to be used for both impact and quasi-static loading,
and the form of the model is based on physical models from surface physics literature. It should be noted that choosing the model parameters to satisfy macroscopic behaviour during impacts corrupts the expected performance under low frequency cycling. A more complete treatment would include parameters taken from the surface physics literature. However, it is convenient to use a single model for both impact and low frequency cycling.

**Detailed Derivation**

The internal friction model provides some insight into the way in which real materials behave, and begins to account for the plastic micro-deformation which occurs during the loading of two objects. We now propose a simple model of surface contact, with the realization that contacts under arbitrary control laws may occur arbitrarily slowly. The model will provide the sharp discontinuities characteristic of rate-independent, static hysteresis. Consider the case of a sphere contacting a rigid plane, where quasi-static stresses can be computed from Hertzian contact stress theory. If we envision the solids as composed of a crystalline structure with dislocations, then relative motion between dislocation boundaries would cause "sliding" friction between them, resulting in an irreversible energy loss through an internal friction effect. If the contact occurs sufficiently slowly, for example under an arbitrary control law, then shock waves would not be expected.

We will assume that internal friction is the main energy dissipator under slow contact. The tangential stresses for a sphere contacting a plane are correlated to the normal stress (see, for example, Shigley [Shi86]). If we assume that the internal friction dissipation is correlated to the tangential stress as the molecules move in the normal direction (in a manner similar to Coulomb friction), then we can start with the following simple model. Let the normal stress be given by $F(g)$, where $g$ measures the amount of compression at the surface. We next assume that the tangential stress can be expressed as a fraction of the normal stress, and that the internal friction force is given by a fraction of the
tangential stress, giving the relation \( f F(g) \) for the internal friction force. This internal friction force is assumed to oppose the motion of the molecules, thus opposing the relative normal velocity of the contacting bodies. The contact force is, therefore, given by

**Assumption 1:** Contact Force; \( F = F(g) + f F(g) \frac{\dot{g}}{|g|} \) \hspace{1cm} (4.75)

Because it is well known that the coefficient of restitution is approximately constant (though actually decreases with the impact speed) for sufficiently slow contact speed (< 3 m/s [Wan89]), we will choose \( f \) to preserve this property. Using the energy interpretation of the coefficient of restitution, we separate the contact into compression and restitution phases, with

**Assumption 2:** Energy Hypothesis; \( r^2 = \frac{E_r}{E_c} \) \hspace{1cm} (4.76)

The net forces during compression and restitution are, respectively

\[
F_c = F(g) + f F(g) = (1 + f) F(g) \hspace{1cm} (4.77)
\]

\[
F_r = F(g) - f F(g) = (1 - f) F(g) \hspace{1cm} (4.78)
\]

If we assume that no permanent deformation occurs, then the energies of compression and restitution are, respectively

\[
E_c = (1 + f) \int_0^{g_{max}} F(g) dg \hspace{1cm} (4.79)
\]

\[
E_r = (1 - f) \int_{g_{max}}^0 F(g) dg \hspace{1cm} (4.80)
\]

If

**Assumption 3:** \( F(g) \) is conservative; \( \int_g F(g) dg = 0 \) \hspace{1cm} (4.81)
it then follows that
\[
\begin{align*}
r^2 &= (1 - f)/(1 + f) \\
\beta &= (1 - r^2)/(1 + r^2)
\end{align*}
\] (4.82) (4.83)

The forces for compression and restitution are then given by
\[
\begin{align*}
F_c &= \frac{2}{(1 + r^2)} F(g) \\
F_r &= 2 \frac{r^2}{(1 + r^2)} F(g)
\end{align*}
\] (4.84) (4.85)

With this definition for the internal friction dissipation factor \(f\), the coefficient of restitution will be constant independent of contact speed, as required by empirical evidence for slow contact speeds. The form of \(F(g)\) will be geometry dependent.

### 4.6.2 Choosing the form of \(F(g)\)

The simplest choice would be a Hooke's Law form such as

**Assumption 4: Form for \(F(g)\); \(F(g) = k g\)**

(4.86)

where \(k\) is the spring-like stiffness parameter.

As shown, for example, in Roarke and Young [RY82], for spheres in contact with planes or other spheres, the form is given by

\[
F(g) = k g^{3/2}
\] (4.87)

The shape of the hysteresis loop occurring during quasi-static loading is predicted quite well by this model. The important aspect is the energy absorption properties of the
Chapter 4. Internal Friction Contact Model

internal friction model under low-speed (quasi-static) loading, which is not observed with traditional spring-damper models. Furthermore, the internal friction model creates a cusp in the force/displacement curve at the transition between compression and restitution, which is not observed with the spring/damper model. For haptic displays, a spring-damper contact model may feel “soft” for low-speed impacts, where the above model stiffens the system for compression, and decreases the stiffness for restitution, making the over-all system feel much “harder” (a proposed implementation is given in D). It should be remembered, however, that although there is experimental justification for this model for crystalline solids under low-speed impact [MSB85], models which include viscoelastic terms may be more appropriate for high speed impacts, and for rubber-like materials.

It is worth noting that the form of $F(g)$ must be energy conservative in the internal friction model. We must, therefore, avoid velocity-dependent terms. In the remainder of the thesis, we will assume the simpler Hooke’s Law form, since it has the exact profile of a spring/damper model with the same stiffness $k$ under quasi-static loading. Figure 4.12 shows the relation between the internal friction (hysteretic) and spring damper (non-hysteretic) models under quasi-static loading.

4.7 Determining the Effective Stiffness $k$: Measuring Impact Duration

We can use the contact duration as one experimental method to determine the effective $k$. Whether a spring damper or internal friction model is used, we need only find that value of $k$ which equates simulated and measured contact duration.

The contact duration can be simply measured for many geometries and materials, as shown in Figure 4.13. If conductive material is placed on the impact point of the body surfaces, then an oscilloscope can be used to measure the contact duration (in the
Figure 4.12: Quasi-static Loading for Internal Friction and Spring/Damper Based Models

Figure 4.13: Contact Duration Experiment
Chapter 4. Internal Friction Contact Model

The simplest experiment, two metal spheres can be used as shown. The contact duration for metal spheres was found to be on the order of one microsecond.

For a spring, the contact duration would be one half the period of a harmonic oscillator (we can also use this as an approximate duration for a lightly damped spring/damper), given by

$$\delta t_{\text{impact}} = \pi \sqrt{\frac{m}{k}}$$  \hspace{1cm} (4.88)

Determining $k$ is, therefore, confounded by the object's mass. Furthermore, multiple simultaneous contact points will also affect the contact duration. We will make the assumption that $k$ can be determined as an invariant material property from a single point impact, as given by equation 4.88. To get an order of magnitude estimate, we consider a unit mass with impact duration of one microsecond, leading to an effective stiffness of $k \sim 10^{12}$.

4.8 Special Case Example: Bar Landing on a Planar Surface

The general planar case is simplified by considering a bar landing under gravity onto a horizontal plane. The ends of the bar are vertices which are treated as the points penetrating the flat surface. The internal friction model was compared to a spring-damper model with an equivalent stiffness $(k)$, and damping $(b)$ computed from the relations (see [SV94])

$$\rho = \frac{b}{2\sqrt{k}}$$  \hspace{1cm} (4.89)

$$r^2 = \exp\left(-\frac{2\pi \rho}{\sqrt{1-\rho^2}}\right)$$  \hspace{1cm} (4.90)

The stiff equations were integrated using the CVODE\textsuperscript{1} software package. Two cases are particularly revealing:

\textsuperscript{1}CVODE is a freeware integration package produced by Scott D. Cohen and Alan C. Hindmarsh, Lawrence Livermore National Lab
• dropping the bar in a vertical orientation, such that only a single vertex participates in the contact

• dropping the bar in a horizontal orientation, such that no torque is applied to the bar during contact

The algorithms presented here will be strictly for the special cases considered; however, an algorithm for handling the general case of multiple contact points is proposed in Appendix C.

4.8.1 Internal Friction Algorithm

With reference to the notation of section 4.2.2, we can easily identify the equilibrium force at each vertex in this simple example to be $f_{\text{ext}}/n_v = f_{\text{gravity}}/n_v$, if $n_v$ is the number of vertices participating in the contact. For a vertical bar, $n_v = 1$, and for the horizontal bar, $n_v = 2$. We can easily compute the penetration range which can produce sufficient force to support the bar in equilibrium, as shown in Figure 4.14. We therefore assume an equilibrium transition when

$$g_{\text{min}} = \left| \frac{f_{\text{ext}}/n_v}{(1 + f)} \right| \leq |g| \leq g_{\text{max}} = \left| \frac{f_{\text{ext}}/n_v}{(1 - f)} \right|$$  \hspace{1cm} (4.91)

If the number of vertices is $n_v$, the algorithm is as shown in Figure 4.15.

4.8.2 Dropping the Bar in a Vertical Orientation

This reduces to the special case of a point mass falling onto the plane. Since only a single vertex participates in the contact, we can take $n_v = 1$. 
Chapter 4. Internal Friction Contact Model

Figure 4.14: Equilibrium Transition for Bar on Planar Surface

\[
\begin{align*}
&\text{for } i = 0 \text{ to } n_v - 1 \text{ do} \\
&\quad \text{if } g_i < 0 \text{ then} \\
&\quad \quad \text{equilibrium} = \text{False} \\
&\quad \quad \text{if } |\frac{dg}{dt}| < \text{Velocity Tolerance and } g_{\text{min}} \leq |g| \leq g_{\text{max}} \text{ then} \\
&\quad \quad \quad \text{equilibrium} = \text{True} \\
&\quad \quad \text{else if } \frac{dg}{dt} < 0 \text{ then} \\
&\quad \quad \quad \quad \lambda_i = (1 + f)F(g) \\
&\quad \quad \text{else} \\
&\quad \quad \quad \lambda_i = (1 - f)F(g) \\
&\quad \quad \text{end if} \\
&\quad \text{else} \\
&\quad \quad \lambda_i = 0 \\
&\quad \text{end if} \\
&\text{end for}
\end{align*}
\]

Figure 4.15: Internal Friction Algorithm for Planar Bar
Figure 4.16: Straight Vertical Bar Landing on a Flat Surface. The curves for the Spring/Damper and Internal Friction models overlap. (IF = Internal Friction model, SD = Spring/Damper model). Note, all units SI (mks).
Implementation Results With $F(g) = k g$

Figure 4.16 shows the bouncing of the point mass on a flat surface. Figure 4.16 clearly shows that the spring/damper and internal friction models have nearly identical performance at the macroscopic level.

Figure 4.17 shows a single bounce, with an integration time step of $\sim 1 \times 10^{-6}$ seconds. The position versus time plots for one impact with internal friction and a spring damper are virtually overlapping, indicating that the models predict the contact event equally well.

4.8.3 Dropping the Bar in a Horizontal Orientation

In this case, the bar remains horizontal, and both vertices have identical forces applied to them at all times. Since both vertices participate in the contact, we take $n_v = 2$.

Preserving the Coefficient of Restitution for Multiple Contacts with a Spring/Damper Model

We now consider the effect of multiple contacts on the coefficient of restitution. As shown in [SV94], the coefficient of restitution for a point unit mass attached to a spring/damper can be expressed as

$$r^2(b, k) = \exp \left( -\pi \frac{b}{\sqrt{k}} \right).$$

(4.92)

Goyal, et. al. [GPS94b] produced a similar result, and also derive an expression for the effective mass for off-center collisions. However, they do not consider the effects of multiple simultaneous contacts, and state that
Figure 4.17: Straight Vertical Bar Landing on a Flat Surface - Single Bounce (IF = Internal Friction model, SD = Spring/Damper model). Note, all units SI (mks).
..., a simple expression for energy loss at impact in terms of spring and damper constants is usually not possible.

We will now consider the case of a bar with unit mass falling horizontally onto a plane. Since the forces from springs and dampers acting in parallel are additive, it follows that for \( n \) spring/dampers acting in parallel the coefficient of restitution is given by

\[
\rho^2(nb, nk) = \exp \left( -\frac{\pi \sqrt{n} \frac{b}{\sqrt{k}}}{\sqrt{1 - \frac{nb^2}{4k}}} \right). \tag{4.93}
\]

Therefore,

\[
r(nb, nk) < r((n-1)b, (n-1)k), \quad n > 2 \tag{4.94}
\]

and in particular

\[
r(n^* b, n^* k) \sim 0 \tag{4.95}
\]

for the correct choice of \( n^* \). This is not a desirable property for simulation purposes, and straightforward application of spring/dampers to the vertices results in coefficients of restitution which are too low (see Figure 4.19 and Figure 4.18). For the simulation shown in Figure 4.18, the coefficient of restitution (with \( n = 2 \) for the two-vertex horizontal bar landing on a flat surface) is computed to be \( r = 0.729 \), which is in exact agreement with the simulated result. However, if the bar were oriented vertically when dropped, so that only a single vertex participated in the contact (which is identical to the point contact case discussed previously), then the coefficient of restitution is \( r = 0.8 \).

Although it is trivial to recompute the correct total damping from the total stiffness when vertex constraint forces are in the same direction, it is not obvious how the correction should be made to arbitrary geometries with objects of arbitrary material properties (stiffness \( k \) and coefficient of restitution), as shown in Figure 4.20. The best approach
Figure 4.18: Straight Horizontal Bar Landing on a Flat Surface - Single Bounce (IF = Internal Friction Model, SD = Spring/Damper Model). Note, all units SI (mks).
Chapter 4. Internal Friction Contact Model

Figure 4.19: Straight Horizontal Bar Landing on a Flat Surface (IF = Internal Friction Model; SD = Spring/Damper Model). Note, all units SI (mks).
may be to simply use the effective mass as given by Goyal, et. al. at each contact point, take $n$ as the total number of contact points for that object at that instant, and compute a damping for that point from the equation above.

**Preserving the Coefficient of Restitution for Multiple Contacts with an Internal Friction Model**

Since the internal friction model makes no assumptions about the form of the underlying nominal constraint force $F(g)$ (other than that it is energy conservative), there is no interdependence on the number of vertices. Therefore, each vertex can be modelled in a straightforward manner with an internal friction force, all using the same friction dissipation factor $f$. From the simulation of Figure 4.19, the coefficient of restitution for the internal friction model is $r = 0.8$, regardless of the number of vertices participating in the contact.
Contact Duration

As can be seen from Figures 4.17 and 4.18, the contact duration decreases by a factor of about $\sqrt{2}$ when both vertices participate in the contact. The cause is attributable to a doubling of the effective stiffness when two vertices are in contact compared with a single vertex contact. This effect is considered negligible for practical simulation purposes.

4.8.4 Numerical Stability

Both methods were found to be stable in the implementations presented, and required about 2 seconds of computation time (on a dedicated Sparc 2 workstation) to obtain about 7 seconds of simulation time. The CVODE integrator was asked to report the state each 33 milliseconds. However, the actual time steps taken were adaptively determined, and, during contact, the time steps would have been substantially smaller.

4.8.5 Identifying Equilibrium Transitions

We note that a feature of the internal friction model is its ability to determine the instant that an object enters into an equilibrium, as defined by zero velocity and acceleration. This can be of significance in complicated simulations where numerous objects are potentially in motion, but only some subset is actually in motion at any given time. For example, a parking lot may be full of cars, but only a small subset are in motion at a given time. Objects identified as being in equilibrium can be ignored for integration purposes, until they interact again with an active object in the environment (where interactions can include changes in force fields). Appendix E explores this idea in more detail, but its implementation is left for future work. A major deficiency of the internal friction model, which is shared by "hard" contact models, is that once objects are in a static, equilibrium state, it is non-trivial to solve a possibly statically indeterminate problem to
get them out of that state. Solutions from hard contact methods may be useful in this regard.

4.9 Using the Internal Friction Model For Haptic Display

A form of the internal friction model which may be useable in haptic display was derived and is presented in Appendix D. The motivation for deriving this form was to give the internal friction model $C^0$ continuity at equilibrium. The model was confirmed to work correctly in simulations similar to those already described, but implementation on a haptic device requires better position encoding than possible with the existing "P2 Pantograph". This implementation is left for future work.
This thesis considers the area of accurate contact force simulation and display. In particular, it is noted that there is a general lack of inclusion of texture forces in haptic display algorithms. The thesis describes a method for generating a realistic texture for haptic devices which is simple and can be extended to incorporate actual physical measures of surface properties to produce the appropriate textural feel. This opens the possibility of using online roughness sensing in teleoperation applications, as well as specifying physical parameters in simulation, to obtain the appropriate textural feel in the haptic device. Initial implementations have shown that the method can produce realistic haptic texture.

One contribution of this thesis is in developing a new model for texture force simulation and display based on the surface roughness of the contacting bodies. The observation is made that texture forces can be characterized and generated with statistical, as opposed to deterministic, properties, and that these properties can be transformed between different reference frames. Thus, instead of generating individual texture forces and mapping each force between working frames, statistical properties can instead be mapped, and texture forces generated in a given frame with the mapped properties. This leads to computational efficiency, and the ability to generate high bandwidth texture forces based on the surface roughness of contacting, relatively sliding objects. This methodology is markedly new and different from existing deterministic models of haptic texture.

Another contribution of the thesis is the identification of deficiencies in the use of spring/damper models for accurate contact simulation and developing an alternative
based on the concept of internal friction. A contact model based on the concept of internal friction is derived, which is inspired by surface physics, and with parameters chosen to allow it to be used in impact situations, as well as slow contact. Along the way, a method for general contact simulation using the Euler-Lagrange formulation of constrained dynamics is presented. The new internal friction model is compared to a spring/damper model, and is shown to correct the following two short-comings of the spring/damper model; the spring/damper is energy conservative in quasi-static loading, and requires modification for multiple contact problems, whereas the internal friction model is energy dissipative in quasi-static loading and can be used in a straightforward manner in multiple contact situations. Furthermore, the internal friction model has the feature that it can determine the time that an object experiences an equilibrium transition, which can be useful for simplifying complex simulations. The new internal friction model is a viable alternative to the traditional spring/damper.
Chapter 6

Future Work

The initial implementations of the stochastic haptic texturing algorithm proved the principle that stochastic models could be used to generate convincing texture forces. Simple refinements to the existing model would be to add a realistic stiction generator. In experiments to compare simulated texture to actual very rough sandpaper texture, stiction was found to be an important component of textural feel. Also, modulating the texture force pulse frequency as an increasing function of relative sliding speed could also improve the textural illusion. To fully exploit the model, the input to the surface-space texture generation function should reflect the surface properties of the contacting materials. This is true whether the contact objects exist only in simulation, or if the haptic device is acting as the master for a telemanipulator. In the former case, the surface properties are specified as simulation inputs. In the latter case, the surface properties are measured by some sensor on the manipulator.

The contact constraint force simulation model presented has been verified for simple, planar geometries. Methods were presented to generalize the motion to arbitrary, multiple contact geometries; however, the verification of those methods remains to be done. Furthermore, the internal friction model possesses the interesting feature that it can isolate the instant that an object enters a static, equilibrium state. Methods were proposed to take advantage of this feature for simplifying complicated dynamic systems where only a subset of objects are in motion at a given time; however, implementation of those methods remains to be done. A modified version of the algorithm was presented
for displaying "hard" contact with a haptic device, with verification still to be done on a haptic device with sufficient spatial encoder resolution. Although the model is presented as an alternative to the spring/damper for soft contact simulation, some surface physics literature suggests that the two model-types should co-exist, and adapting the internal friction model to use available parameters from the surface physics literature could prove useful.
Bibliography


Bibliography


Appendix A

Pantograph Dynamics

A.1 Determining Mass Center for Links

The link is supported such that the constraining forces are vertical (see figure A.21). The constraining force $f_1$ can be measured with a scale. The mass center for any link can then be determined through static force balance, using the equations:

$$L = L_1 + L_2, \quad L_2 = \frac{f_1 L}{mg} \quad (A.96)$$

A.2 Link Inertia

Because there are no complicated drive components contributing to the system inertia, the individual link inertias could be measured while the device was disassembled. The motors were rigidly fixed to the base links and, therefore, their inertia could be added to the corresponding base link inertia. Armstrong et al [AKB86] describe a method of measuring link inertias by suspending the link about its mass centre, and measuring the oscillation period of the suspended link about the mass centre. The major deficiencies of this method are an inability to handle links with non-parallel faces and include drive components. Here we use a different method based on measuring the period of a revolute link as it rotates about its joint axis (we can thus easily include the inertia of drive components).

The inertia is measured by considering the link as a simple pendulum, rotating about the joint axis. The governing equation for a link swinging in a conservative gravitational field is the elliptic differential equation:

$$I \ddot{q} + mg l_c \sin(q) - \tau_f - \tau_a = 0 \quad (A.97)$$
where

\[ I = \text{inertia about the rotation axis} \]
\[ q = \text{angle of the link relative to vertical} \]
\[ m = \text{total link mass} \]
\[ l_c = \text{perpendicular distance from the fulcrum to the center of mass} \]
\[ \tau_f = \text{friction torque about fulcrum} \]
\[ \tau_a = \text{torque due to air resistance} \]

Once the link inertia about the fulcrum is known, it can be projected onto any other axis of the body. To obtain \( I \), it is necessary to measure the oscillation period, then choose \( I \) such that the period predicted by the above equation matches the measured period. There are two sources of error in the procedure:

1) accounting for energy losses

2) measurement error in obtaining the period.
A.2.1 Energy Losses

Friction

A friction torque acts on the pendulum link, due to relative sliding at the fulcrum. This torque can be approximated as the effective friction force acting with a lever arm equal to the distance from the center of the fulcrum to the sliding surface. The friction torque is given by

\[ \tau_f(q) = f_f l_f \]  
\[ f_f = \mu f_n \]  
\[ f_n = \sqrt{f_{nx}^2 + f_{ny}^2} \]  
\[ f_{nx} = f_c \sin(q) \]  
\[ f_{ny} = mg + f_c \cos(q) \]  
\[ f_c = m \dot{q}^2 l_c \]

where

- \( l_f \) = lever arm from fulcrum to sliding surface
- \( f_n \) = normal force
- \( \mu \) = coefficient of friction
- \( f_c \) = centripetal force from pendulum swing
- \( l_c \) = distance from fulcrum to center of mass

In most practical cases, the friction torque will be small due to a small coefficient of friction (about 0.1 for lubricated surfaces), and small lever arms between the fulcrum and sliding surface.

Air Resistance

As the pendulum swings, air resistance will act on the face perpendicular to the angular velocity. When the faces are approximately parallel to a line from the fulcrum to the mass center, this torque is given by:

\[ d\tau_a(l) = f_a l \]  
\[ f_a = c_a s^2(l) w(l) \, dl \]  
\[ s(l) = l \dot{q} \]
where

\[ d\tau_a(l) = \text{infinitesimal torque at distance } l \]

\[ f_a = \text{air resistance force} \]

\[ c_a = \text{coefficient of aerodynamic drag} \] (A.110)

\[ w(l) = \text{width of face at distance } l \]

\[ w(l) \, dl = \text{area of face at distance } l \]

\[ s(l) = \text{speed at distance } l \text{ from the fulcrum} \]

and

\[ \tau_a = \int d\tau_a = \int c_a \, w(l) \, q^2 \, t^3 \, dl \] (A.111)

If \( w(l) \) is approximately constant, then \( \tau_a \) is given by

\[ \tau_a = \frac{1}{4} c_a \, w \, q^2 \, t_e^4 \] (A.112)

where

\[ t_e = \text{effective pendulum length} \] (A.113)

This torque will be small since \( c_a \) is small, and in particular when the pendulum is short \((l << 1)\).

### A.2.2 Error Analysis

In simulations with reasonable values for energy losses, there was no appreciable variation in the simulation period with different initial release angles. Furthermore, a simulation of ten consecutive swings showed that while the peak angle can decrease by a factor of about 0.08, the period of each swing changes by less than a factor of about 0.005. This indicates that the best timing method is to take multiple swings, and average the results to spread the measurement error in the final period over all periods. Any numerical integration has a globally increasing error, even if local errors can be bounded. To minimize this global integration error, the integration is performed only to the first quarter period, where the theoretical position of the pendulum is exactly vertical. This provides the shortest total integration time to a well established reference, and therefore, has the lowest possible global integration error. This technique also minimizes the effects of measurement error since the absolute error in the period is spread evenly over the quarter periods, as shown below.

We can investigate the error of the method using the loss-free, small oscillation equation for the period of a pendulum, assuming the form of the solution is applicable to larger, lossy oscillations. The period is given by
Appendix A. Pantograph Dynamics

\[ T = 2\pi \sqrt{\frac{I}{mg \, l_c}} \]  \hspace{1cm} (A.114)

Let \( e \) be the absolute error in measuring the period

\[ e = T_m - T_a \]  \hspace{1cm} (A.115)

where

\[ T_m = \text{measured period} \]
\[ T_a = \text{actual period} \]  \hspace{1cm} (A.116)

Then, the relative error is given by

\[ \frac{I_m}{I_a} = 1 + 2 \frac{e}{T_a} + \left( \frac{e}{T_a} \right)^2 \]  \hspace{1cm} (A.117)

where

\[ I_a = \text{actual inertia} \]
\[ I_m = \text{inertia calculated from the measured period} \]  \hspace{1cm} (A.118)

This indicates that the period \( T_a \) should be maximized; the initial release angle for the pendulum should, therefore, be maximized. Typical values for the variables are

\[ e = 0.1 \text{ seconds} \]  \hspace{1cm} (A.119)
\[ T_a = 1 \text{ second} \]  \hspace{1cm} (A.120)

If only a single period is used, then the relative error in the inertia would be about 0.1. However, by averaging \( n \) oscillations, the relative error becomes

\[ \frac{I_a}{I_m} = 1 + \frac{e}{n} \left( \frac{1}{T_a} \right) + \frac{e^2}{n^2} \left( \frac{1}{T_a} \right)^2 \]  \hspace{1cm} (A.121)

Because the first quarter period is used as the reference, the averaging is further increased, and the relative error is given by

\[ \frac{I_a}{I_m} = 1 + \frac{e}{4n} \left( \frac{1}{T_a} \right) + \frac{e^2}{16n^2} \left( \frac{1}{T_a} \right)^2 \]  \hspace{1cm} (A.122)

With \( n = 5 \), relative error is reduced to about 0.005, and \( n = 10 \) gives a relative error of approximately 0.0025.
Figure A.22: Pendulum Inertia versus Oscillation Period
A.2.3 Algorithm

As shown in figure A.22, the inertia is an increasing function of period. After the oscillation period $T_m$ is measured, the algorithm first calculates two guesses for the corresponding inertia. One initial guess ($I_1$) is based on the small oscillation approximation (by substituting $T_m$ into the small oscillation equation), and its actual period is guaranteed to be higher than $T_m$ for any release angle greater than zero. Using $I_1$, the pendulum equation is integrated to obtain $T_1$, the true period associated with $I_1$. In simulations, the variation of the actual period was never seen to greater than about 0.2 of the small oscillation approximation, so a good second guess for the inertia ($I_2$) is $0.5I_1$. The pendulum equation is again integrated to obtain the period $T_2$. These initial guesses will generally bound the true inertia, although the algorithm will converge even if this were not the case. A linear interpolation is then used to obtain a new guess for the inertia, $I_3$, and the elliptic pendulum equation is integrated to obtain the period corresponding to this inertia, $T_3$. The process is repeated until the difference between the newly calculated period and the measured period is below some threshold. The method was found to converge within five iterations in all tests.

A.3 Handle Mass

Once the link inertias were determined, a simple SDFast \(^1\) model was constructed to determine the effective mass at the handle. If this mass was sufficiently constant for various expected configurations, then its variation could be ignored and an average, constant value used in control. All centripital forces are ignored (there are no Coriollis forces since the linkages are constrained to a plane), since they would be relatively small for the velocities encountered. If the mass is sufficiently small, then it can be ignored for the purposes of control. The effective simulated mass at the center of the workspace was found to vary between 120 to 260 grams. This variation was ignored for the purposes of this thesis work.

---

\(^1\)SDFast is a dynamic system modelling software package produced by Symbolic Dynamics Inc.
Appendix B

Relevant Statistical Concepts

Consider a system defined by

\[ X \in \mathbb{R}^m, Y \in \mathbb{R}^n, A \in \mathbb{R}^{n \times m} \]  \hspace{1cm} (B.123)

\[ Y = AX \]  \hspace{1cm} (B.124)

If the components of \( X \) are samples from a statistical distribution, then the system has the following statistical properties.

B.1 Notation

A scalar variable with Gaussian or normal distribution is denoted

\[ x \sim \text{Normal}(\mu, \sigma) \]  \hspace{1cm} (B.125)

where the mean and variance of the population from which \( x \) is taken are \( \mu \) and \( \sigma \), respectively. For a vector \( X \), each of its components can be taken from a different population, with notation

\[ X \sim \text{Normal}(X, \mu, \sigma^2) \]  \hspace{1cm} (B.126)

The mean and covariance of \( X \) are respectively denoted by

\[
\mu = \begin{bmatrix} E[x_1] \\ E[x_2] \\ \vdots \\ E[x_m] \end{bmatrix} = \begin{bmatrix} x_\mu_1 \\ x_\mu_2 \\ \vdots \\ x_\mu_m \end{bmatrix}
\]  \hspace{1cm} (B.127)

and

\[
\sigma^2 = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1} & \sigma_{m2} & \cdots & \sigma_{mm} \end{bmatrix}
\]  \hspace{1cm} (B.128)
where $E$ is the statistical expectation, and $\sigma_{ij}$ is the covariance between the $i$'th and $j$'th components. Upper-case leading super-scripts indicate a vector and refer to vector statistical properties, while lower-case leading super-scripts are scalars and refer to the statistical properties of individual vector components.

**B.2 Covariance**

If the covariance matrix for $X$ is $X\sigma^2$ then the covariance matrix for $Y$ is given by

$$Y\sigma^2 = A(X\sigma^2)A^T \quad (B.129)$$

$$\text{cov}(x, ax + by) = a \text{cov}(x, x) + b \text{cov}(x, y) \quad (B.130)$$

**B.3 Expectation (Mean)**

If the mean of $X$ is given as $X\mu$ then the mean of $Y$ is given by

$$Y\mu = A(X\mu) \quad (B.131)$$

**B.4 Generating Random Deviates with Given Statistical Properties**

Some algorithms are now presented for generating random deviates with given statistical properties, as described by their mean and covariance.

**B.4.1 Generating Gaussian Deviates**

**Floating Point**

Numerous algorithms exist and many are presented in [PTVF92] and [Knu81], to which the reader is referred.

**Fixed Point**

If it is computationally impossible to generate floating point normal distributions (as is often the case if the texture is to be added at a very low-level control layer) the following simple, widely known algorithm can be utilized to provide fixed point normal distributions (see [Knu81] for other methods).

Let $x \sim \text{Uniform}(\mu, \sigma)$, $X$ a sample from $x$, and $n$ the number of samples of $X$. The sample mean $E[X]$ is normally distributed with mean and standard deviation given as simple functions of the underlying uniform distribution of $x$. Since $x$ can be easily generated by a technique such as linear congruence sequences [PTVF92], it is possible to generate Gaussian deviates of desired mean and standard deviation by generating 30 or more pseudo-random numbers per deviate.
Memory Look-Up

Perhaps the fastest method of generating Gaussian deviates is to pre-compute a series of standard normal deviates, then draw randomly from this pool, scaling and translating each deviate to obtain the required mean and standard deviation. Although this method requires only a single pseudo-random number per Gaussian deviate (to determine the memory location), it also requires a large pool of stored values.

B.4.2 Covariant Deviates

To generate deviates with non-zero covariance, consider the following algorithms (another algorithm is presented in Knuth [Knu81]). The choice of method is left to the user, depending on whatever code is readily available.

Method 1

Suppose a vector, \( \mathbf{X} \), is required with components consisting of random numbers which are normally distributed, and the components must be mutually covariant. The mean and covariance for \( \mathbf{X} \) are specified by \( \mu \) and \( \sigma^2 \).

Begin by letting \( x_1 \sim \text{Normal}(\mu_1, \sigma_1^2) \). Let the remainder of the components of \( \mathbf{X} \) be recursively determined by the system

\[
\begin{bmatrix}
  x_2 \\
  x_3 \\
  \vdots \\
  x_m
\end{bmatrix} = L
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_{m-1}
\end{bmatrix} + \mathbf{N}
\]  

(B.132)

\[
L = \begin{bmatrix}
  l_{11} & 0 & 0 & \cdots \\
  l_{21} & l_{22} & 0 & \cdots \\
  \vdots & \vdots & \vdots & \ddots \\
  l_{m1} & l_{m2} & \cdots & l_{m-1m-1}
\end{bmatrix}
\]  

(B.133)

\( L \) is a lower triangular matrix, and \( \mathbf{N} \) is a vector of uncorrelated perturbation terms with

\[
\mathbf{N} \sim \text{Normal}(\mu, \sigma^2)
\]  

(B.134)

where \( \sigma^2 \) is diagonal.

The unknowns to be determined are the components of \( L \), and the mean and variance of each of the \( m - 1 \) terms of \( \mathbf{N} \). Through judicious application of the formulas for
computing the mean and covariance, the following sparse system can be derived for the other unknowns.

Let

\[
\sigma_i = \begin{bmatrix}
x_{\sigma_{i1}} \\
x_{\sigma_{i2}} \\
\vdots \\
x_{\sigma_{i,i-1}} \\
\end{bmatrix}, \quad \tilde{l}_i = \begin{bmatrix}
l_{i1} \\
l_{i2} \\
\vdots \\
l_{i,i} \\
\end{bmatrix}
\]

(B.135)

Then,

\[
X \sigma_{ii} \equiv \begin{bmatrix}
x_{\sigma_{11}} & x_{\sigma_{1,2}} & \cdots & x_{\sigma_{1,i}} \\
x_{\sigma_{21}} & \ddots & \cdots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
x_{\sigma_{i,1}} & x_{\sigma_{i,2}} & \cdots & x_{\sigma_{ii}} \\
\end{bmatrix}
\]

(B.136)

\[
I_{i,i} \in \mathbb{R}^{i \times i} = \text{Identity}
\]

Then,

\[
\begin{bmatrix}
\sigma_2 \\
\sigma_3 \\
\vdots \\
\sigma_m \\
\vdots \\
x_{\sigma_{22}} \\
x_{\sigma_{33}} \\
\vdots \\
x_{\sigma_{mm}} \\
\end{bmatrix} \quad \begin{bmatrix}
x_{\sigma_{11}} & 0 & 0 & 0 & 0 \\
0 & x_{\sigma_{2,2}} & 0 & 0 & 0 \\
0 & 0 & \ddots & 0 & 0 \\
0 & 0 & 0 & x_{\sigma_{m-1,m-1}} & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
x_{\sigma_{22}} & \sigma_2^T & 0 & 0 & 0 \\
x_{\sigma_{33}} & 0 & \sigma_3^T & 0 & 0 \\
\vdots & 0 & \vdots & 0 & 0 \\
x_{\sigma_{mm}} & 0 & 0 & \sigma_m^T & 0 \\
\end{bmatrix} \begin{bmatrix}
l_1 \\
l_2 \\
\vdots \\
\tilde{l}_{m-1} \\
\end{bmatrix} = \begin{bmatrix}
l_{i1} \\
l_{i2} \\
\vdots \\
l_{i,i} \\
\end{bmatrix}
\]

Note that the \(l\)'s are not dependent on the \(^n\sigma_{ii}\)'s; therefore the upper block diagonal matrix can be solved for the \(l\)'s first, then the sparse lower matrix can be used to trivially solve for the \(^n\sigma_{ii}\)'s.
The components for the mean $N\mu$ are given by:

$$
\begin{bmatrix}
\mu_1 \\
\mu_2 \\
\vdots \\
\mu_{m-1,m-1}
\end{bmatrix}
= 
\begin{bmatrix}
\mu_2 \\
\mu_3 \\
\vdots \\
\mu_{m,m}
\end{bmatrix}
- L
\begin{bmatrix}
\mu_1 \\
\mu_2 \\
\vdots \\
\mu_{m-1,m-1}
\end{bmatrix}
$$

(B.138)

As an example, when $m = 3$, the following system results:

$$
\begin{bmatrix}
\sigma_{12} \\
\sigma_{31} \\
\sigma_{32} \\
\sigma_{33}
\end{bmatrix}
= 
\begin{bmatrix}
\sigma_{11} & 0 & 0 & 0 \\
0 & \sigma_{11} & \sigma_{12} & 0 \\
0 & \sigma_{21} & \sigma_{22} & 0 \\
0 & \sigma_{31} & \sigma_{32} & 1
\end{bmatrix}
\begin{bmatrix}
l_1 \\
l_2 \\
l_3 \\
n\sigma_1
\end{bmatrix}
$$

(B.139)

$$
\begin{bmatrix}
\mu_1 \\
\mu_2 \\
\mu_3
\end{bmatrix}
= 
\begin{bmatrix}
\mu_2 \\
\mu_3 \\
\mu_4
\end{bmatrix}
- L
\begin{bmatrix}
\mu_1 \\
\mu_2 \\
\mu_3
\end{bmatrix}
$$

(B.140)

Method 2

An alternative is to consider the orthogonal eigenvalue decomposition of the symmetric matrix

$$
X\sigma^2 = LDL^T
$$

(B.141)

where $D$ is a diagonal matrix of the eigenvalues of $X\sigma^2$. $X\sigma^2$ is a legitimate covariance matrix if its eigenvalues (the elements of $D$) are real and positive. The covariant deviates are then generated with

$$
X = LN
$$

(B.142)

$$
N\sigma^2 = D
$$

(B.143)

$$
N\mu = L^T(X\mu)
$$

(B.144)
Appendix C

Generalization to Multiple Contacts

An algorithm for solving the multiple contact problem is now proposed. The internal friction model is characterized by a "hard" contact in equilibrium. In other words, no change in constraint violation (penetration depth) is allowed for some range of contact constraint forces. To implement this effect, we can use the classical dynamic constraint enforcement algorithm; by differentiating the constraints twice with respect to time, the constraint forces can be solved explicitly in terms of the current state. These constraint forces are added to the existing constraint forces (from the internal friction model). If the net constraint force is within the internal friction equilibrium bound, then the net force is applied. If the net constraint force is outside the equilibrium bound, then only the closest equilibrium bound force is applied, and the system is integrated forward with this constraint force.

Starting with the constraint \( g(q) = g_{eq} \), differentiating twice with respect to time leads to:

\[
\begin{align*}
G \dot{q} &= 0 \\
\dot{G} \dot{q} + G \ddot{q} &= 0
\end{align*}
\]

where

\[
G = \frac{\partial g}{\partial q}
\]

\[
G_T = \left[ \{ v_{ij}; i = 0, \ldots, n_v - 1, j = 0, \ldots, n_o - 1 \} \right]
\]

\[
\dot{G}_T = \left[ \{ \dot{v}_{ij}; i = 0, \ldots, n_v - 1, j = 0, \ldots, n_o - 1 \} \right]
\]

84
Appendix C. Generalization to Multiple Contacts

Note that if \( j \) represents a fixed obstacle, then \( \dot{\mathbf{w}}_j = 0 \).

The system is described by

\[
\dot{\mathbf{q}} = \mathbf{v} \quad \text{(C.153)}
\]

\[
M(q) \dot{\mathbf{v}} = f(q, \mathbf{v}) - G^T \lambda \quad \text{(C.154)}
\]

\[
g(q) \geq g_{eq}(q_{eq}) \quad \text{where} \quad g_{eq}(q_{eq}) < 0 \quad \text{(C.155)}
\]

Let

\[
\lambda^* = (G M^{-1} G^T)^{-1}(G M^{-1} \dot{f} + \dot{h}) \quad \text{(C.156)}
\]

\[
h = \frac{d^2}{dq^2}g(q)(v, v) = \dot{G}q \quad \text{(C.157)}
\]

Then, the components of \( \lambda \) are chosen according to

\[
\lambda_i = \begin{cases}
\lambda_{i(j)} & \text{if} \quad \dot{g}_i \neq 0 \\
F_c(g_i) + \lambda_i^* & \text{if} \quad F_c(g_i) \leq F_c(g_i) + \lambda_i^* \leq F_r(g_i) \\
F_r(g_i) + \lambda_i^* & \text{if} \quad F_c(g_i) \leq F_r(g_i) + \lambda_i^* \leq F_r(g_i)
\end{cases} \quad \text{(C.158)}
\]

where

\( \lambda = \) total constraint forces

\( \lambda_{i(j)} = \) constraint forces computed from the internal friction model to enforce \( g(q) = 0 \)

\( \lambda^* = \) constraint forces computed to enforce \( g(q) = g_{eq}(q_{eq}) < 0 \)

\( g_{eq}(q_{eq}) = \) the equilibrium penetration

\( F_c(g_i) = \) internal friction compression force for the given \( g_i \)

\( F_r(g_i) = \) internal friction restitution force for the given \( g_i \)

The states can now be integrated forward, with the constraint forces effectively bounded by the internal friction model. When all constraint forces are within the internal friction bounds, we place the object in equilibrium, and set all of its derivatives to zero (this acts as a stabilizing step which eliminates the discontinuous nature of the internal friction model).
Appendix D

Internal Friction Model for Haptic Devices

We now propose a form of the internal friction model which may be useful for haptic display of hard contact. In order to make the internal friction model useful for haptic display, we must eliminate the discontinuity at equilibrium. We will now slightly modify the transition between compression and restitution, as shown in figure D.23.

By tilting the line segment between the compression and restitution phases, the constraint force becomes a monotonic function of the penetration $g$. With this modification, the internal friction model provides a localized solution, which is continuous with respect to position. Provided that the haptic device can adequately spatially resolve the distance $e$, this model can be used for position force feedback.

We must now re-analyze the required coefficient of internal friction $f$ during a compression/restitution cycle. The energy of compression is given by

$$E_c = \int_0^{g_{\text{max}}} (1 + f) F(g) \, dg = (1 + f) a \quad (D.159)$$

$$a = \int_0^{g_{\text{max}}} F(g) \, dg \quad (D.160)$$

We have no a priori knowledge of $g_{\text{max}}$, so we must produce an approximate solution (the approximations made, in fact, lead to an exact solution if $F(g) = kg$). We begin by approximating the restitution energy by

$$E_r = \int_0^{g_{\text{max}}} (1 - f) F(g) \, dg + e/2 \left[(1 + f)F(g_{\text{max}}) - (1 - f)F(g_{\text{max}})\right] \quad (D.161)$$

$$= (1 - f) a + e f F(g_{\text{max}}) \quad (D.162)$$

where $e$ is an arbitrary distance, as shown in D.23. But

$$F_{\text{average}} \equiv \frac{1}{g_{\text{max}}} \int_0^{g_{\text{max}}} F(g) \, dg \simeq \frac{1}{2} F(g_{\text{max}}) \quad (D.163)$$

$$\Rightarrow F(g_{\text{max}}) \simeq \frac{2}{g_{\text{max}}} a \quad (D.164)$$

Therefore,

$$E_r = (1 - f) a + e f \frac{2}{g_{\text{max}}} a \quad (D.165)$$

$$= (1 - f) a + 2 f \frac{e}{g_{\text{max}}} a \quad (D.166)$$

86
Appendix D. Internal Friction Model for Haptic Devices

Figure D.23: Internal Friction Model with $C^0$ Continuity

Now, let

$$e = h g_{\text{max}}$$ \hspace{1cm} (D.167)

Then

$$E_r = a [(1 - f) + 2hf]$$ \hspace{1cm} (D.168)

and

$$r^2 = \frac{E_r}{E_c} = \frac{a [(1 - f) + 2hf]}{a (1 + f)} = \frac{1 + (2h - 1)f}{1 + f}$$ \hspace{1cm} (D.169)

And, finally,

$$f = \frac{1 - r^2}{r^2 - (2h - 1)}$$ \hspace{1cm} (D.170)

Note that as $h \to 0$, the results agree with the previous values of $r^2$ and $f$. At the beginning of the simulation, we need only specify the value of $h$, which determines the tilt of the line segment connecting the compression and restitution phases. We would generally choose $h \ll 1/2$ and such that $F(g)$ is monotonic to numerical precision and the spatial resolution of the haptic device.
Appendix E

Using Static Equilibrium in Simulation

E.1 Newton's Cradle

Consider Newton's Cradle, which consists of a series of contacting pendulum, and can be used to verify the efficacy of contact simulations. A very coarse analysis of the impact is now presented. As given in [CRC91], the approximate phonon speed in steel is 6000 m/s. If the contact duration is on the order of 1 \( \mu s \), then in order for the contact to terminate before the adjoining objects are "notified", the diameter of the colliding objects must be on the order of 6 mm or larger; otherwise, the effects from collision with the adjoining object are felt during the initial impact. This is a very conservative estimate, assuming that the entire energy and momentum of the original impact is propagated in a single wave. As a consequence, any colliding objects whose dimensions are greater than 6 mm should be assumed to undergo force propagation similar to Newton's Cradle. The significance will be in determining which objects to include in the simulation at any given time.

E.2 Real-Time versus Quasi-Real-Time

E.2.1 Real-Time

A real time algorithm is defined here as any algorithm capable of reliably producing a given system output at a sufficiently high rate to make the algorithm output indiscernible from the real system to an external observer. In other words, anyone watching a simulation on a monitor should not be able to distinguish the simulation output from a video image of the actual system, where both the simulated and real systems are started at the same instant with identical initial conditions. Clearly, the algorithm's computational cost must be bounded, and the implementation details (software and hardware) must allow the upper bound computations to be performed. In other words, integrating the system from one state to the next must take no more time than the size of the integration step. Furthermore, the integration step size must be small enough to allow the system to be perceived as continuous, and the error controlled to be imperceptible (both very subjective).

E.2.2 Quasi-Real-Time

The conditions above can be relaxed in various ways to create a quasi-real-time system. Often, the boundedness condition is relaxed, and the system is deemed to be real-time
even if certain scene complexity levels cause an external observer to perceive differences between real and simulated systems (usually in the form of delays in the update of the scene geometry).

E.3 Equilibrium List

E.3.1 Definition

Many (perhaps most) typical dynamic scenes are a combination of moving and stationary objects. If a constraint-based approach is considered, then it remains to determine the system constraint multipliers at each integration time instant, which can be costly. If the cost of computing the multipliers can make the constraint-based approach non-real-time, then a quasi-real-time algorithm may still be possible. Consider the following observations. Once an object enters equilibrium (zero velocity and acceleration), it remains in equilibrium until

- it undergoes an interactive collision with another object.
- an object with which it shared constraint forces upon entering equilibrium leaves equilibrium.
- it never leaves equilibrium - it is a rigid, fixed obstacle.

For each object, an equilibrium list is defined which contains all the objects with which constraint forces are shared upon entering equilibrium. For example, consider three objects $A, B, C$. The coefficient of friction between $C$ and $B$ is higher than between $A$ and $C$, and $A$ and $B$ initially sit on $C$ as shown in Figure E.25. Since we assume that all objects are at rest, initially the system has no entries. At time $t=0$, $C$ is re-inserted into the system and begins to rotate. All items in $C$'s equilibrium list are re-inserted into the system. Since both $A$ and $B$ must have shared constraints with $C$ when entering equilibrium, both $A$ and $B$ are re-inserted into the system, bringing the total system size to three objects. When $C$ reaches the angle of repose for $A$, $A$ begins to slide, and at time $t=1$, $C$ stops rotating and again re-enters equilibrium. Since $A$ must share a constraint with $C$ as it slides down $C$'s surface, $C$ is repeatedly put back into the system by $A$'s motion. When $C$ is re-inserted, it checks its equilibrium list and re-inserts $B$. Thus, although both $C$ and $B$ are stationary they are maintained in the system, and any constraints between $C$ and $B$ can thus be propagated. This is essential since, although $C$ is in equilibrium, its constraint forces are changing as the weight of $A$ shifts on its surface. If $B$ is contributing to those constraint forces, as it must through at least its own weight, then the effect must be propagated. At time $t=2$, $A$ collides with $B$, and comes to an equilibrium while sharing constraint forces with both $B$ and $C$. All objects are then in equilibrium, and the system size is again zero. If at time $t=3$ object $B$ is suddenly removed, then all items on its equilibrium list are re-inserted and the system size is again three objects.
E.3.2 Fixed Obstacles

Since object \( C \) itself must be connected to something, we must define the concept of a rigid obstacle - one which never leaves equilibrium. We can thus break constraint propagation between sub-systems. The rigid obstacle is trivial to implement since we assume that its rigid constraints are unbreakable by any interaction with moving objects in the scenario. We simply maintain its equilibrium list as null so that it never propagates a constraint force when it undergoes an interaction with any moving object, and we never enter it into the dynamic system. It therefore only acts to define a reference volume into which no moving object is allowed to enter. In the above example, object \( G \) is a rigid-obstacle, and although object \( D \) is connected to \( C \) through \( G \), the motion of \( C \) does not cause \( D \) to be re-inserted, since the equilibrium list of a rigid obstacle is always null. Therefore, object \( D \) always remains in equilibrium, its constraint forces are never re-computed, and its motion is never integrated.
Figure E.25: System Scenario