Optimization-based Analysis of Rigid Mechanical Systems with Unilateral Contact and Kinetic Friction

Miguel J. Gomez

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2008

Program Authorized to Offer Degree:
Applied Mathematics

ProQuest LLC
789 E. Eisenhower Parkway
PO Box 1346
Ann Arbor, MI 48106-1346

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Miguel J. Gomez

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

Anne Greenbaum
Anne Greenbaum

Reading Committee:

James V. Burke

Brian C. Fabien

Anne Greenbaum
Anne Greenbaum

Date: 3/18/2008

In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of the dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to ProQuest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature Miguel Gomez

Date 3/18/2008

University of Washington

Abstract

Optimization-based Analysis
of
Rigid Mechanical Systems
with
Unilateral Contact and Kinetic Friction

Miguel J. Gomez

Chair of the Supervisory Committee:
Professor Anne Greenbaum
Department of Mathematics

A widely-accepted technique for the analysis of rigid mechanical systems with unilateral contact and Coulomb friction is to formulate the contact forces (or impulses) as the unknowns of a *linear complementarity problem* (LCP). However, the solution set of an arbitrary LCP may be nonconvex, which poses the following computational and theoretical problems: if an arbitrary solution of the LCP is chosen, the computed force distributions over continuous contact regions may change discontinuously as a function of time. This could lead to erroneous analysis of continuum properties, such as heat or stress. On the other hand, if a minimum-norm solution is sought, the solution set must be fully characterized, which could be computationally prohibitive for complex problems.

During this course of research, we have used Gauss' Principle of Least Constraint to formulate the normal forces of the kinetic (sliding) problem as the minimum-Euclidean norm solution of a *convex quadratic program*. When the

feasible region is nonempty, the solution set is *convex*, regardless of the friction coefficients, which makes it computationally efficient to compute the minimum Euclidean norm solution. To illustrate this point, Painleve's example is characterized and compared with the corresponding LCP solution. Based on these results, we suspect that solution multiplicity in frictional contact problems has no physical basis and that this phenomenon is simply a deficiency of the LCP formulation.

We also demonstrate how our formulation can be used in conjunction with explicit Runge-Kutta differential-algebraic solvers for these types of systems. To experimentally validate this approach, a 5^{th}-order implementation has been used to accurately predict the motion of an unlubricated pendulum. By matching individual trajectories, friction coefficients were calibrated for four different material interface types: brass-aluminum, steel-aluminum, Teflon-aluminum, and nylon-aluminum. In order to accurately match trajectories, we have found that *two* friction coefficients are necessary: a constant Coulomb term and a viscous damping term. We also used this technique to examine the distribution of the normal load as a function of time, and the results are consistent with a qualitative force-balance analysis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank all the members of my committee for the insight, guidance, and patience they have given me during the course of this research. I would also like to thank my wife for her love and unwavering support over the past 8 years.

Introduction

In order to evaluate key behaviors of a multi-component mechanical system, such as clearance and operational efficiency, a simulation tool must accurately predict component trajectories and contact forces. We are also interested in the *relative distribution* of the contact forces created at the areas between components so that we can analyze important properties such as wear, stress, and heat dissipation. Even when surfaces are well-lubricated, friction between components play a significant role in determining these quantities.

To achieve efficiency in modeling such systems, we seek friction coefficients which can be confidently applied across a wide range of mechanisms with similar material properties. In other words, a set of friction coefficients calibrated from one experiment should give us reasonably accurate results for any other experiment. If this is the case, then we have some confidence in the physical basis of these parameters, unlike heuristic models which require recalibration of these coefficients for each new system.

The classical Coulomb model for *kinetic* (or *dynamic* or *sliding*) friction:

$$\mathbf{f} = -\mu N \hat{\mathbf{v}}$$

relates the frictional force **f** to the normal load N by a positive scalar called the *kinetic friction coefficient* μ. In this formula, the frictional force acts exactly opposite to $\hat{\mathbf{v}}$, which is the direction of the sliding velocity **v**.

In fact, experiments show that the behavior of friction can be much more complicated than this classical model might imply, especially at low velocities [8], due to various competing physical processes at the molecular level [47]. It is therefore important to choose a friction model appropriate for the system under investigation. Ignoring time- and temperature-dependent behavior, a slight generalization of the Coulomb model is

$$\mathbf{f} = -f\left(\|\mathbf{v}\|\right) N \hat{\mathbf{v}},$$

where f maps the magnitude of $\mathbf{v}$ to a dimensionless scalar value. For many common materials, such as plastics and metals, f is smooth for all $\mathbf{v} \neq 0$ and is independent of the direction of $\mathbf{v}$. The behavior of f at various sliding velocities depends on the material properties of the two contact surfaces.

If N is known a priori, then **f** is trivial to compute. In systems with one or more unilateral contacts, however, determining the value of N at each contact point becomes difficult due to the interdependence of all the forces. This is complicated by the fact that we must have $N_i \geq 0$ at all contacts in order for this formula to remain physically meaningful.

One common technique for simulating these types of systems is through *finite-element modeling* (FEM). This technique discretizes each mechanical component into a set of elements (e.g., tetrahedra), and the deformation of each body is computed explicitly [77]. As the system evolves, the contact normal loads follow from the stress induced by the deformation of the body. Advantages of this approach are that it can be used to compute internal stresses, strains, and heat distribution, and the continuously distributed contact forces are more accurately modeled through discretization of the contact surfaces. The disadvantages of FEM stem from the computational complexity of discretizing each component into finite elements: for an element dimension (e.g., edge length) dx, the number of elements in 3-dimensions is proportional to $1/dx^3$. Since the computational complexity of most sparse linear solvers is $O\left(n^2\right)$, with n being the number of finite elements, the overall complexity is $O\left(\frac{1}{dx^6}\right)$. In other words, doubling the spatial resolution of a simulation increases the computational work by a factor of 64! In addition, nearly rigid systems must be time-stepped at very short intervals through the duration of contact in order to get accurate results. So for systems in which the deformation of components is small, it makes sense to look for a more efficient approximation.

A simplification of this idea is to calculate the motion of the system as if each component were perfectly rigid, but at the same time allow for a small amount of interpenetration between each component. At each time step, the depth of interpenetration at each contact point is estimated, and a restoring force (the *penalty*) is computed as a function of this value based on some heuristic model. The number of points necessary to discretize continuous contact interfaces in 3-dimensions grows in proportion to $\frac{1}{dx^2}$, so doubling the contact resolution only increases the computational complexity by a factor of 16 when continuum properties are not desired. While this scheme is relatively simple to implement and far more efficient than finite-element modeling, it can also lead to a system of stiff ordinary differential equations (ODE's) which must be time-stepped at very short intervals through the duration of contact. More importantly, the parameters of the contact force model may not correspond to any measurable material properties. In effect, the penalty parameters become "magic numbers" which must be calibrated against experimental data for each particular system. This brings into question the value of penalty methods as a predictive tool [14].

A very efficient approach is to treat each component as perfectly rigid and compute contact forces which strictly enforce noninterpenetration. This technique benefits from the simplicity of the rigid body assumption while avoiding the stiff systems that can arise from the finite element or penalty formulations. Since deformation at each contact is not modeled, the time step need not be refined so much when the set of active constraints is not changing. Thus, the rigid-body approximation has clear computational advantages when it can be applied.

Most of the research on this approach has focused on formulating the contact forces as the unknowns of a *linear complementarity problem* (LCP). Brogliato provides an excellent survey of the main formulations and numerical techniques in [14], and more recent time-stepping schemes have been presented in [70, 7, 59]. But while the time-stepping schemes mentioned above provide a solution for any set of initial conditions and friction coefficients, they do not guarantee a *unique* solution. In fact, it is well-known that the solution set of an arbitrary LCP may not be convex or even connected [21]. In the late 19th century, the French mathematician Paul Painleve provided a single-contact problem in which the LCP may have multiple isolated solutions or no solution at all [56]. As acknowledged in [5, 6, 7], nonconvexity of the solution set may have serious numerical consequences. For example, if an arbitrary solution of the LCP is chosen at each time step, then the computed forces over continuous contact regions may change discontinuously with time, leading to erroneous analysis of continuum properties such as heat or stress. On the other hand, if a minimum-norm solution is sought, the solution set must be fully characterized, which could be computationally prohibitive for complex problems. In fact, a minimum-norm solution of an LCP is may not even be unique.

This dissertation describes how Gauss' Principle of Least Constraint [24] can be used to formulate the kinetic frictional contact problem as a *convex quadratic program* (convex QP). When this QP is feasible, the solution set is convex, making it computationally efficient to identify the *minimum Euclidean norm solution*. This represents a huge advantage over LCP-based methods.

To illustrate this approach, we demonstrate how Gauss' Principle uniquely resolves the solution of Painleve's example in all sliding configurations. Interestingly, these solutions correspond to the stable solutions found in [62] using singular perturbation analysis. Our analysis closely parallels that of Genot and Brogliato [13, 26], in which the well-posedness of this problem is analyzed by mapping out the vanishing and nonvanishing behavior in state space. In order to draw more general conclusions, though, we analyze the dimensionless system with arbitrary control.

We also show how this formulation can be used in conjunction with an explicit s-stage Runge-Kutta solver of arbitrary order to integrate the differential-algebraic equations (DAE's) of motion. At each stage, the normal forces are resolved by finding the minimum Euclidean norm solution of the convex QP. The positions and velocities are integrated using these forces, and then projected onto the constraint manifold. For situations in which constraint changes occur relatively infrequently, this scheme adapts easily to higher-order methods, and it can be combined with automatic time-step adjustment for highly-accurate simulations.

In order to experimentally validate this approach, we have implemented an explicit 5th-order solver based on the formula by Cash and Karp [16], and we have used it to accurately predict the trajectories of a pendulum with unlubricated contact. To our knowledge, this is the first time that continuously-distributed, load-dependent Coulomb friction in a journal bearing has been modeled using a rigid-body unilateral contact model.

Moreau and Brogliato [54, 14] divide the current expanse of rigid body numerical schemes (excluding the *penalty-constraint* schemes described above) into two distinct classes: (a) *event-based schemes*, which apply a differential algebraic equation (DAE) solver between discontinuities in the velocity [58, 28] (see [14] for many more references), and (b) *time-stepping schemes*, which resolve feasibility within the integration scheme itself [7, 42, 43, 59, 64, 70, 52, 54]. Under this classification system, our approach would qualify as an event-based integration scheme. The disadvantage of event-based schemes is that they do not efficiently handle systems in which impacts occur frequently, such as granular flows, or systems with finite accumulation points, such as the bouncing ball example. However, there are many practical systems (e.g., rotating machinery) in which impacts occur infrequently or not at all, and event-based schemes can be applied successfully.

This is certainly not the first time Gauss' Principle has been used to formulate the unilateral contact problem. In the late 19th century [27], Gibbs demonstrated the equivalence between D'Alembert's Principle and Gauss' Principle under variations of the δ_2-type (acceleration) in the absence of friction. Later on Moreau applied Gauss' Principle to the frictionless unilateral contact problem in [48, 49]. More recent work includes [72, 29, 30]. In [29, 30], Glocker formulates nonsmooth dynamical systems in terms of set-valued force laws derived from convex potentials; however, only friction with a *known* normal force can be treated within this framework, and systems with Coulomb friction are treated in terms of linear and nonlinear complementarity (see Chapter 10 of [30]). In [72], Udwadia shows how Gauss' Principle can be used to formulate frictionless systems with nonholonomic equality constraints, but does not address friction or unilateral contact in this work. In [73], however, he proposes a generalization of Gauss' Principle which removes the *nonideal* forces (forces which do work until virtual displacements) from the objective function.

D'Alembert's Principle effectively states that constraint forces do no work, and in the absence of friction it can be shown that this principle is equivalent to Gauss' Principle. Since kinetic friction *does* work, it is traditionally not thought of as a constraint force; instead, it is thought of as an external (applied) force that happens to depend on the normal force (which *is* a constraint force). It is clear, however, that if a contact vanishes (a surface suddenly drops out from underneath an object, for example), then both the normal forces *and* the frictional forces associated with this contact vanish, as well. For multi-contact systems in which the normal and frictional forces have components along the same directions (such as our experimental device) treating the frictional force as an applied force makes little sense since it is tightly coupled with the normal force. We have therefore decided to include friction as part of the *total contact force* which is minimized by Gauss' Principle. We have not yet established the class of systems for which this approach satisfies linear complementarity, but we provide a set of key example problems where it does. In addition, our numerical predictions match our experimental results almost perfectly. It is quite possible that this approach is valid for *any* mechanical system with load-dependent Coulomb friction.

A brief outline of this dissertation is as follows: Chapter 1 reviews some basic concepts of rigid-body mechanics and derives the generalized equations of motion of a rigid mechanical system in 3-dimensions. In Chapter 2, we derive the kinetic frictional contact LCP and discuss some shortcomings of this formulation. We then propose an alternative formulation based on Gauss' Principle and show how this leads to a convex quadratic program in the normal load $\mathbf{N}$. In Chapter 3 we demonstrate how this approach can be used in conjunction with an explicit s-stage Runge-Kutta method to integrate the motion of a system without impacts or stick-slip transitions. Finally, in Chapter 4 we describe the experimental setup and present the experimental results. We show that our numerical scheme very accurately predicts the experimental trajectories of this system and provides physically reasonable force distributions for a discrete approximation of the contact interface.

Chapter 1 : Mechanics of Multibody Systems

In this chapter we review the basic concepts of 3-dimensional rigid body mechanics, including local coordinate frames, mass and inertia, and the Newton-Euler equations of motion. These preliminary concepts are used to formulate the problem of unilateral contact with kinetic friction at the end of this chapter.

1.1 Local Cartesian Coordinate Frames

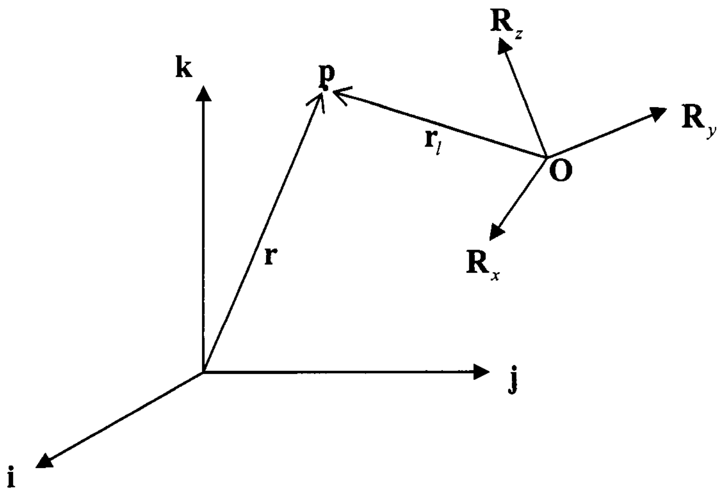

Figure 1.1. A point p specified with respect to a local and the global coordinate frame.

In order to specify positions relative to individual objects, we define a local Cartesian coordinate frame consisting of an origin $\mathbf{O} \in \mathbb{R}^3$ and a set of three orthonormal vectors $\{\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z\} \subset \mathbb{R}^3$. These represent the local *x*, *y*, and *z* axes of the coordinate frame. The *orientation* (or *attitude*) *matrix*, **R**, is a 3 x 3 orthogonal matrix whose first, second, and third columns are $\mathbf{R}_x$, $\mathbf{R}_y$, and $\mathbf{R}_z$, respectively. We denote this local coordinate frame by $\mathcal{F}$. From the definition, we see that **R** satisfies

$$\mathbf{I} = \mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T. \tag{1.1}$$

In other words, $\mathbf{R}^{-1} = \mathbf{R}^T$. In order to state the Newton-Euler equations of rigid body motion later on, we assume that the local coordinate frames of the components of our mechanical system are specified relative to some inertial frame, which we refer to as the *global frame*.

Let $\mathbf{r}$ and $\mathbf{r}_l$ denote the respective coordinate vectors of a point $\mathbf{p}$ relative to the global frame and a local frame $\mathcal{F}$, as shown in Figure 1.1. These two vectors are related by the affine transformation

$$\mathbf{r} = \mathbf{O} + \mathbf{R}\cdot\mathbf{r}_l, \tag{1.2}$$

the inverse of which is

$$\mathbf{r}_l = \mathbf{R}^T(\mathbf{r} - \mathbf{O}). \tag{1.3}$$

Simply put, equation (1.2) transforms a coordinate vector from the local frame $\mathcal{F}$ to the global frame, and (1.3) transforms it back.

Now suppose that this local coordinate frame is rotating at an angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$, where the unit vector $\hat{\boldsymbol{\omega}}$ is the axis of rotation, and $\|\boldsymbol{\omega}\|$ is the rate of rotation in radians per unit time about this axis. The time rate of change of **R** is

$$\dot{\mathbf{R}} = \boldsymbol{\omega} \times \mathbf{R}, \tag{1.4}$$

where the skew-symmetric linear operator

(1.5) $$\boldsymbol{\omega}\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

acts like the vector cross-product on each column of **R**. It is easy to see that this operator satisfies

(1.6) $$\boldsymbol{\omega}\times\boldsymbol{\omega} = 0$$

and

(1.7) $$\boldsymbol{\omega}\times\mathbf{v} = -\mathbf{v}\times\boldsymbol{\omega}$$

for all $\mathbf{v}\in\mathbb{R}^3$. Differentiating equation (1.2) and substituting (1.4), we get the relation

(1.8) $$\dot{\mathbf{r}} = \dot{\mathbf{O}} + \boldsymbol{\omega}\times\mathbf{R}\cdot\mathbf{r}_l + \mathbf{R}\cdot\dot{\mathbf{r}}_l,$$

which is known as Coriolis' Law. If $\mathbf{r}_l$ is fixed with respect to the local coordinate frame, then the third term vanishes, and this expression simplifies to

(1.9) $$\dot{\mathbf{r}} = \dot{\mathbf{O}} + \boldsymbol{\omega}\times\mathbf{R}\cdot\mathbf{r}_l.$$

Expanding (1.4) gives

(1.10) $$\dot{R}_{xx} = \omega_y R_{xz} - \omega_z R_{xy},$$

(1.11) $$\dot{R}_{xy} = \omega_z R_{xx} - \omega_x R_{xz},$$

and so on. These relationships will be used later on in Section 1.3 to construct the *kinematic map*, which relates the velocity variables of the system to the rates of change of the position variables (see equations (1.26) and (1.27)).

1.2 Mechanics of a Single Rigid-Body

The *linear momentum* of a rigid body is defined as

$$\mathbf{p} \equiv m\dot{\mathbf{r}}_{cm}, \tag{1.12}$$

where m is the mass of the body and r_{cm} is the position of the center of mass of the body with respect to the global coordinate frame. The angular momentum of a rigid body about its center of mass is given by

$$\mathbf{L} = \mathbf{J}\boldsymbol{\omega}. \tag{1.13}$$

In this equation $\boldsymbol{\omega}$ is the angular velocity of the body, and the matrix $\mathbf{J} \in \mathbb{R}^{3\times 3}$ is called the *inertia tensor*. The entries of **J** are given by the volume integrals

$$J_{xx} = \iiint_V \left(y^2 + z^2\right)\rho(x,y,z)\,dxdydz, \tag{1.14}$$

$$J_{xy} = -\iiint_V xy\rho(x,y,z)\,dxdydz, \tag{1.15}$$

and so on [71]. The density function $\rho(x,y,z)$ gives the mass per unit volume at each point within the body. These equations can be used directly to compute principal axes and principal moments of inertia for simple shapes of constant density (see [71], pp. 458 – 464).

Clearly, **J** symmetric, and it can be shown to be positive-definite; therefore, the eigenvalues of **J** are positive real numbers, and its eigenvectors are orthonormal. The eigenvalues of **J** are known as the *principal moments of inertia*, and the eigenvectors are know as the *principal axes* of the rigid-body. We can write **J** in its diagonalized form

$$\mathbf{J} = \mathbf{R}\mathbf{J}_p\mathbf{R}^T, \tag{1.16}$$

where the columns of **R** are the principal axes, and the diagonal matrix

$$\mathbf{J}_p = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \tag{1.17}$$

is the *principal inertia tensor*. If the geometry of the body is specified with respect to a local frame with its origin at $\mathbf{r}_{cm}$ and the principal axes are the columns of **R**, then we need only compute J_x, J_y, and J_z once, and then update **J** using equation (1.16) as the orientation **R** changes with each time step. We will refer to this particular coordinate frame as the *body frame* (or *center-of-mass frame*).

When modeling components in CAD software, the local coordinate frame used for geometric modeling usually does not coincide with the body frame described above. If the density function is known, then the following steps can be used to compute the center of mass, principal moments of inertia, and principal axes:

1) In the modeling coordinate frame, compute the center of mass of the body from its geometric description and its density function.

2) Translate all geometric coordinates (vertices, spline control points, etc.) by this value, and then compute the entries of **J** using integrals similar to (1.14) and (1.15).

3) Compute an eigenvalue decomposition of this matrix to find the principal axes and principal moments of inertia.

4) Project the translated geometric coordinates (computed in step 2) onto the principal vectors. The geometry should now be specified with respect to the body frame.

Newton's second law of motion states that the linear momentum of a body's center of mass changes if and only if it is acted upon by a net *force* $\mathbf{f} \in \mathbb{R}^3$, and its rate of change is described by the well-known relationship

$$\mathbf{f} \equiv \dot{\mathbf{p}} = m\ddot{\mathbf{r}}_{cm} . \tag{1.18}$$

Similarly, the net *torque* $\boldsymbol{\tau} \in \mathbb{R}^3$ on the body is defined as

$$\boldsymbol{\tau} \equiv \dot{\mathbf{L}} . \tag{1.19}$$

From equations (1.4) and (1.13), the right-hand side of this expression expands to

$$\boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{R}\mathbf{J}_p\mathbf{R}^T\boldsymbol{\omega} - \mathbf{R}\mathbf{J}_p\mathbf{R}^T\boldsymbol{\omega} \times \boldsymbol{\omega} , \tag{1.20}$$

but since $\boldsymbol{\omega} \times \boldsymbol{\omega} = 0$, this is simply

$$\boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{L} . \tag{1.21}$$

Rearranging (1.18) and (1.21) gives the Newton-Euler equations of rigid-body motion:

$$m\ddot{\mathbf{r}}_{cm} = \mathbf{f} , \tag{1.22}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} - \boldsymbol{\omega} \times \mathbf{L} . \tag{1.23}$$

From (1.23) it follows that any angular velocity $\boldsymbol{\omega} \neq 0$ not parallel to a principal axis will result in $\boldsymbol{\omega} \times \mathbf{L} \neq 0$. This causes the angular velocity to oscillate about a fixed axis even when $\boldsymbol{\tau} = 0$, a phenomenon known as *torque-free precession*[1].

1.3 Generalized Coordinates

To put these equations into a more general context, we define the *generalized position* (or *configuration*) of a single rigid-body $\mathcal{B}$ to be the augmented vector

$$\textbf{(1.24)} \qquad \mathbf{x} \equiv \left[r_{cm,x}, r_{cm,y}, r_{cm,z}, R_{xx}, R_{xy}, \ldots, R_{zy}, R_{zz} \right]^T \in \mathbb{R}^{12},$$

and its *generalized velocity* as

$$\textbf{(1.25)} \qquad \mathbf{u} \equiv \left[\dot{r}_{cm,x}, \dot{r}_{cm,y}, \dot{r}_{cm,z}, \omega_x, \omega_y, \omega_z \right]^T \in \mathbb{R}^6 .$$

These two vectors comprise the mechanical *state* of $\mathcal{B}$. It is clear from (1.24) and (1.25) that $\mathbf{u} \neq \dot{\mathbf{x}}$, and in general these two variables are related by the quasilinear differential relationship

$$\textbf{(1.26)} \qquad \dot{\mathbf{x}} = \mathbf{H}(\mathbf{x})\mathbf{u},$$

where the matrix $\mathbf{H}(\mathbf{x})$ is known as the *kinematic map* [70]. It follows from (1.4) and (1.6) that

---

[1] To observe torque-free precession, spin a pencil along its long axis and then throw it up in the air. The pencil will have the majority of its spin directed along its long axis, but it will also "wobble" slowly about a fixed axis of precession.

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} [\mathbf{I}_{3\times3}] & [\mathbf{0}_{3\times3}] \\ [\mathbf{0}_{3\times3}] & [-\mathbf{R}_x \times] \\ [\mathbf{0}_{3\times3}] & [-\mathbf{R}_y \times] \\ [\mathbf{0}_{3\times3}] & [-\mathbf{R}_z \times] \end{bmatrix} \in \mathbb{R}^{12\times6}, \tag{1.27}$$

where, following equation (1.5), we have

$$\mathbf{R}_x \times = \begin{bmatrix} 0 & -R_{xz} & R_{xy} \\ R_{xz} & 0 & -R_{xx} \\ -R_{xy} & R_{xx} & 0 \end{bmatrix}, \tag{1.28}$$

and so on. We can show that **H** has full column rank by first computing

$$\mathbf{H}^T\mathbf{H} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -(\mathbf{R}_x \times)^2 - (\mathbf{R}_y \times)^2 - (\mathbf{R}_z \times)^2 \end{bmatrix}. \tag{1.29}$$

It follows from the vector identity [23]

$$\mathbf{a} \times \mathbf{b} \times \mathbf{c} = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{a} \cdot \mathbf{b})\mathbf{c} \tag{1.30}$$

that

$$\left[-(\mathbf{R}_x \times)^2 - (\mathbf{R}_y \times)^2 - (\mathbf{R}_z \times)^2\right]\mathbf{v} = 3\mathbf{v} - \mathbf{R}\mathbf{R}^T\mathbf{v} = 2\mathbf{v}. \tag{1.31}$$

From this we have

$$\mathbf{H}^T\mathbf{H} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & 2\mathbf{I}_{3\times3} \end{bmatrix}, \tag{1.32}$$

and therefore $rank(\mathbf{H}) = 6$. It is clear from (1.32) that the largest singular value of **H** is $\sigma_1 = \sqrt{2}$, and the smallest singular value is $\sigma_6 = 1$. The Euclidean condition number is therefore $\kappa(\mathbf{H}) = {}^{\sigma_1}\!/_{\sigma_6} = \sqrt{2}$, regardless of the orientation. This is nearly ideal from our point of view, since it will not significantly increase the condition number of the linear systems we must solve when we integrate the equations of motion under constraints[2] [67].

The *mass matrix* of a rigid body is the symmetric positive-definite matrix

$$\mathbf{M}(\mathbf{x}) \equiv \begin{bmatrix} \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{J} \end{bmatrix} \in \mathbb{R}^{6\times 6}, \tag{1.33}$$

and the net *generalized force* is the augmented vector

$$\mathbf{F}(t,\mathbf{x},\mathbf{u}) \equiv [\mathbf{f} \mid \boldsymbol{\tau} - \boldsymbol{\omega} \times \mathbf{L}]^T \in \mathbb{R}^6. \tag{1.34}$$

1.4 The Newton-Euler Equations

Using these generalized variables, the Newton-Euler equations simplify to

$$\dot{\mathbf{x}} = \mathbf{H}(\mathbf{x})\mathbf{u}, \tag{1.35}$$

$$\mathbf{M}(\mathbf{x})\,\dot{\mathbf{u}} = \mathbf{F}(t,\mathbf{x},\mathbf{u}). \tag{1.36}$$

In fact, these equations generalize in their current form to systems with multiple rigid bodies. In this case, all of the positions and orientations are collected into a single generalized position $\mathbf{x} \in \mathbb{R}^{n_x}$, and all the linear and angular velocities are collected into a single generalized velocity $\mathbf{u} \in \mathbb{R}^n$. The time derivative of $\mathbf{x}$ is still related to $\mathbf{u}$ by equation (1.26), and $\mathbf{H} \in \mathbb{R}^{n_x \times n}$ must still have full column rank and be continuously differentiable in $\mathbf{x}$. The net generalized force $\mathbf{F}$ now contains

---

[2] This is discussed in more detail in Chapter 3.

all of the generalized forces on the system, including torque-free precession terms. For a multibody system, the mass matrix has the block-diagonal form

$$\mathbf{M}(\mathbf{x}) = \begin{bmatrix} \mathbf{M}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_{n_c} \end{bmatrix}, \tag{1.37}$$

where each $\mathbf{M}_k$ is the mass matrix of the k^{th} component, and n_c is the total number of components in the system. The positive-definiteness of $\mathbf{M}$ follows from the positive-definiteness of each $\mathbf{M}_k$.

Chapter 2 : Unilateral Contact with Kinetic Friction

In the previous chapter, we derived a generalized form of the Newton-Euler equations which takes into account every dynamic variable in the system. These equations provide the basis for formulating the unilateral contact problem. In this chapter we derive the force relationships between contacting bodies, and we state the problem of unilateral contact with kinetic friction in concise terms. Two approaches to resolving the contact forces are described: one based on formulating the normal loads as the unknowns of a *linear complementarity problem* (LCP), and the other based on Gauss' Principle of Least Constraint. The latter approach leads to a *convex quadratic program*, which offers much better mathematical properties and avoids some numerical difficulties associated with the LCP.

2.1 Nonpenetration Constraints

In multi-component systems, individual parts affect one another through direct contact. In general, these contacts are continuous (journal bearings, universal joints, etc.), but may be accurately approximated by discretizing them into a sufficient number of isolated points. Each of these contact points has an associated *unit contact normal* $\hat{\mathbf{n}}_i$ as shown in Figure 2.1, below.

If there are $m+s$ such contacts between the various components, then the nonpenetration constraints can be expressed as a sequence of algebraic inequalities

$$g_i(\mathbf{x}) \le 0 \, , \; i = 1, 2, \ldots, m+s \, , \tag{2.1}$$

where each g_i is assumed to be twice-continuously differentiable, and its gradient $\mathbf{g}_{i,\mathbf{x}} \equiv \frac{\partial \mathbf{g}_i}{\partial \mathbf{x}} \in \mathbb{R}^{n_x *}$ does not vanish in a neighborhood of $\mathbf{x}$. Due to the sign convention[3], these functions provide a local measure of the *depth of penetration* at the contact point; however, it is not essential that they give the actual (dimensional) penetration depth, provided that $g_i(\mathbf{x}) < 0$ when the bodies are separated, $g_i(\mathbf{x}) = 0$ when they are in contact, and $g_i(\mathbf{x}) > 0$ when they are interpenetrating.

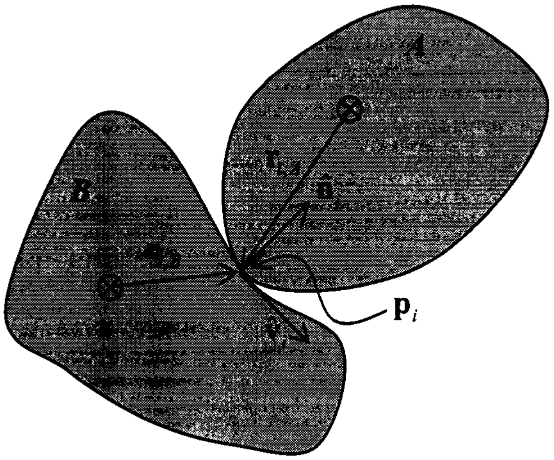

Figure 2.1. Two rigid-bodies *A* and *B* contacting at a single point.

If $\mathbf{x}$ is a twice-continuously differentiable function of time, then (2.1) can be expanded as a 2nd order Taylor series

$$g_i(\mathbf{x}(t)) = g_i(\mathbf{x}_0) + \dot{g}_i(\mathbf{x}_0)(t - t_0) + \frac{1}{2}\ddot{g}_i(\mathbf{x}_0)(t - t_0)^2 + o\left(|t - t_0|^2\right) \le 0, \tag{2.2}$$

which is valid on a sufficiently small neighborhood of t_0. If both $g_i(\mathbf{x}_0) \le 0$ and $\dot{g}_i(\mathbf{x}_0) \le 0$, then the initial values $\mathbf{x}_0$ and $\mathbf{u}_0$ are said to be *consistent*. In practice,

---

[3] This sign convention comes from optimization theory, which is opposite the standard engineering sign convention where nonpenetration constraints specify the distance of *separation* between objects.

contacts for which $g_i(\mathbf{x}_0) < 0$ or $\dot{g}_i(\mathbf{x}_0) < 0$ can be safely ignored since they do not produce contact forces.

When $g_i(\mathbf{x}_0) = 0$ the contact is said to be *active*, and we are left with

$$g_i(\mathbf{x}(t)) = \dot{g}_i(\mathbf{x}_0)(t - t_0) + O\left(|t - t_0|^2\right) \le 0. \tag{2.3}$$

From this we see that if $\dot{g}_i(\mathbf{x}_0) > 0$, then (2.1) cannot be satisfied[4] on $(t_0, t_0 + h)$ for any $h > 0$. Therefore, we have the *velocity-level* constraint

$$\dot{g}_i(\mathbf{x}_0) \le 0. \tag{2.4}$$

Similarly, at the instant when both $g_i(\mathbf{x}_0) = 0$ and $\dot{g}_i(\mathbf{x}_0) = 0$, the *acceleration-level* constraint

$$\ddot{g}_i(\mathbf{x}_0) \le 0 \tag{2.5}$$

must be satisfied. If $\ddot{g}_i(\mathbf{x}_0) < 0$ at this instant, then the contact is said to be *vanishing*.

In order to express the above constraints in terms of $\mathbf{u}$ and $\dot{\mathbf{u}}$, we can apply the Chain Rule and substitute equation (1.35) to get

$$\dot{g}_i(\mathbf{x}(t)) = \mathbf{g}_{i,\mathbf{x}}\dot{\mathbf{x}} = \mathbf{g}_{i,\mathbf{x}}\mathbf{H}\mathbf{u} \le 0. \tag{2.6}$$

Differentiating this expression once more gives

---

[4] We do not consider impacts in which $g_i(\mathbf{x}(t_0)) = 0$ and $\dot{g}_i(\mathbf{x}(t_0)) > 0$. The solution of this problem requires the use of differential measures and restitution laws, and the reader is referred to [13] for an in-depth discussion of these topics.

(2.7) $$\ddot{g}_i\left(\mathbf{x}(t)\right)=\frac{d}{dt}\left(\mathbf{g}_{i,\mathbf{x}}\mathbf{H}\right)\mathbf{u}+\mathbf{g}_{i,\mathbf{x}}\mathbf{H}\dot{\mathbf{u}}\leq 0\,.$$

From (2.6) we can see that the value of $\dot{g}_i\left(\mathbf{x}_0\right)$ is easily computed from the prescribed initial values. However, if $\dot{g}_i\left(\mathbf{x}_0\right)=0$, then determining the value of $\ddot{g}_i\left(\mathbf{x}_0\right)$ is not as straightforward since $\dot{\mathbf{u}}$ is not known a priori. In the next section, we convert (2.7) into additional constraints upon the normal forces of the system.

2.2 Contact Forces

Let $\mathbf{p}_i$ and $\hat{\mathbf{n}}_i$ denote the i^{th} contact point and the associated unit normal, respectively, with the convention that $\hat{\mathbf{n}}_i$ points from component ***B*** to component ***A***. At this point, ***B*** exerts a normal force on ***A*** equal to

(2.8) $$\mathbf{N}_i=N_i\hat{\mathbf{n}}_i,$$

where $N_i\geq 0$. In other words, normal forces are *compressive.* Tangential slippage induces a *kinetic* frictional force on ***A*** equal to

(2.9) $$\mathbf{f}_i=-f_i\left(\left\|\mathbf{v}_i\right\|\right)N_i\hat{\mathbf{v}}_i,$$

and ***B*** experiences an *equal and opposite* force. Here $\mathbf{v}_i$ is the sliding velocity of ***A*** relative to ***B*** at the point $\mathbf{p}_i$, and $f_i\left(\left\|\mathbf{v}_i\right\|\right)$ maps the magnitude of the sliding velocity to the magnitude of kinetic friction.

Due to these contact forces, ***A*** experiences a torque equal to

(2.10) $$\boldsymbol{\tau}_{i,A}=\mathbf{r}_{i,A}\times\left(\mathbf{f}_i+\mathbf{N}_i\right),$$

where $\mathbf{r}_{i,A}$ is the vector from the center of mass of A to the contact point $\mathbf{p}_i$ with respect to the global coordinate frame. The torque that this contact point exerts on $\boldsymbol{B}$ is

$$\boldsymbol{\tau}_{i,B} = \mathbf{r}_{i,B} \times \left(\mathbf{f}_i + \mathbf{N}_i\right). \tag{2.11}$$

The *generalized contact force* that arises from this interaction is therefore:

$$\begin{aligned} \mathbf{F}_{c,i} = \big[0,\ldots 0, | \, \hat{\mathbf{n}}_i - f_i \hat{\mathbf{v}}_i \, | \, \mathbf{r}_{i,A} \times \left(\hat{\mathbf{n}}_i - f_i \hat{\mathbf{v}}_i\right) | \, 0,\ldots \\ \ldots 0, | \, f_i \hat{\mathbf{v}}_i - \hat{\mathbf{n}}_i \, | \, \mathbf{r}_{i,B} \times \left(f_i \hat{\mathbf{v}}_i - \hat{\mathbf{n}}_i\right) | \, 0,\ldots 0\big]^T N_i. \end{aligned} \tag{2.12}$$

The zero entries account for the fact that these forces directly influence only the two interacting bodies. This expression can be written more concisely as

$$\mathbf{F}_{c,i} = -\mathbf{w}_i^T N_i, \tag{2.13}$$

where the row vector

$$\begin{aligned} \mathbf{w}_i = -\big[0,\ldots 0, | \, \hat{\mathbf{n}}_i - f_i \hat{\mathbf{v}}_i \, | \, \mathbf{r}_{i,A} \times \left(\hat{\mathbf{n}}_i - f_i \hat{\mathbf{v}}_i\right) | \, 0,\ldots \\ \ldots 0, | \, f_i \hat{\mathbf{v}}_i - \hat{\mathbf{n}}_i \, | \, \mathbf{r}_{i,B} \times \left(f_i \hat{\mathbf{v}}_i - \hat{\mathbf{n}}_i\right) | \, 0,\ldots 0\big] \in \mathbb{R}^{n^*} \end{aligned} \tag{2.14}$$

is the *contact wrench*. The sum of these forces over all m+s contacts gives the net generalized contact force

$$\mathbf{F}_c = -\sum_{i=1}^{m+s} \mathbf{w}_i^T N_i = -\tilde{\mathbf{W}}^T \tilde{\mathbf{N}}, \tag{2.15}$$

where the rows of $\tilde{\mathbf{W}} \in \mathbb{R}^{(m+s) \times n}$ are of the form specified in equation (2.14), and the elements of $\tilde{\mathbf{N}} \in \mathbb{R}_+^{m+s}$ are the individual normal forces N_i. The matrix $\tilde{\mathbf{W}}$ is the *wrench matrix*, and the vector $\tilde{\mathbf{N}}$ is the *generalized normal force*. The '~' denotes the fact that s of these contact forces are vanishing at this instant.

The right-hand side of equation (1.36) can now be decomposed into two components: *noncontact forces*, $\mathbf{F}_{nc}$, such as gravity or torque-free precession terms, and *contact forces*, $\mathbf{F}_c$, which arise from the force interactions at the contact points:

$$\mathbf{M}(\mathbf{x})\dot{\mathbf{u}} = \mathbf{F}_{nc}(t,\mathbf{x},\mathbf{u}) - \tilde{\mathbf{W}}^T\tilde{\mathbf{N}}. \tag{2.16}$$

Since $\tilde{\mathbf{W}}$ can be calculated explicitly from the initial values, each $\ddot{g}_i(\mathbf{x})$ can be expressed in terms of $\tilde{\mathbf{N}}$ by first solving equation (2.16) for $\dot{\mathbf{u}}$ to get

$$\dot{\mathbf{u}} = \mathbf{M}^{-1}\mathbf{F}_{nc} - \mathbf{M}^{-1}\tilde{\mathbf{W}}^T\tilde{\mathbf{N}}, \tag{2.17}$$

and then substituting this back into (2.7) to get

$$\ddot{g}_i(\mathbf{x}) = \frac{d}{dt}\left(\mathbf{g}_{i,\mathbf{x}}\mathbf{H}\right)\mathbf{u} + \mathbf{g}_{i,\mathbf{x}}\mathbf{H}\mathbf{M}^{-1}\mathbf{F}_{nc} - \mathbf{g}_{i,\mathbf{x}}\mathbf{H}\mathbf{M}^{-1}\tilde{\mathbf{W}}^T\tilde{\mathbf{N}} \le 0. \tag{2.18}$$

If we define

$$\tilde{\mathbf{G}}(\mathbf{x}) \equiv \left[g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_{m+s}(\mathbf{x})\right]^T, \tag{2.19}$$

then we can express (2.18) in matrix notation as

$$\ddot{\tilde{\mathbf{G}}}(\mathbf{x}) = \frac{d}{dt}\left(\tilde{\mathbf{G}}_{\mathbf{x}}\mathbf{H}\right)\mathbf{u} + \tilde{\mathbf{G}}_{\mathbf{x}}\mathbf{H}\mathbf{M}^{-1}\mathbf{F}_{nc} - \left(\tilde{\mathbf{G}}_{\mathbf{x}}\mathbf{H}\mathbf{M}^{-1}\tilde{\mathbf{W}}^T\right)\tilde{\mathbf{N}} \le 0. \tag{2.20}$$

In order to write (2.20) more concisely, we define

$$\tilde{\mathbf{b}} \equiv \frac{d}{dt}\left(\tilde{\mathbf{G}}_{\mathbf{x}}\mathbf{H}\right)\mathbf{u} + \tilde{\mathbf{G}}_{\mathbf{x}}\mathbf{H}\mathbf{M}^{-1}\mathbf{F}_{nc}, \tag{2.21}$$

$$\tilde{\mathbf{A}} \equiv \tilde{\mathbf{G}}_{\mathbf{x}}\mathbf{H}\mathbf{M}^{-1}\tilde{\mathbf{W}}^T, \tag{2.22}$$

so that

(2.23) $$\ddot{\tilde{\mathbf{G}}}(\mathbf{x}) = \tilde{\mathbf{b}} - \tilde{\mathbf{A}}\tilde{\mathbf{N}} \le 0 .$$

2.3 The Differential-Algebraic System

Since we do not consider systems in which impacts could occur, we seek a twice-continuously differentiable solution $\mathbf{x}(t)$ to the following *differential-algebraic inequality* (DAI):

(2.24) $$\dot{\mathbf{x}} = \mathbf{H}(\mathbf{x})\mathbf{u} ,$$

(2.25) $$\mathbf{M}(\mathbf{x})\dot{\mathbf{u}} = \mathbf{F}_{nc}(t, \mathbf{x}, \mathbf{u}) - \tilde{\mathbf{W}}^T\tilde{\mathbf{N}} ,$$

(2.26) $$\ddot{\tilde{\mathbf{G}}}(\mathbf{x}) = \tilde{\mathbf{b}} - \tilde{\mathbf{A}}\tilde{\mathbf{N}} \le 0 ,$$

(2.27) $$\tilde{\mathbf{N}} \ge 0 ,$$

(2.28) $$\mathbf{x}(t_0) = \mathbf{x}_0 ,$$

(2.29) $$\mathbf{u}(t_0) = \mathbf{u}_0 ,$$

where the initial values are assumed to be consistent. If $\tilde{\mathbf{N}}$ can be found for which both inequalities are satisfied, then it can be substituted back into (2.26) to determine the m contacts for which $\ddot{g}_i(\mathbf{x}) = 0$ (the *active set*). For the s vanishing contacts, we know that $N_i = 0$, and there is a *reduced* normal force $\mathbf{N} \in \mathbb{R}_+^m$ which produces exactly the same motion as $\tilde{\mathbf{N}}$. From this, we are left with the following index 1 *differential-algebraic equation* (DAE):

(2.30) $$\dot{\mathbf{x}} = \mathbf{H}(\mathbf{x})\mathbf{u} ,$$

(2.31) $$\mathbf{M}(\mathbf{x})\dot{\mathbf{u}} = \mathbf{F}_{nc}(t, \mathbf{x}, \mathbf{u}) - \mathbf{W}^T\mathbf{N} ,$$

(2.32) $$\ddot{\mathbf{G}}(\mathbf{x}) = \mathbf{b} - \mathbf{A}\mathbf{N} = 0 ,$$

(2.33) $$\mathbf{x}(t_0) = \mathbf{x}_0 ,$$

(2.34) $$\mathbf{u}(t_0) = \mathbf{u}_0 .$$

Due to the interdependence of contact forces with nonpenetration constraints, efficiently computing an $\tilde{\mathbf{N}}$ which solves the system in (2.30) – (2.34) is rarely straightforward. In general, simple principles such as conservation of momentum and energy will not resolve the inequality constraints. And since the combinatorial complexity grows as 2^m, with *m* being the number of contacts, problems with more than a few contacts cannot be solved efficiently using brute force trial and error. So in order to efficiently solve large-scale unilateral contact problems, we must employ a more general principle.

2.4 Linear Complementarity

If we assume that we cannot simultaneously have $N_i > 0$ and $\ddot{g}_i(\mathbf{x}) < 0$ (i.e., a positive normal force precludes a vanishing contact and vice-versa), then the generalized normal force $\tilde{\mathbf{N}}$ must satisfy the following component-wise *linear complementarity problem* (LCP):

(2.35)
$$\begin{gathered} N_i \geq 0, \\ \ddot{g}_i(\mathbf{x}) \leq 0, \\ N_i \ddot{g}_i(\mathbf{x}) = 0. \end{gathered}$$

In other words, we seek $\tilde{\mathbf{N}} \in \mathbb{R}_+^{m+s}$ such that $\ddot{g}_i(\mathbf{x}) < 0 \Rightarrow N_i = 0$, and $N_i > 0 \Rightarrow \ddot{g}_i(\mathbf{x}) = 0$ for each contact (it is also valid to have $N_i = 0$ and $\ddot{g}_i(\mathbf{x}) = 0$ simultaneously, a situation known as *grazing contact*). If we substitute (2.21) – (2.22) into (2.35), then we can rewrite the LCP in terms of $\tilde{\mathbf{N}}$ alone:

(2.36)
$$\begin{gathered} \tilde{\mathbf{N}} \geq 0, \\ \tilde{\mathbf{A}}\tilde{\mathbf{N}} - \tilde{\mathbf{b}} \geq 0, \\ \tilde{\mathbf{N}}^T\left(\tilde{\mathbf{A}}\tilde{\mathbf{N}} - \tilde{\mathbf{b}}\right) = 0. \end{gathered}$$

It is easy to show that this LCP is equivalent to the component-wise LCP (2.35).

The *feasible set* (or *feasible region*), denoted by Φ, is the set of $\tilde{\mathbf{N}}$ for which both inequalities in (2.36) are satisfied. The solution set of this LCP is

$$\text{SOL}\left(\tilde{\mathbf{A}},\tilde{\mathbf{b}}\right)=\left\{\tilde{\mathbf{N}}\in\Phi:\tilde{\mathbf{N}}^T\left(\tilde{\mathbf{A}}\tilde{\mathbf{N}}-\tilde{\mathbf{b}}\right)=0\right\}. \tag{2.37}$$

Graphically, $\text{SOL}\left(\tilde{\mathbf{A}},\tilde{\mathbf{b}}\right)$ is the intersection of the feasible region with the quadratic isocline $\tilde{\mathbf{N}}^T\left(\tilde{\mathbf{A}}\tilde{\mathbf{N}}-\tilde{\mathbf{b}}\right)=0$. When Φ is nonempty, an LCP is said to be *feasible*, and if it has at least one solution, then it is said to be *solvable*. It is easy to see from (2.35) that any solution of the LCP lies on the boundary of Φ.

In general, the feasibility of an LCP does not imply its solvability: if $\tilde{\mathbf{A}}$ is positive-semidefinite, then $\text{SOL}\left(\tilde{\mathbf{A}},\tilde{\mathbf{b}}\right)$ is convex. However, if $\tilde{\mathbf{A}}$ is indefinite or negative-definite, then (2.36) may not even have a solution, and if it does, $\text{SOL}\left(\tilde{\mathbf{A}},\tilde{\mathbf{b}}\right)$ may not be convex or even connected [21]. In the late 19th century, the French mathematician Paul Painleve pointed out that the contact LCP for an inclined rod sliding along a horizontal surface may have multiple isolated solutions (*multiplicity*) or no solution at all (*inconsistency*), depending on the friction coefficient and other parameters of the problem [56].

By formulating the contact problem in terms of velocities and impulses, several authors have devised time-stepping schemes which guarantee the existence of a solution at every time step, regardless of the friction coefficients [7, 22, 59], [64]. However, all of these time-stepping schemes formulate their solution as either an LCP or a mixed LCP, and thus may possess a nonconvex solution space, as noted in [5–7].

In the next section, we examine an *optimality* formulation for rigid mechanical systems with kinetic frictional contact. In this formulation, the normal forces are provided as the solution of a convex quadratic program, which resolves the issues discussed above.

2.5 Gauss' Principle of Least Constraint

In 1829 the German mathematician Karl Friedrich Gauss proposed the Principle of Least Constraint [24]:

> "The motion of a system of material points, connected with each other in an arbitrary way and subjected to arbitrary influences takes place at every instant, in the most perfect accordance possible with the motion that they would have if they became completely free, that is to say, with the smallest possible constraint, taking as measure of the constraint [that the system goes through] during an infinitesimally small instant, the sum of the products of the mass of each point with the square of the quantity by which it deviates from the position that it would have taken, if it had been free."
>
> K.F. Gauss, 1829
> (translation from [57]).

We take an expanded view of this statement and interpret it to mean that the motion of a constrained mechanical system will result in a net generalized contact force $\mathbf{F}_c$ which minimizes

(2.38)

$$\varphi(\mathbf{F}_c) = \frac{1}{2}\mathbf{F}_c^T\mathbf{M}^{-1}\mathbf{F}_c$$

over the set of feasible $\mathbf{F}_c$. Since $\mathbf{M}^{-1}$ is positive-definite, $\varphi(\mathbf{F}_c)$ is *strictly* convex. If the feasible region is convex and nonempty, then the constrained minimum of (2.38) is unique. If no constraints are imposed on the system, then the minimum lies at $\mathbf{F}_c = 0$, and (2.38) reduces to Newton's 2nd Law.

Substituting (2.15) into (2.38) and using the acceleration-level constraints (2.26) and (2.27), we get the following *convex quadratic program* (convex QP):

$$\text{(2.39)}\qquad \begin{gathered} \min \tfrac{1}{2}\tilde{\mathbf{N}}^T\left(\tilde{\mathbf{W}}\mathbf{M}^{-1}\tilde{\mathbf{W}}^T\right)\tilde{\mathbf{N}} \\ \text{subject to:} \\ \tilde{\mathbf{N}} \geq 0, \\ \ddot{\tilde{\mathbf{G}}}(\mathbf{x}) = \tilde{\mathbf{b}} - \tilde{\mathbf{A}}\tilde{\mathbf{N}} \leq 0. \end{gathered}$$

Note that the feasible set Φ of this convex QP is identical to that of the LCP (2.36). If Φ is nonempty, then an optimal solution $\tilde{\mathbf{N}}_0$ exists, and the *fundamental normal cone optimality condition*

$$\text{(2.40)}\qquad -\partial_{\tilde{\mathbf{N}}}\varphi\left(\tilde{\mathbf{N}}_0\right) \in \hat{N}_\Phi\left(\tilde{\mathbf{N}}_0\right)$$

must be satisfied. In this inclusion, $\partial_{\tilde{\mathbf{N}}}\varphi\left(\tilde{\mathbf{N}}_0\right)$ is the gradient of φ at $\tilde{\mathbf{N}}_0$, and $\hat{N}_\Phi\left(\tilde{\mathbf{N}}_0\right)$ is the *regular normal cone* of the feasible set Φ at this point [18, 61]. Since problem (2.39) is *convex*, condition (2.40) is *sufficient* for optimality (see [61], Theorem 9), and condition (2.40) can also be expressed equivalently using Lagrange multipliers:

$$\text{(2.41)}\qquad \left(\tilde{\mathbf{W}}\mathbf{M}^{-1}\tilde{\mathbf{W}}^T\right)\tilde{\mathbf{N}} = \tilde{\mathbf{A}}^T\boldsymbol{\lambda} + \mathbf{z}.$$

Here, the entries of $\boldsymbol{\lambda} \in \mathbb{R}_+^{m+s}$ are the Lagrange multipliers for the nonpenetration constraints in (2.26), and the entries of $\mathbf{z} \in \mathbb{R}_+^{m+s}$ are the Lagrange multipliers for the nonnegativity constraints in (2.27). Since the normal cone is the nonnegative span of the gradients of the active constraints at $\tilde{\mathbf{N}}_0$, the following implications hold:

(2.42) $$\ddot{g}_i\left(\tilde{\mathbf{N}}_0\right)<0 \Rightarrow \lambda_i=0,$$

(2.43) $$\ddot{g}_i\left(\tilde{\mathbf{N}}_0\right)=0 \Rightarrow \lambda_i \geq 0,$$

(2.44) $$N_{0,i}>0 \Rightarrow z_i=0,$$

(2.45) $$N_{0,i}=0 \Rightarrow z_i \geq 0.$$

If (2.39) is feasible, $\tilde{\mathbf{W}}$ has full rank, and $m+s \leq n$, then $\tilde{\mathbf{W}}\mathbf{M}^{-1}\tilde{\mathbf{W}}^T$ is positive definite and the solution of (2.39) is unique. However, when continuous contact interfaces are approximated by multiple contact points (as we have done in our experiment), it may be that $m+s>n$ and $\tilde{\mathbf{W}}\mathbf{M}^{-1}\tilde{\mathbf{W}}^T$ is positive-semidefinite. In this case, the solutions form a convex set.

By the uniqueness of $\mathbf{F}_c$, any of these solutions will give identical motion. This multiplicity can be resolved by choosing the particular $\tilde{\mathbf{N}}_0$ which *minimizes the Euclidean norm.* This choice works in many simple cases, and it provides acceptable force distributions for our particular experimental setup (see Figure 4.17 and Figure 4.18). Whether or not the minimum Euclidean norm solution is the physically correct choice remains to be established by experiment.

2.6 Comments on D'Alembert's Principle

It was shown by Gibbs [27] that for unilateral contact problems without friction, D'Alembert's principle generalizes to the condition that

(2.46) $$\left[\mathbf{M}\ddot{\mathbf{u}}_0-\mathbf{F}_{nc}\right]^T \delta_2\ddot{\mathbf{u}}_0 \geq 0$$

must be satisfied for all vectors $\delta_2\dot{\mathbf{u}}_0 \in \mathbb{R}^n$ at $\dot{\mathbf{u}}_0$, where $\delta_2\dot{\mathbf{u}}_0$ is a "virtual variation" of $\dot{\mathbf{u}}_0$ "compatible" with the acceleration-level constraints. These variations are taken without taking any variations of the position, velocity or time. More precisely, the $\delta_2\dot{\mathbf{u}}_0$ are the vectors *tangent* (in the sense of Rockafellar) to the feasible region at the point $\dot{\mathbf{u}}_0$. Defining

$$\varphi(\dot{\mathbf{u}}) = \frac{1}{2}[\mathbf{M}\dot{\mathbf{u}} - \mathbf{F}_{nc}]^T \mathbf{M}^{-1} [\mathbf{M}\dot{\mathbf{u}} - \mathbf{F}_{nc}] \tag{2.47}$$

we see that (2.46) is simply the first-order necessary condition for optimality:

$$\partial_{\dot{\mathbf{u}}}\varphi(\dot{\mathbf{u}}_0)^T \delta_2\dot{\mathbf{u}}_0 \geq 0 \text{ for all } \delta_2\dot{\mathbf{u}}_0 \in T_\Phi(\dot{\mathbf{u}}_0) \tag{2.48}$$

where $T_\Phi(\dot{\mathbf{u}}_0)$ is the *tangent cone* of Φ at $\dot{\mathbf{u}}_0$. Since φ is convex, $\dot{\mathbf{u}}_0$ is the *unique* optimal point (see [61], Theorems 6 and 7), and since

$$\mathbf{F}_c = \mathbf{M}\dot{\mathbf{u}} - \mathbf{F}_{nc} \tag{2.49}$$

the problem of minimizing (2.38) over Φ is equivalent to finding the point $\dot{\mathbf{u}}_0$ that satisfies (2.46) on Φ.

If $\mathbf{F}_c$ includes the frictional forces, however, then our expanded interpretation of Gauss' Principle is no longer equivalent to D'Alembert's Principle. We can see this through the following idealized example: suppose an infinitesimal particle of mass *m* is sliding along the positive *x*-direction between two horizontal planes with gravity pulling downward. Let N_1 denote the *upward* normal force that the lower plane exerts on the particle, and let N_2 denote the *downward* normal force that the upper plane exerts on the particle. The equations of motion for this system are simply

$$m\ddot{r}_x = -\mu\left(N_1 + N_2\right), \tag{2.50}$$

$$m\ddot{r}_y = -mg + N_1 - N_2 . \tag{2.51}$$

The solution to this problem is obviously $N_1 = mg$, $N_2 = 0$.

The contact constraints for this problem are

$$\ddot{g}_1\left(\mathbf{x}\right) = -\ddot{r}_y \le 0 , \tag{2.52}$$

$$\ddot{g}_2\left(\mathbf{x}\right) = \ddot{r}_y \le 0 , \tag{2.53}$$

so the tangent cone at any point in the set of feasible accelerations is the x-axis. Thus, D'Alembert's Principle would require that

$$F_{c,x} = 0 , \tag{2.54}$$

which is clearly *not* satisfied by the total contact force $\mathbf{F}_c = \left[-\mu mg, mg\right]^T$. However, we will show below that our application of Gauss' Principle (as formulated in (2.39)) solves this problem correctly for all $\mu > 0$, and when $\mu = 0$ the correct answer is given by choosing the minimum Euclidean solution.

Using the *dimensionless time* as defined in equation (2.69) in the next section (for this problem, l is arbitrary), we get the dimensionless equations of motion

$$\ddot{\bar{r}}_x = -\mu\left(\bar{N}_1 + \bar{N}_2\right), \tag{2.55}$$

$$\ddot{\bar{r}}_y = -1 + \bar{N}_1 - \bar{N}_2 , \tag{2.56}$$

where $\bar{N}_i \equiv {N_i}/{mg}$. This gives the wrench matrix

$$\mathbf{W} = \begin{bmatrix} \mu & -1 \\ \mu & 1 \end{bmatrix}, \tag{2.57}$$

and the objective function

(2.58)
$$\varphi\left(\tilde{\mathbf{N}}\right)=\frac{1}{2}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^{T}=\frac{1}{2}\begin{bmatrix}\mu^2+1 & \mu^2-1\\ \mu^2-1 & \mu^2+1\end{bmatrix},$$

The dimensionless constraints in terms of $\bar{N}_i$ are

(2.59)
$$\ddot{g}_1\left(\tilde{\mathbf{N}}\right)=1-\bar{N}_1+\bar{N}_2\le 0,$$

(2.60)
$$\ddot{g}_2\left(\tilde{\mathbf{N}}\right)=-1+\bar{N}_1-\bar{N}_2\le 0.$$

Graphically, the set of feasible normal forces is a ray with unit slope emanating from the point $(1,0)$ (or $(mg,0)$ in the dimensional problem). If $(1,0)$ is indeed a minimum, then condition (2.41) requires the existence of Lagrange multipliers $\lambda_1\ge 0$ and $z_2\ge 0$ such that

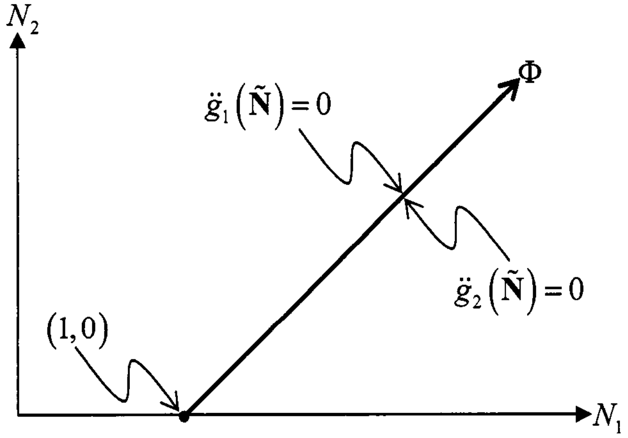

Figure 2.2. The feasible set of a point mass sliding between two planes.

(2.61) $$\partial_{\mathrm{N}}\varphi(1,0)=\begin{bmatrix}\mu^2+1\\ \mu^2-1\end{bmatrix}=\begin{bmatrix}1 & 0\\ -1 & 1\end{bmatrix}\begin{bmatrix}\lambda_1\\ z_2\end{bmatrix}.$$

Inverting this linear system gives $\lambda_1=1+\mu^2$ and $z_2=2\mu^2$, both of which are nonnegative for all values of μ. When $\mu=0$, this optimality condition holds at every feasible point; however, $(1,0)$ is the minimum Euclidean norm solution.

Also notice that since the normal forces applied by the upper and lower planes oppose one another, their sum is always polar to the tangent cone (the *x*-axis). In other words, D'Alembert's Principle is satisfied by every feasible point, regardless of the value of μ, and thus provides no help in determining the correct value of $\tilde{\mathbf{N}}$.

In the next section, we demonstrate how Gauss' Principle resolves the dynamic domains of Painleve's example without ambiguity, even when the LCP does not.

2.7 Painleve's Example with Arbitrary Control

Let us now apply the theoretical framework developed above to Painleve's classical example. Consider a rod of length 2*l*, mass *m*, and moment of inertia *J* sliding along a flat surface, as shown in Figure 2.3. Let $\mathbf{x}\equiv\left[r_x,r_y,\theta\right]^T\in\mathbb{R}^3$ denote the generalized position of the rod, where $\left[r_x,r_y\right]^T$ denotes its center of mass and $\theta\in(0,\pi)$ denotes its counter-clockwise angle with respect to the horizontal. We can define the generalized velocity to be $\mathbf{u}\equiv\dot{\mathbf{x}}=\left[\dot{r}_x,\dot{r}_y,\dot{\theta}\right]^T$, so that the kinematic map is simply the 3 by 3 identity:

(2.62) $$\mathbf{H}(\mathbf{x})=\mathbf{I}_{3\times 3}.$$

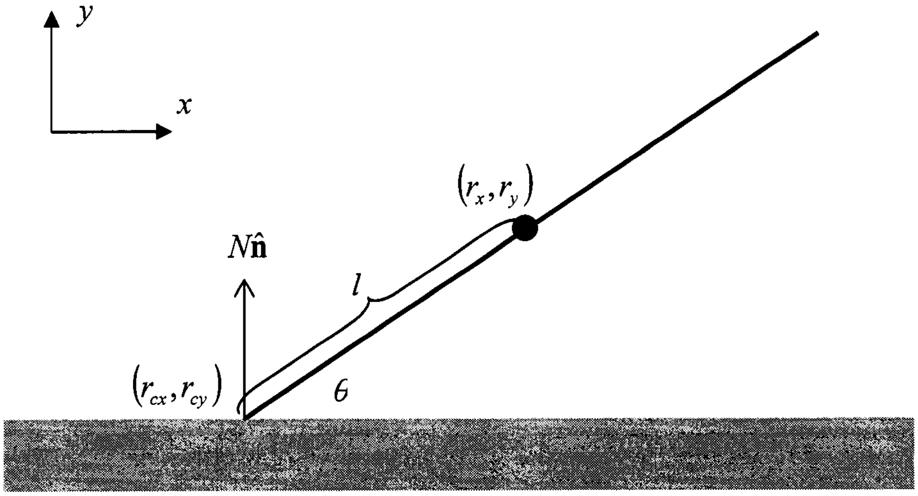

Figure 2.3. Painleve's example: a thin, rigid rod sliding on a flat surface.

The single point of contact $\left[r_{cx}, r_{cy}\right]^T$ in this system is related to the center of mass position by

(2.63)
$$r_{cy} = r_y - l\sin\theta,$$

and so the nonpenetration constraint for this contact point is simply

(2.64)
$$g_1(\mathbf{x}) = -r_{cy} = l\sin\theta - r_y \le 0.$$

If the rod can be initialized in a sliding state ($\dot{r}_{cx} \neq 0$) with $r_{cy} = 0$ and $\dot{r}_{cy} = 0$, then there exists a kinetic frictional force equal to

(2.65)
$$\mathbf{f} = -\mu N \operatorname{sgn} \dot{r}_{cx} \hat{\mathbf{i}}.$$

From the symmetry of this system, no generality is lost by choosing $\operatorname{sgn} \dot{r}_{cx} = -1$.

Let $\mathbf{F}(t,\mathbf{x},\mathbf{u}) \equiv \left[F_x, F_y, F_\theta\right]^T$ denote a generalized non-constraint force (the *control*) that is continuous in t and uniformly Lipschitz in $\mathbf{x}$ and $\mathbf{u}$. The equations of motion for this system are:

$$m\ddot{r}_x = \mu N + F_x, \tag{2.66}$$

$$m\ddot{r}_y = N + F_y, \tag{2.67}$$

$$J\ddot{\theta} = l\left(\mu\sin\theta - \cos\theta\right)N + F_\theta. \tag{2.68}$$

To simplify these equations, we substitute the dimensionless variables

$$\bar{t} \equiv t\sqrt{\frac{g}{l}}, \tag{2.69}$$

$$\bar{r}_x \equiv \frac{r_x}{l}, \tag{2.70}$$

$$\bar{r}_y \equiv \frac{r_y}{l}, \tag{2.71}$$

into (2.66) – (2.68) to get

$$\dot{\bar{u}}_1 = \mu\bar{N} + \bar{F}_x, \tag{2.72}$$

$$\dot{\bar{u}}_2 = \bar{N} + \bar{F}_y, \tag{2.73}$$

$$\varepsilon\dot{\bar{u}}_3 = \left(\mu\sin\theta - \cos\theta\right)\bar{N} + \bar{F}_\theta, \tag{2.74}$$

where $\bar{N} = \frac{1}{mg}N$ is the dimensionless normal force, and $\bar{F} = \left[\frac{1}{mg}F_x, \frac{1}{mg}F_y, \frac{1}{mgl}F_\theta\right]^T$ is the dimensionless non-constraint force. The parameter

$$\varepsilon = \frac{J}{ml^2} \tag{2.75}$$

is simply the ratio between the moment of inertia J and its supremum ml^2. A value of $\varepsilon = 0$ would correspond to a rod with all of its mass concentrated at its center, while a value of $\varepsilon = 1$ would correspond to a rod with all of its mass concentrated at its end points. Neither of these extremes is physically realizable so we must have $0 < \varepsilon < 1$.

From equations (2.72) – (2.74) it follows that the dimensionless mass matrix is

$$\bar{\mathbf{M}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \varepsilon \end{bmatrix}, \tag{2.76}$$

and the dimensionless wrench matrix is

$$\bar{\mathbf{W}} = \left[-\mu, -1, \cos\theta - \mu\sin\theta\right]. \tag{2.77}$$

The dimensionless nonpenetration constraint is

$$\bar{g}_1\left(\bar{\mathbf{x}}\right) = \bar{r}_{cy} = \sin\theta - \bar{r}_y \le 0, \tag{2.78}$$

and differentiating this twice with respect to $\bar{t}$ gives

$$\ddot{\bar{g}}_1\left(\bar{\mathbf{x}}\right) = \left[0, -1, \cos\theta\right]\dot{\bar{\mathbf{u}}} - \sin\theta\dot{\theta}^2 \le 0. \tag{2.79}$$

Following the procedure outlined in Section 2.5, we can express this in terms of $\bar{N}$:

$$\ddot{\bar{g}}_1\left(\bar{N}\right) = b - A\bar{N} \le 0, \tag{2.80}$$

where[5]

$$A = \tfrac{1}{2}\left(1 + 2\varepsilon + \cos 2\theta - \mu \sin 2\theta\right), \tag{2.81}$$

$$b = \bar{F}_\theta \cos\theta - \varepsilon\left(\bar{F}_y + \sin\theta\dot{\theta}^2\right). \tag{2.82}$$

The objective function for this system is

$$\varphi\left(\bar{N}\right) = \frac{1}{2}\bar{N}^2\left(\mathbf{W}\bar{\mathbf{M}}^{-1}\mathbf{W}^T\right) = \frac{1}{2}\bar{N}^2\left(1 + \mu^2 + \frac{\left(\cos\theta + \mu\sin\theta\right)^2}{\varepsilon}\right). \tag{2.83}$$

To resolve $\bar{N}$, we minimize $\varphi\left(\bar{N}\right)$ subject to inequality (2.80) and the range constraint $\bar{N} \geq 0$, and the optimal solution of this problem depends upon the signs of A and b.

If both $A > 0$ and $b > 0$, then (2.80) is equivalent to

$$\bar{N} \geq \frac{b}{A} > 0. \tag{2.84}$$

Since this is the active constraint, the optimal solution is:

$$\bar{N}_0 = \frac{b}{A} = \frac{\bar{F}_\theta \cos\theta - \varepsilon\left(\bar{F}_y + \sin\theta\dot{\theta}^2\right)}{\tfrac{1}{2}\left(1 + 2\varepsilon + \cos 2\theta - \mu\sin 2\theta\right)}, \tag{2.85}$$

with $\ddot{g}_1 = 0$. The feasible set and objective function for this case are shown in Figure 2.4.

When $A > 0$ and $b < 0$, inequality (2.80) reduces to

$$\bar{N} \geq \frac{b}{A}, \tag{2.86}$$

---

[5] These follow from the identities $\cos\theta\sin\theta = \tfrac{1}{2}\sin 2\theta$, and $\cos^2\theta = \tfrac{1}{2} + \tfrac{1}{2}\cos 2\theta$.

with $\frac{b}{A} < 0$. In this case, only the range constraint is active, and so the optimal solution is $\bar{N} = 0$, as illustrated in Figure 2.5. Substituting this result back into (2.80) gives $\ddot{g}_1 < 0$, which implies vanishing contact.

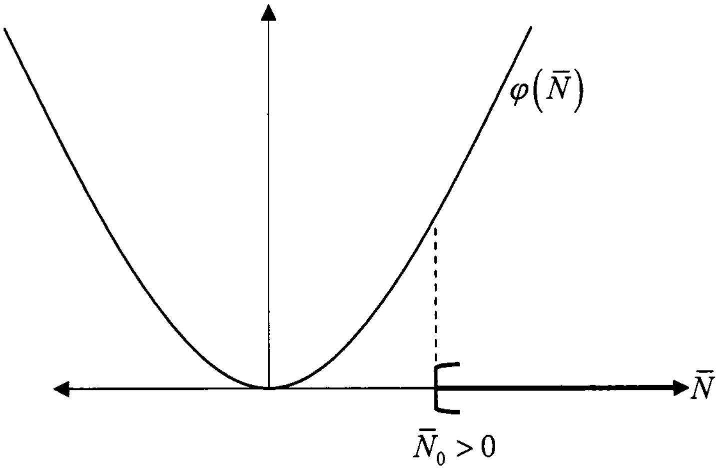

Figure 2.4. The objective function and feasible set for a sliding rod with nonvanishing contact.

From equation (2.81) we can see that for any ε we may choose μ sufficiently large such that there exist two distinct angles $0 < \theta_{c,1} < \theta_{c,2} < \frac{\pi}{2}$ at which $A = 0$, and for which[6] $A < 0$ for all $\theta \in (\theta_{c,1}, \theta_{c,2})$. Therefore we must also consider the behavior of this system when $A = 0$ and when $A < 0$.

If both $A < 0$ and $b < 0$, then (2.80) is equivalent to

---

[6] This sign change is independent of **F** and will not occur for $\mu = 0$.

(2.87)
$$\bar{N} \le \frac{b}{A},$$

with $\frac{b}{A} > 0$. We can easily see from Figure 2.6 below that only the range constraint is active, so the optimal solution is $\bar{N} = 0$. Substituting this back into (2.80) gives $\ddot{\bar{g}}_1 < 0$, again a vanishing contact.

In the case that $A = 0$ and $b < 0$, we have $\ddot{\bar{g}}_1(\bar{N}) = b < 0$ (vanishing contact). In other words, (2.80) is trivially satisfied, and again only the range constraint is active so that $\bar{N} = 0$. This case is identical to that shown in Figure 2.5:

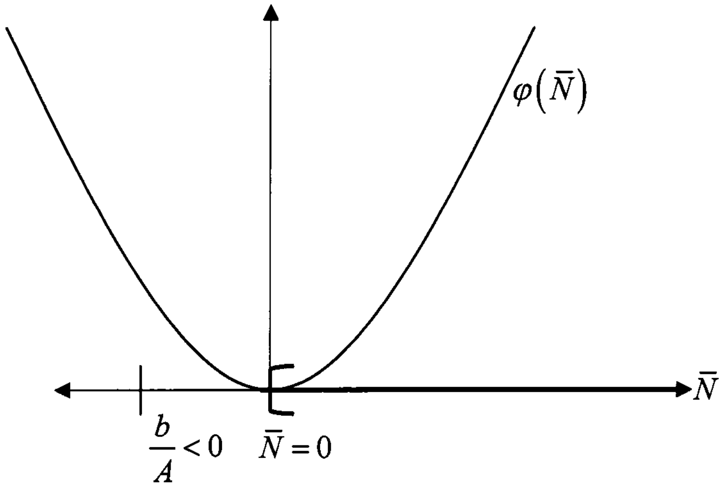

Figure 2.5. The objective function and feasible set for a sliding rod with vanishing contact.

The above analyses establish that contact vanishes whenever $b < 0$. This occurs whenever

(2.88)
$$\left|\dot{\theta}\right| > \sqrt{\frac{\frac{1}{\varepsilon}\bar{F}_\theta \cos\theta - \bar{F}_y}{\sin\theta}}$$

is satisfied. Therefore, in order to have any nonvanishing states for a particular value of $\bar{\mathbf{F}}$, we must have $\frac{1}{\varepsilon}\bar{F}_\theta \cos\theta - \bar{F}_y > 0$ for some values of θ. Let $\theta_v \in (0,\pi)$ for which $\theta_v = \cos^{-1}\left(\varepsilon \frac{\bar{F}_y}{\bar{F}_\theta}\right)$. If $\bar{F}_\theta > 0$, then $\theta < \theta_v$ implies that $b > 0$.

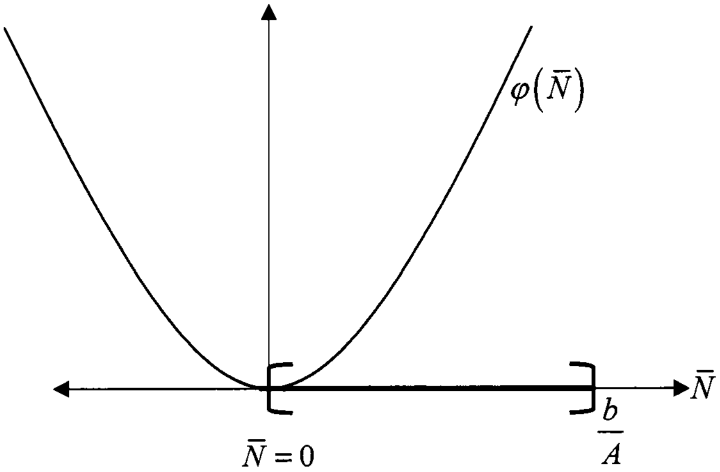

Figure 2.6. Another example of vanishing contact.

Finally, consider the initial conditions for which $A < 0$ and $b > 0$. From (2.80) we get the constraint

(2.89)
$$\bar{N} \le \frac{b}{A} < 0,$$

which combined with the range constraint $\bar{N} \ge 0$ gives an empty feasible set. These are the *inconsistent* states (the states for which no sliding solution exists).

We have seen that for all $\theta \in (\theta_{c,1}, \theta_{c,2})$, A is negative, and therefore we should not expect a unique solution for the contact LCP, or any solution at all for that matter. Indeed, when $A < 0$ and $b < 0$, this LCP has exactly two isolated solutions: (i) $\left\{\bar{N} = -\frac{b}{A}, \ddot{g}_1 = 0\right\}$ and (ii) $\{\bar{N} = 0, \ddot{g}_1 = b\}$, with solution (ii) corresponding to the optimal solution using the Least Constraint formulation.

The three primary domains of behavior for this system are mapped out in state space below in Figure 2.7. The generalized control $\bar{\mathbf{F}}$ has been chosen so that $\theta_{c,2} < \theta_v$, the crossing point for the two curves $\dot{\theta} = \sqrt{\frac{\frac{1}{\varepsilon}\bar{F}_\theta \cos\theta - \bar{F}_y}{\sin\theta}}$ and $\dot{\theta} = -\sqrt{\frac{\frac{1}{\varepsilon}\bar{F}_\theta \cos\theta - \bar{F}_y}{\sin\theta}}$. The interior of region $\mathcal{A}$ corresponds to states with nonvanishing contact in which the normal force is given by equation (2.85), the interior of region $\mathcal{B}$ corresponds to states with vanishing contact, and the interior of region $\mathcal{C}$ corresponds to states for which no solution exists. Any transition from the boundary of $\mathcal{C}$ into its interior would result in an instantaneous transition to zero relative velocity at the contact point (i.e., an *impulse*).

On the boundaries between and $\mathcal{A}$ and $\mathcal{B}$ and between $\mathcal{B}$ and $\mathcal{C}$, we have $b = 0$, and the optimal solution is $\bar{N} = 0$, regardless of the sign of A. Substituting these back into (2.80) shows that $\ddot{g}_1 = 0$. Moreover, at the triple points between $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, we have $A = 0$ and $b = 0$, again giving us $\ddot{g}_1 = 0$ and $\bar{N} = 0$. While these three solutions trivially satisfy the complementarity conditions, it is unclear whether the contact is vanishing or not. To answer this question, we cannot simply look at the direction of the differential vector field defined by $\left[\dot{\theta}, \ddot{\theta}\right]^T$, as was done in [26], since this approach is valid only when $\bar{\mathbf{F}}$ does not vary in time. With a time-dependent control, the boundaries between these regions are changing as $\overline{\mathbf{F}}$ changes, so we must look at the sign of $\dddot{g}_1$.

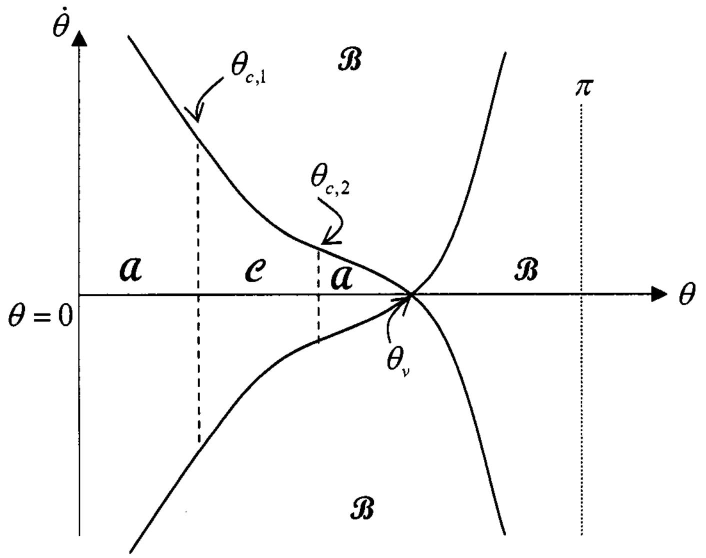

Figure 2.7. The domains of behavior for the generalized Painleve' system.

2.8 Painleve's Classical Example: $\mathbf{F}_{nc} = -m\mathbf{g}$

In order to characterize Painleve's original example, we simply let $\overline{F}_x = \overline{F}_\theta = 0$ and $\overline{F}_y = -1$ (dimensionless gravitational force). The domains of behavior for this particular example are mapped out in Figure 2.8.

Interestingly, Song and others have shown that certain LCP solutions of Painleve's example are stable while others or not with respect to singular perturbation analysis of a particular compliant model [62]. In fact, their "stable" solutions are precisely the solutions given by Gauss' Principle. These results suggest that multiplicity of solutions in kinetic frictional contact problems is simply an artifact of the LCP formulation, with no physical basis.

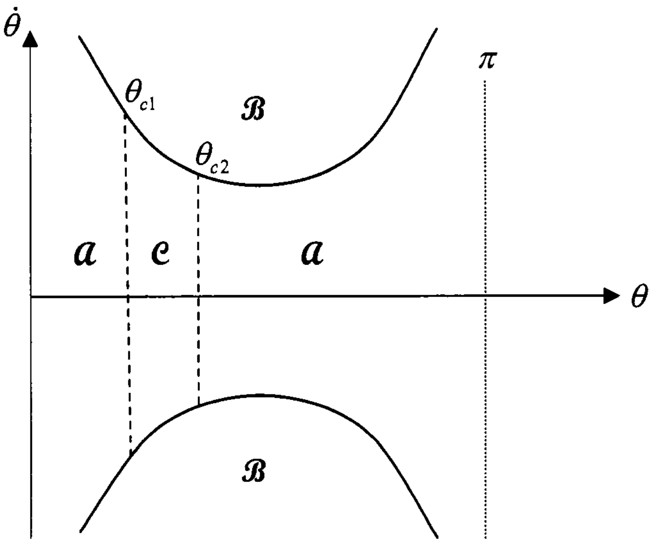

Figure 2.8. The domains of behavior for Painleve's classical example.

In the next section we analyze the classical example of a rectangular block sliding on a flat surface.

2.9 Two-Point Analysis of a Sliding Block

The example of a sliding block is often used to introduce the concept of Coulomb friction in elementary mechanics textbooks. The classical analysis of this system treats the block as if it were a point mass, with gravity pulling downward, a single normal force pointing upward, and a single frictional force opposing the motion, as shown in Figure 2.9. Since the acceleration of the block in the y-direction is zero, the normal force and the gravitational force must be equal, and therefore the equations of motion for this system are simply

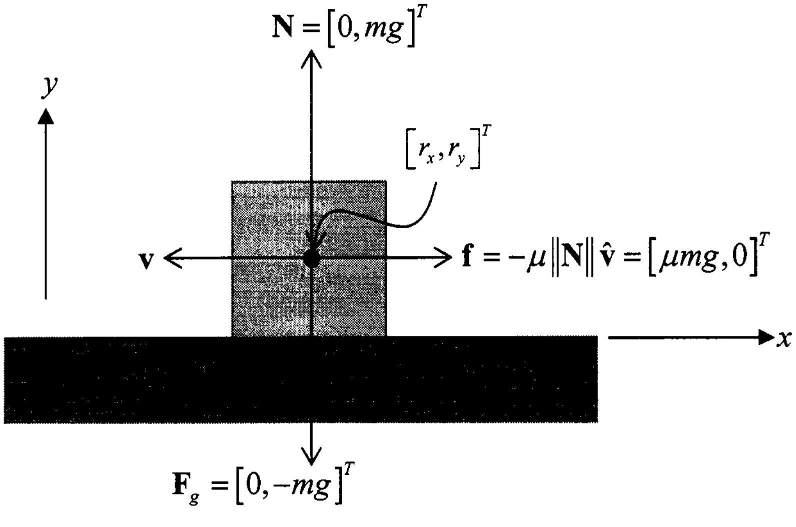

Figure 2.9. A sliding block treated as a point mass.

$$m\ddot{r}_x = \mu mg, \tag{2.90}$$

$$m\ddot{r}_y = N - mg = 0. \tag{2.91}$$

However, such a simple analysis ignores two fundamental aspects of this system: (1) the forces over the contact interface may not be evenly distributed, and (2) the block can rotate allowing one contact to vanish.

A more physically realistic model approximates the continuous contact interface with two points: the first one at the lower-left corner of the block, and the second one at the lower-right corner. In order to predict the rotational behavior of the block, its dimensions must be taken into account, with w denoting its half-width and h denoting its half-height. This improved model is illustrated in Figure 2.10, below.

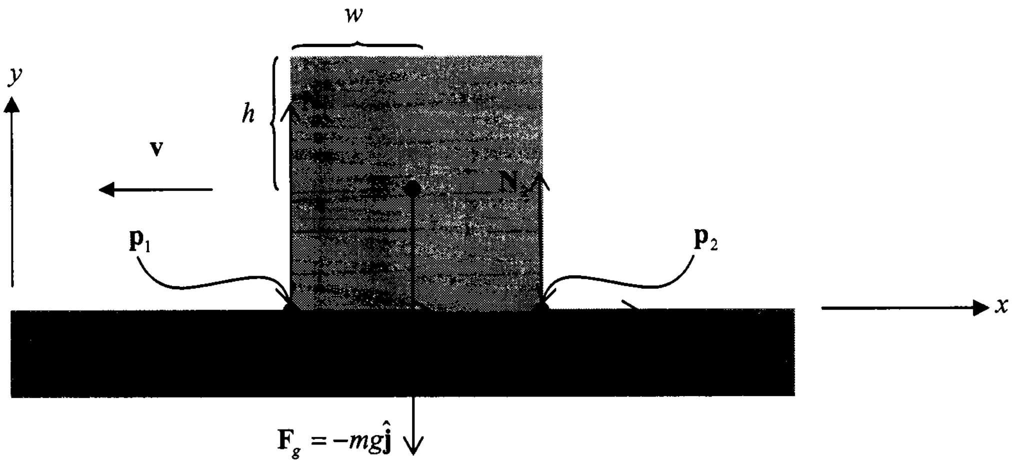

Figure 2.10. A sliding block with two points of contact.

In this case, the equations of motion are

$$m\ddot{r}_x = \mu N_1 + \mu N_2, \tag{2.92}$$

$$m\ddot{r}_y = -mg + N_1 + N_2, \tag{2.93}$$

$$J\ddot{\theta} = -\left(w - \mu h\right) N_1 + \left(w + \mu h\right) N_2. \tag{2.94}$$

As we have done previously, we will simplify the analysis of this system by introducing the dimensionless variables

$$l \equiv \sqrt{w^2 + h^2}\ , \tag{2.95}$$

$$\alpha \equiv {}^{w}\!/_{l}\,, \tag{2.96}$$

$$\beta \equiv {}^{h}\!/_{l}\,. \tag{2.97}$$

By definition, $\alpha, \beta \in \left(0,1\right)$ and $\alpha^2 + \beta^2 = 1$. Using these along with the dimensionless time defined in equation (2.69) and the dimensionless moment of inertia (2.75), we get the dimensionless equations of motion

$$\ddot{\bar{r}}_x = \mu \bar{N}_1 + \mu \bar{N}_2, \tag{2.98}$$

$$\ddot{\bar{r}}_y = -1 + \bar{N}_1 + \bar{N}_2, \tag{2.99}$$

$$\varepsilon\ddot{\bar{\theta}} = -(\alpha - \mu\beta)\bar{N}_1 + (\alpha + \mu\beta)\bar{N}_2, \tag{2.100}$$

where $\bar{N}_i \equiv N_i / mg$ for $i = 1,2$. The wrench matrix at this instant is therefore

$$\tilde{\mathbf{W}} = \begin{bmatrix} -\mu & -1 & \alpha - \mu\beta \\ -\mu & -1 & -\alpha - \mu\beta \end{bmatrix}, \tag{2.101}$$

the rows of which are linearly independent for $\alpha > 0$. Thus

$$\tilde{\mathbf{W}}\mathbf{M}^{-1}\tilde{\mathbf{W}}^T = \frac{1}{\varepsilon}\begin{bmatrix} \varepsilon\left(1+\mu^2\right)+\left(\alpha-\mu\beta\right)^2 & \varepsilon\left(1+\mu^2\right)-\alpha^2+\mu^2\beta^2 \\ \varepsilon\left(1+\mu^2\right)-\alpha^2+\mu^2\beta^2 & \varepsilon\left(1+\mu^2\right)+\left(\alpha+\mu\beta\right)^2 \end{bmatrix} \tag{2.102}$$

is positive definite. The dimensionless nonpenetration constraints for this system are

$$g_1(\bar{\mathbf{x}}) = -\bar{r}_y + \alpha\sin\theta + \beta\cos\theta \le 0, \tag{2.103}$$

$$g_2(\bar{\mathbf{x}}) = -\bar{r}_y - \alpha\sin\theta + \beta\cos\theta \le 0. \tag{2.104}$$

Differentiating these with respect to $\bar{\mathbf{x}}$, we get

$$\tilde{\mathbf{G}}_{\bar{\mathbf{x}}}(\bar{\mathbf{x}}) = \begin{bmatrix} 0 & -1 & \alpha\cos\theta - \beta\sin\theta \\ 0 & -1 & -\alpha\cos\theta - \beta\sin\theta \end{bmatrix}, \tag{2.105}$$

which leads to

$$\frac{d}{dt}\left(\tilde{\mathbf{G}}_{\bar{\mathbf{x}}}\mathbf{H}(\bar{\mathbf{x}})\right) = \begin{bmatrix} 0 & 0 & -(\alpha\sin\theta + \beta\cos\theta)\dot{\bar{\theta}} \\ 0 & 0 & (\alpha\sin\theta - \beta\cos\theta)\dot{\bar{\theta}} \end{bmatrix}, \tag{2.106}$$

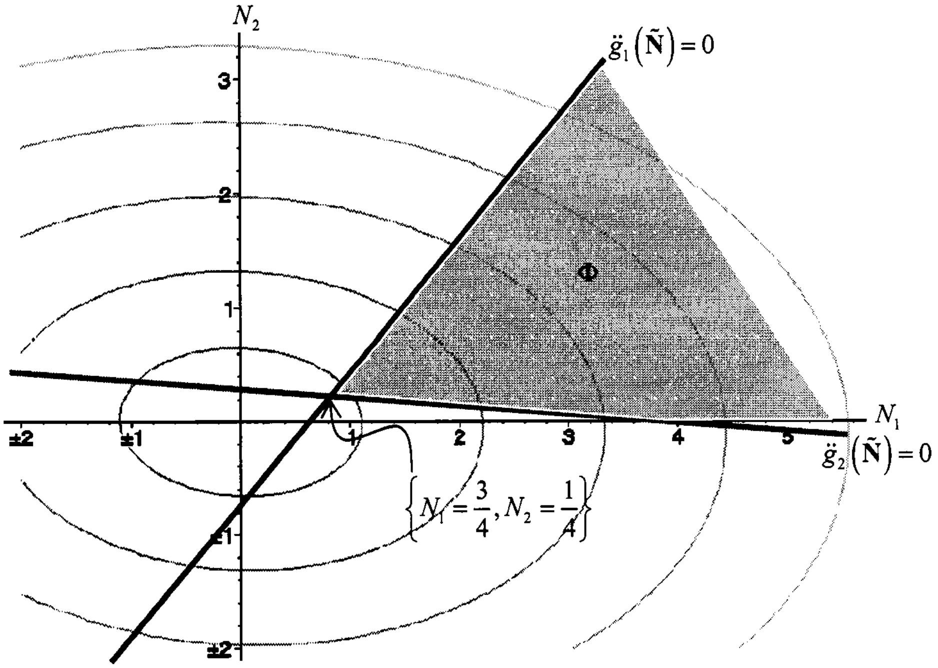

Figure 2.11. The objective function and feasible set for a sliding block, $\mu = 1/2$.

When the block is in contact with the plane, we have $\theta = \dot{\theta} = 0$, and so

$$\tilde{\mathbf{A}} = \frac{1}{\varepsilon}\begin{bmatrix} \varepsilon + \alpha(\alpha - \mu\beta) & \varepsilon - \alpha(\alpha + \mu\beta) \\ \varepsilon - \alpha(\alpha - \mu\beta) & \varepsilon + \alpha(\alpha + \mu\beta) \end{bmatrix}, \tag{2.107}$$

$$\tilde{\mathbf{b}} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T. \tag{2.108}$$

Let us now explore the advantages of this improved model by examining three key behaviors of this system: persistent sliding contact, grazing contact, and vanishing contact. For a square block with uniform density, we have

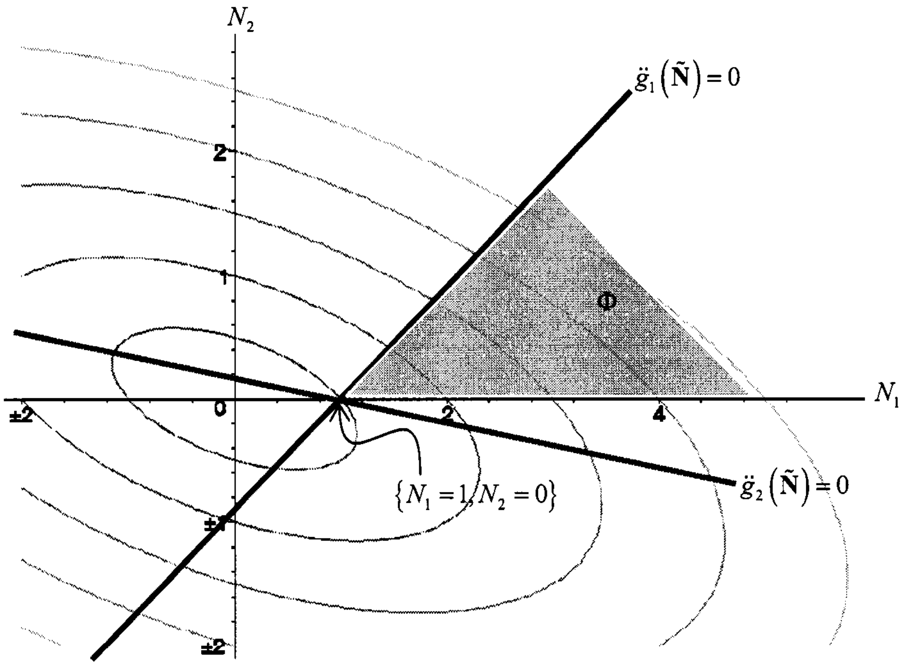

Figure 2.12. The objective function and feasible set for a sliding block, $\mu = 1$.

$\alpha = \beta = \sqrt{2}/2$ and $\varepsilon = 1/3$. When $\mu = 1/2$, the nonpenetration constraints in terms of $\tilde{\mathbf{N}}$ are

$$\ddot{g}_1(\tilde{\mathbf{N}}) = 1 - \frac{7}{4}N_1 + \frac{5}{4}N_2 \leq 0, \tag{2.109}$$

$$\ddot{g}_2(\tilde{\mathbf{N}}) = 1 - \frac{1}{4}N_1 - \frac{13}{4}N_2 \leq 0. \tag{2.110}$$

The objective function and feasible set for this configuration are shown in Figure 2.11, and it is clear that the optimal point lies at $\left\{N_1 = \frac{3}{4}, N_2 = \frac{1}{4}\right\}$. We now see that even with a fairly low friction coefficient, the predicted contact forces on the left side of the block are *3 times greater* than those on the right side of the block. This prediction is consistent with the uneven wear patterns observed in disk brake pads.

When $\mu = 1$, the nonpenetration constraints become

(2.111) $$\ddot{g}_1\left(\tilde{\mathbf{N}}\right) = 1 - N_1 + 2N_2 \leq 0,$$

(2.112) $$\ddot{g}_2\left(\tilde{\mathbf{N}}\right) = 1 - N_1 - 4N_2 \leq 0.$$

The objective function and feasible set for this friction coefficient are shown in Figure 2.12. In this case, the optimal solution is $\{N_1 = 1, N_2 = 0\}$, exactly where the lines $\ddot{g}_1\left(\tilde{\mathbf{N}}\right) = 0$ and $\ddot{g}_2\left(\tilde{\mathbf{N}}\right) = 0$ intersect the N_1-axis. This is an example of grazing contact, in which $N_2 = 0$ and $\ddot{g}_2\left(\tilde{\mathbf{N}}\right) = 0$ simultaneously.

Finally, for $\mu = 3/2$, the nonpenetration constraints become

(2.113) $$\ddot{g}_1\left(\tilde{\mathbf{N}}\right) = 1 - \frac{1}{4}N_1 + \frac{11}{4}N_2 \leq 0,$$

(2.114) $$\ddot{g}_2\left(\tilde{\mathbf{N}}\right) = 1 - N_1 - 4N_2 \leq 0.$$

The objective function and feasible set are shown in Figure 2.13. The optimal solution in this case is $\{N_1 = 4, N_2 = 0\}$, at which $\ddot{g}_2\left(\tilde{\mathbf{N}}\right) < 0$. So for this friction coefficient, the lower-right corner of the block lifts off the plane while the lower-left corner experiences a total contact force (normal + friction) approximately 7 times its weight concentrated at $\mathbf{p}_1$.

In the next chapter, we will show how this optimality formulation can be used in conjunction with an explicit Runge-Kutta solver of arbitrary order to integrate the differential-algebraic equations of motion.

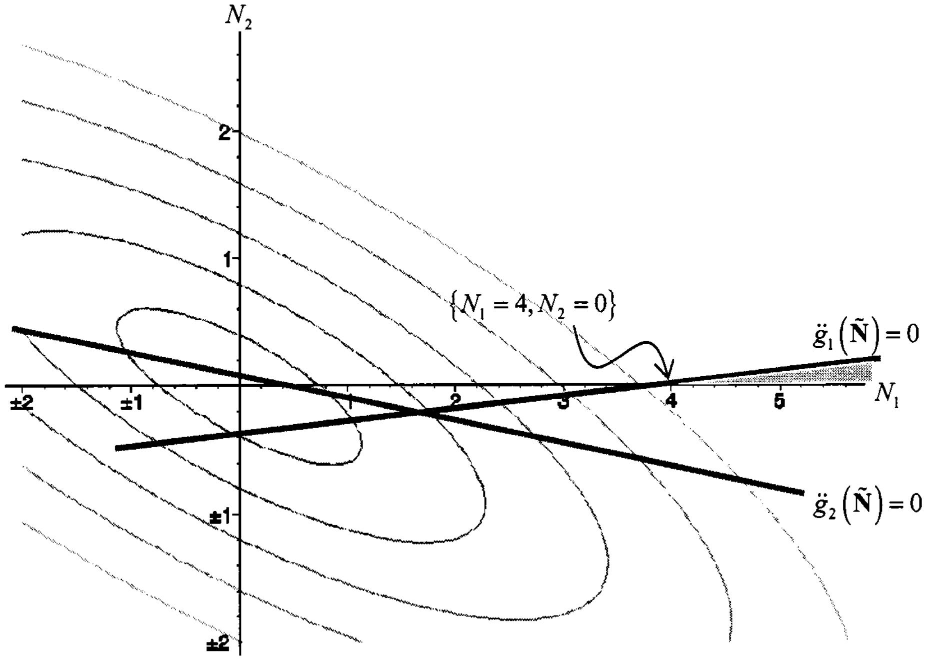

Figure 2.13. The objective function and feasible set for a sliding block, $\mu = 3/2$.

Chapter 3 : Numerical Solution

The previous chapters have mainly dealt with theoretical aspects of mechanical systems with friction. In this chapter, we describe how Gauss' Principle can be used to easily construct high-order time stepping schemes for the mechanical systems of interest. Since we are interested in constrained mechanical systems, we describe a simple projection procedure that ensures the solution satisifies the necessary constraints at the velocity and position-level. In Chapter 4, a 5th order implementation of this technique is used to accurately predict the oscillations of an unlubricated pendulum oscillating under gravity.

3.1 First-order Discretization

The optimization procedure described in Section 2.5 can be used to compute the generalized normal force $\tilde{\mathbf{N}}_0$, which then gives us the DAE in (2.30) – (2.34). By discretizing (2.30) and (2.31) we can compute first-order approximations of the solution at time $t_0 + h$:

$$\tilde{\mathbf{x}}_1 = \mathbf{x}_0 + h\mathbf{H}(\mathbf{x}_0)\mathbf{u}_0 \approx \mathbf{x}(t_0 + h), \tag{3.1}$$

$$\tilde{\mathbf{u}}_1 = \mathbf{u}_0 + h\mathbf{M}_0^{-1}\left[\mathbf{F}_{nc}(t_0, \mathbf{x}_0, \mathbf{u}_0) - \tilde{\mathbf{W}}_0^T \tilde{\mathbf{N}}_0\right] \approx \mathbf{u}(t_0 + h). \tag{3.2}$$

In addition to (2.32) the numerical solution must also satisfy the spatial and velocity constraints:

$$\mathbf{G}(\mathbf{x}_1) = 0, \tag{3.3}$$

$$\dot{\mathbf{G}}(\mathbf{x}_1) = \mathbf{G}_{\mathbf{x}}\mathbf{H}(\mathbf{x}_1)\mathbf{u}_1 = 0. \tag{3.4}$$

It is unlikely, however, that this first-order procedure will produce an $(\tilde{\mathbf{x}}_1, \tilde{\mathbf{u}}_1)$ pair which will satisfy (3.3) – (3.4), and if this naïve scheme is repeated over many iterations without correcting the solution, error may accumulate to a point where the solution no longer makes physical sense. This numerical phenomenon is known as *drift* [10, 15, 36] and is simply due to the discretization of the continuous system.

3.2 Constraint Stabilization

If $\mathbf{G}$ is smooth, then it is usually possible to find another solution $\mathbf{x}_1$ nearby $\tilde{\mathbf{x}}_1$ which satisfies (3.3), a technique known as *projection*. Hairer, et. al., [36] propose a projection scheme based on [46]. The idea is to solve:

$$\mathbf{x}_1 - \tilde{\mathbf{x}}_1 + \mathbf{G}_{\mathbf{x}}^T(\tilde{\mathbf{x}}_1)\boldsymbol{\eta} = 0, \tag{3.5}$$

$$\mathbf{G}(\mathbf{x}_1) = 0, \tag{3.6}$$

for $\mathbf{x}_1$ and $\boldsymbol{\eta}$, which gives $n_x + m$ unknowns at each Newton step. In solving this nonlinear system, we find a nearby point $\mathbf{x}_1$ which lies along the line $\mathbf{x}_1 = \tilde{\mathbf{x}}_1 - \mathbf{G}_{\mathbf{x}}^T(\tilde{\mathbf{x}}_1)\boldsymbol{\eta}$ and satisfies $\mathbf{G}(\mathbf{x}_1) = 0$.

Recall that for a smooth function $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n$, we have

$$\mathbf{F}(\mathbf{z}) = \mathbf{F}(\mathbf{z}^\nu) + \mathbf{F}_{\mathbf{z}}(\mathbf{z}^\nu)(\mathbf{z} - \mathbf{z}^\nu) + o(\|\mathbf{z} - \mathbf{z}^\nu\|) \tag{3.7}$$

in a neighborhood of the point $\mathbf{z}^\nu$. If $\mathbf{F}_{\mathbf{z}}(\mathbf{z}^\nu)$ is nonsingular, then there exists a unique value $\mathbf{z}^{\nu+1}$ for which

$$\mathbf{F}_{\mathbf{z}}(\mathbf{z}^\nu)\mathbf{z}^{\nu+1} = \mathbf{F}_{\mathbf{z}}(\mathbf{z}^\nu)\mathbf{z}^\nu - \mathbf{F}(\mathbf{z}^\nu). \tag{3.8}$$

Newton's method produces a sequence of approximations $\{\mathbf{z}^\nu\}$ by successively solving (3.8) for the next iterate $\mathbf{z}^{\nu+1}$. If a root $\mathbf{z}$ of $\mathbf{F}$ exists and the initial estimate

$\mathbf{z}^0$ is sufficiently close to $\mathbf{z}$, then the sequence $\{\mathbf{z}^\nu\}$ converges to $\mathbf{z}$. If $\mathbf{F}_\mathbf{z}^{-1}(\mathbf{z}^\nu)$ is bounded in this neighborhood, then the rate of convergence is *Q-linear*, and if $\mathbf{F}_\mathbf{z}$ is Lipschitz continuous in this neighborhood, then the rate of converge is *Q-quadratic* (see [55], sections 2.2 and 11.1).

However, if $\mathbf{F}_\mathbf{z}$ is *singular* but the right-hand side of (3.8) is in the range of $\mathbf{F}_\mathbf{z}$ in a neighborhood of $\mathbf{z}$, then equation (3.8) will have a subspace of solutions. In this case, we use the *Moore-Penrose generalized inverse* [67] of $\mathbf{F}_\mathbf{z}$ to get the recursion formula:

$$\mathbf{z}^{\nu+1} = \mathbf{F}_\mathbf{z}(\mathbf{z}^\nu)^\dagger\left[\mathbf{F}_\mathbf{z}(\mathbf{z}^\nu)\mathbf{z}^\nu - \mathbf{F}(\mathbf{z}^\nu)\right]. \tag{3.9}$$

This equation gives us the $\mathbf{z}^{\nu+1}$ of minimum Euclidean norm which solves (3.8), and the convergence properties remain the same.

The Newton recursion that follows directly from (3.5) – (3.6) is

$$\begin{bmatrix} \mathbf{I} & \mathbf{G}_\mathbf{x}^T(\tilde{\mathbf{x}}_1) \\ \mathbf{G}_\mathbf{x}(\mathbf{x}^\nu) & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{x}^{\nu+1} \\ \boldsymbol{\eta}^{\nu+1} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{x}}_1 \\ \mathbf{G}_\mathbf{x}(\mathbf{x}^\nu)\mathbf{x}^\nu - \mathbf{G}(\mathbf{x}^\nu) \end{bmatrix}. \tag{3.10}$$

In our code, however, we use the slightly different formula:

$$\begin{bmatrix} \mathbf{I} & \mathbf{G}_\mathbf{x}^T(\mathbf{x}^\nu) \\ \mathbf{G}_\mathbf{x}(\mathbf{x}^\nu) & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{x}^{\nu+1} \\ \boldsymbol{\eta}^{\nu+1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^\nu \\ \mathbf{G}_\mathbf{x}(\mathbf{x}^\nu)\mathbf{x}^\nu - \mathbf{G}(\mathbf{x}^\nu) \end{bmatrix}, \tag{3.11}$$

which has the advantage of being symmetric so that efficient iterative linear solvers such as GMRES can be used [32, 41]. In addition, $\mathbf{G}_\mathbf{x}$ is updated at each step, giving slightly better convergence properties when $\mathbf{G}_\mathbf{x}$ changes rapidly in the neighborhood of the solution.

It should be emphasized that formulas (3.10) and (3.11) will in general *not* converge to the same result. This follows from the fact that the upper triangular part of the left-hand side of (3.10) is evaluated only once at $\tilde{\mathbf{x}}_1$, whereas (3.11) updates the left-hand side at each $\mathbf{x}^{\nu}$. At a stationary point of (3.10), may have $\eta > 0$; whereas, a stationary point of (3.11) has $\eta = 0$.

If $\mathbf{G}_\mathbf{x}$ has full row rank in this neighborhood, then (3.11) has a unique solution; however, in many practical situations (including our experiment) we have $m > n_x$. In this case $\mathbf{G}_\mathbf{x}$ is rank-deficient, and (3.11) is solved using the Moore-Penrose generalized inverse. Once the constraint error $\left\|\mathbf{G}\left(\mathbf{x}_{\nu+1}\right)\right\|_\infty$ falls below an acceptable tolerance, the recursion is terminated, and $\mathbf{x}_1 \equiv \mathbf{x}_{\nu+1}$ is accepted.

When a satisfactory $\mathbf{x}_1$ has been found, we can find a velocity estimate which satisfies (3.4) by solving the symmetric linear system

(3.12)
$$\begin{bmatrix} \mathbf{I} & \mathbf{H}^T\mathbf{G}_\mathbf{x}^T(\mathbf{x}_1) \\ \mathbf{G}_\mathbf{x}\mathbf{H}(\mathbf{x}_1) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{u}}_1 \\ 0 \end{bmatrix}$$

for $\mathbf{u}_1$ and $\boldsymbol{\lambda}$. Since equation (3.4) requires that $\mathbf{u}$ be orthogonal to the *rows* of $\mathbf{G}_\mathbf{x}\mathbf{H}$, solving (3.12) simply removes any components of $\mathbf{u}_1$ which lie in the subspace spanned by the rows of $\mathbf{G}_\mathbf{x}\mathbf{H}$.

3.3 Runge-Kutta Methods and Time-Stepping

The scheme described above can be easily adapted to achieve higher orders of accuracy. For an initial value problem

(3.13)
$$\dot{\mathbf{y}} = \mathbf{F}(t, \mathbf{y}),$$

(3.14)
$$\mathbf{y}(t_0) = \mathbf{y}_0,$$

the general form of an explicit s-stage Runge-Kutta method is

$$\mathbf{k}_i = \mathbf{F}\left(t_0 + a_i h, \mathbf{y}_0 + h\textstyle\sum_{j=1}^{i-1} b_{i,j}\mathbf{k}_j\right), \quad i = 1, \ldots, s \tag{3.15}$$

$$\mathbf{y}_1 = \mathbf{y}_0 + h\sum_{i=1}^{s} c_i \mathbf{k}_i, \tag{3.16}$$

where the constants a_i, b_i, and c_i have been chosen so that the *local truncation error* (LTE) satisfies

$$\left\|\mathbf{y}_1 - \mathbf{y}(t_0 + h)\right\| < Ch^{p+1} \tag{3.17}$$

as $h \to 0$ for fixed constants C and p. The value p is called the *order* of the method.

Adapting (3.15) – (3.16) to solve (2.39) – (2.31), at each stage we evaluate $\mathbf{H}_{i-1}$, $\mathbf{M}_{i-1}$, $\mathbf{F}_{i-1}$ and $\tilde{\mathbf{W}}_{i-1}$ at the values

$$t_{i-1} = t_0 + a_i h, \tag{3.18}$$

$$\mathbf{x}_{i-1} = \mathbf{x}_0 + h\sum_{j=1}^{i-1} b_{i,j}\dot{\mathbf{x}}_j, \tag{3.19}$$

$$\mathbf{u}_{i-1} = \mathbf{u}_0 + h\sum_{j=1}^{i-1} b_{i,j}\dot{\mathbf{u}}_j, \tag{3.20}$$

which have been projected back onto the constraint manifold. Once the optimization procedure has been solved for $\tilde{\mathbf{N}}_{i-1}$, the derivatives

$$\dot{\mathbf{x}}_i = \mathbf{H}_{i-1}\mathbf{u}_{i-1}, \tag{3.21}$$

$$\dot{\mathbf{u}}_i = \mathbf{M}_{i-1}^{-1}\left(\mathbf{F}_{i-1} - \tilde{\mathbf{W}}_{i-1}^{T}\tilde{\mathbf{N}}_{i-1}\right), \tag{3.22}$$

can be used to compute the intermediate values for the following stage. Finally after all stages have been computed, a p^{th}-order estimate of the solution is given by

(3.23) $$\tilde{\mathbf{x}}_1 = \mathbf{x}_0 + h\sum_{i=1}^{s} c_i \dot{\mathbf{x}}_i \,,$$

(3.24) $$\tilde{\mathbf{u}}_1 = \mathbf{u}_0 + h\sum_{i=1}^{s} c_i \dot{\mathbf{u}}_i \,.$$

These values are then projected onto the constraint manifold defined by (3.3) and (3.4).

When integrating the initial value problem (3.13) – (3.14), we would like to ensure that h is small enough so that the LTE stays below some prescribed value. On the other hand, we would also like to increase h if the LTE is very small so that computational effort is not wasted when the solution is not changing substantially. An s-stage *embedded* Runge-Kutta method is one which has two sets of constants: one set $\{c_i\}$ which provides a higher-order order estimate $\mathbf{y}_1$, and another set $\{c_i^*\}$ which provides a lower-order estimate $\mathbf{y}_1^*$, all for only s evaluations of $\mathbf{F}$. Such formulas can be used to estimate the LTE and adjust h based on user-specified error tolerances.

3.4 Statement of the Algorithm

The following pseudo-code summarizes the main loop of the time-stepping scheme, where T denotes the duration of the simulation:

```
//starting time
$t = 0$;

while($t < T$)
{
    while($err > tol$)
    {
```

```
//compute the derivatives for each stage
for( $i=1$ ; $i \le s$ ; $i$++ )
    {
        $d\mathbf{x}_{i-1} = h\sum_{j=1}^{i-1} b_{i,j}\dot{\mathbf{x}}_j$
        $d\mathbf{u}_{i-1} = h\sum_{j=1}^{i-1} b_{i,j}\dot{\mathbf{u}}_j$
        $[\dot{\mathbf{x}}_i, \dot{\mathbf{u}}_i] = RK\_stage(\mathbf{x}_0 + d\mathbf{x}_{i-1}, \mathbf{u}_0 + d\mathbf{u}_{i-1})$
    }

    //compute a high and a low-order
    //estimate of the solution
    $\mathbf{x}^* = \mathbf{x}_0 + h\sum_{j=1}^{6} c_j^*\dot{\mathbf{x}}_j$
    $\mathbf{u}^* = \mathbf{u}_0 + h\sum_{j=1}^{6} c_j^*\dot{\mathbf{u}}_j$
    $\mathbf{x} = \mathbf{x}_0 + h\sum_{j=1}^{6} c_j\dot{\mathbf{x}}_j$
    $\mathbf{u} = \mathbf{u}_0 + h\sum_{j=1}^{6} c_j\dot{\mathbf{u}}_j$

    //estimate the local truncation error
    $err = estimate\_LTE(\mathbf{x}^*, \mathbf{u}^*, \mathbf{x}, \mathbf{u})$;

    //adjust time step
    if( $err > tol$ )
    {
        $h = adjust\_time\_step(err, tol, h)$;
    }
```

```
        }

        //project the solution onto
        //the constraint manifold
        $[\mathbf{x},\mathbf{u}] = project\_solution(\mathbf{x},\mathbf{u})$;

        //increment time
        $t = t + h$;
}
```

The calculations done within the function *RK_stage*() look like:

```
$[\dot{\mathbf{x}},\dot{\mathbf{u}}]$ = RK_stage( $\mathbf{x}_0,\mathbf{u}_0$ )
{
        //project initial values onto
        //the constraint manifold
        $[\mathbf{x}_0,\mathbf{u}_0] = project\_solution(\mathbf{x}_0,\mathbf{u}_0)$;

        //optimize to find $\tilde{\mathbf{N}}$
        $\tilde{\mathbf{N}} = compute\_N(\mathbf{x}_0,\mathbf{u}_0)$;

        //compute derivatives
        $\dot{\mathbf{x}} = \mathbf{H}(\mathbf{x}_0)\mathbf{u}_0$;
        $\dot{\mathbf{u}} = \mathbf{M}^{-1}\left(\mathbf{F}_{nc} - \tilde{\mathbf{W}}^T\tilde{\mathbf{N}}\right)$;
}
```

The 6-stage, 5th order Runge-Kutta method of Cash and Karp [16, 60]) was used to implement a time-stepping scheme for 2-dimensional simulations within Matlab.

This method actually contains embedded formulas of orders 1 through 5; however, our implementation uses only the 4th and 5th-order. To keep the local truncation error below user-specified tolerances, we utilized the time-step adjustment formula described in [35]. Matlab's Optimization Toolbox was used to solve the quadratic program at each stage, with a regularization term added to the objective function in order to approximate the minimum Euclidean norm solution.

Chapter 4 : Experimental Validation

In order to experimentally validate this approach, numerical predictions were compared with actual trajectories of the pendulum device illustrated in Figure 4.1. One of the primary advantages of modeling such a simple system is that the data measurements were easy to obtain, accurate, and consistent. In addition, analysis of this system has provided some surprising insights into the mechanics of an unlubricated full journal bearing, a component very common in rotating machinery.

4.1 Experimental Setup

The pendulum itself consists of a cylindrical pivot (the *journal*) made completely out of the test material, a steel rod threaded at both ends, and one or two washers affixed to the end of the rod. The upper end of the rod is fastened to the pivot, while one or two washers are affixed to the lower end. This modular setup enabled us to collect data for four different contact materials (brass, nylon, stainless steel, and Teflon®) in two different mass configurations.

Before assembling the device, both the journal and the bearings were cleaned with 100% ethyl alcohol and then dried with Kimwipes® to remove any surface oils or residues. The journal was then inserted into the aluminum bearings from the side, as shown in Figure 4.2. The clearance between the journal and the bearings was small enough (~1/1000th of the journal radius) such that the journal fit cleanly into the bearings with little to no ambient pressure on the journal.

Next, the washers were affixed to the lower end of the rod. The rod's threaded upper portion was then wrapped with Teflon™ tape and securely screwed halfway into the journal. The Teflon® tape ensured that the rod did not wiggle out of the journal as oscillations progressed. A digital caliper was used to ensure that the top of the rod aligned with the axis of rotation of the journal. Finally, the axle of the optical encoder was inserted into a small hole in one end of the pivot and secured with a small jackscrew. After the pendulum was assembled, the supporting base plate was leveled by adjusting three different jackscrews that extended through the base plate onto the surface below. These jackscrews were adjusted until the bubbles of two perpendicular liquid levels were centered.

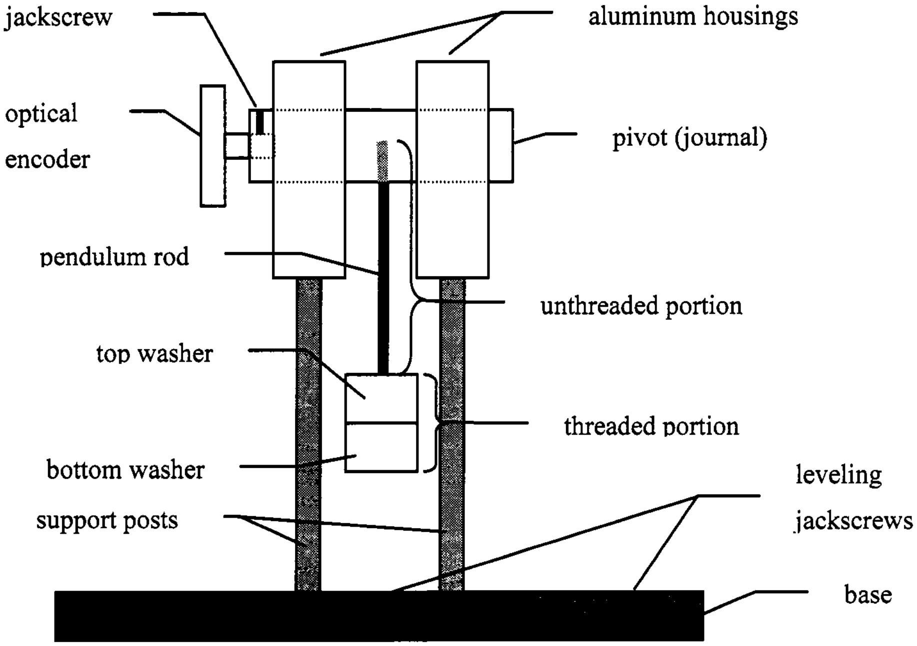

Figure 4.1. A side view of the experimental pendulum device.

To ensure that the pendulum was released from a very nearly horizontal position (to within ~1 degree), a metal strip was fastened horizontally across one side of the support plates. First, the pendulum was raised until the rod comes into contact with the metal strip. A small level was then placed flat on the pendulum rod while the jackscrew directly underneath the pendulum was adjusted until the bubble was centered. This gave us confidence that the pendulum was being released from the horizontal to within about 0.5 degrees. Angular data was collected until the pendulum came to rest.

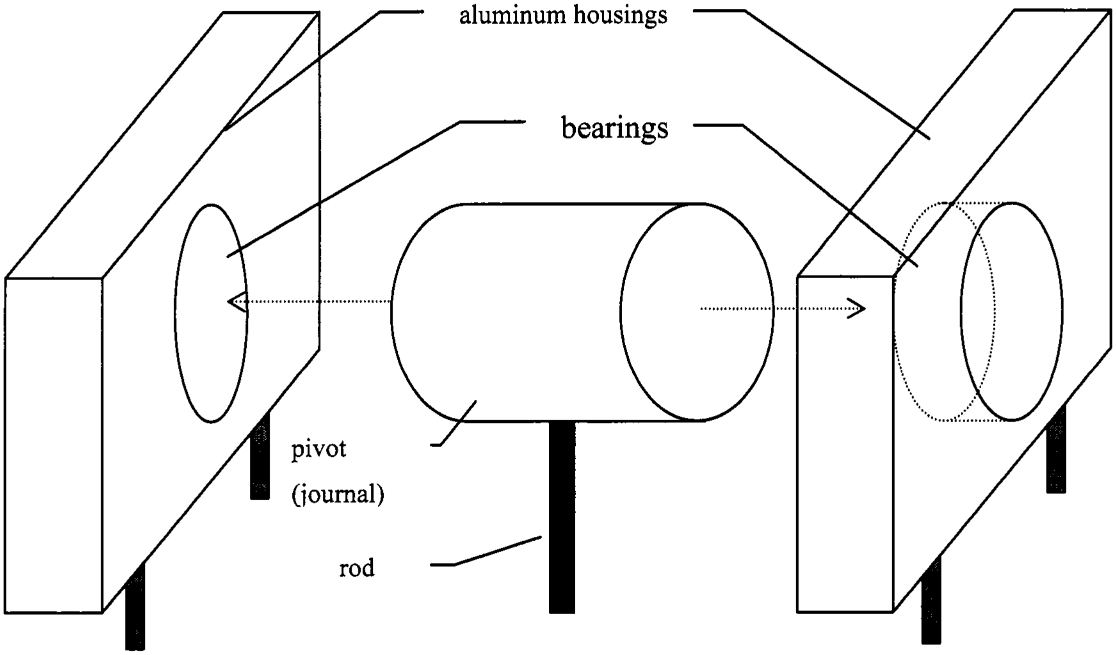

Figure 4.2. Orthographic view of the journal and aluminum bearings.

The pendulum angle θ at each instant was computed from the output of an optical encoder which recorded angle transitions of $\Delta\theta = 0.36$ degrees (~0.0063 radians) as transitions between two voltage channels. Since the initial angle θ_0 was known to be very close to zero, the values for $\theta(t)$ could be calculated by accumulating $\Delta\theta$ as channel transitions were detected. Since detecting each transition was critical, voltages from each channel were monitored at 10 KHz, high enough to catch transitions at peak rotational velocity. Reversals in the angular velocity were detected by monitoring reversals of the relative order of channel transitions.

The optical encoder was chosen due to its simplicity and the fact that the angle transitions were constant for the full range of motion. In addition, the device has very little resistance to motion, which left the inertia of the pendulum unaffected.

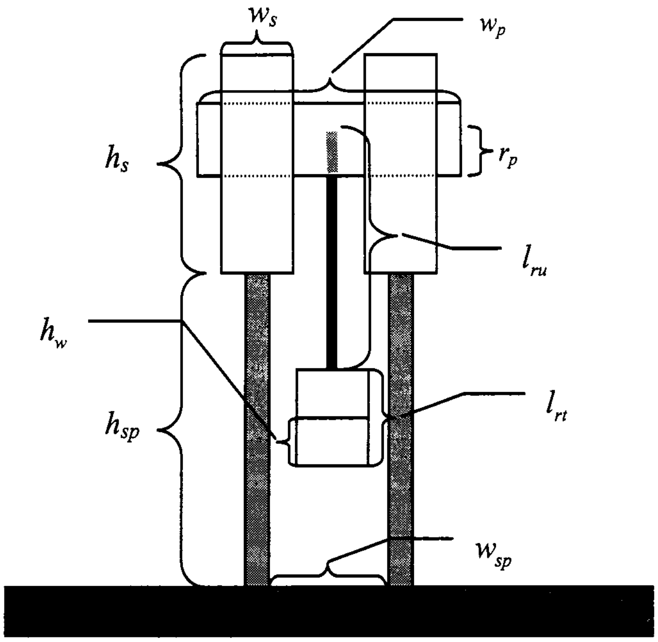

Figure 4.3. Spatial dimensions of the pendulum components.

The spatial dimensions of the pendulum components are shown in Figure 4.3 above, and actual values with uncertainties are given in Table 4.1, below. Length measurements were taken with a digital caliper and with a regular metric ruler. Masses were measured using a digital scale. Uncertainties in the measurements were estimated by taking multiple readings and using the median value.

Table 4.1. Physical dimensions of the pendulum device.

mass of pivot, brass	$m_{p,br}$	196 g
mass of pivot, nylon	$m_{p,ny}$	27.5 g
mass of pivot, stainless steel	$m_{p,ss}$	188 g
mass of pivot, Teflon®	$m_{p,te}$	52 g
radius of pivot	r_p	1.0 ± 0.01 cm
width of pivot	w_p	8.26 ± 0.01 cm
mass of rod	m_r	74 g
Length of unthreaded portion of rod	l_{ru}	9.8 ± 0.01 cm
Length of threaded portion of rod	l_{rt}	4.2 ± 0.01 cm
radius of rod	r_r	0.48 ± 0.01 cm
mass of a single washer	m_w	90.5 g
inner radius of a washer	r_{wi}	0.48 ± 0.01 cm*
outer radius of a washer	r_{wo}	1.5 ± 0.01 cm
height of a washer	h_w	1.87 ± 0.05 cm
width of support plate	w_s	2.0 ± 0.02 cm
height of support plate	h_s	6.0 ± 0.05 cm
width of support post	w_{sp}	1.13 ± 0.05 cm
height of support post	h_{sp}	12.0 ± 0.5 cm
distance between support posts	w_{sp}	4.7 ± 0.05 cm

4.2 Mechanical Analysis of the Pendulum

In order to apply the time-stepping scheme developed in Chapter 3, the experimental setup must be analyzed within the theoretical framework developed within Chapter 2. To do this, we choose a fixed global coordinate frame centered at the axis of rotation with its x-axis pointing rightward and its y-axis pointing upward, as illustrated below in Figure 4.4. Since the dynamics are effectively described in 2-dimensions, we ignore the z-coordinate. Let (r_x, r_y) denote the x and y-coordinates of the pendulum's center of mass with respect to the global frame. The *local* coordinate frame of the pendulum is centered at (r_x, r_y) with its x-axis pointing along the rod and its y-axis at 90 degrees from the x-axis in the counter-clockwise direction. The orientation θ of the pendulum is the counter-clockwise angle from the global x-axis to the pendulum's local x-axis. The distance from (r_x, r_y) to the center of the pivot is denoted by l, and the radius of the pivot is denoted by r_p.

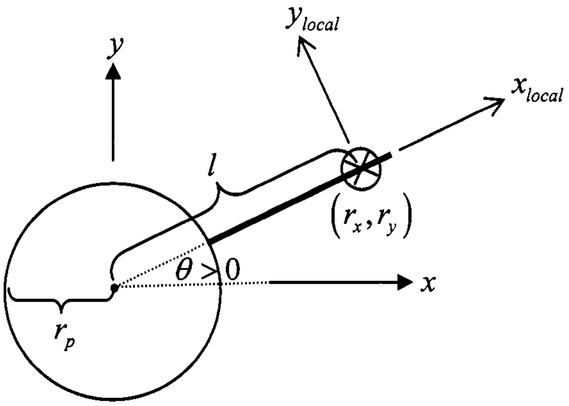

Figure 4.4. The global and local body coordinate frames of the pendulum.

Let m denote the total mass of the pendulum. The only non-constraint force on the pendulum is gravity. Since gravity exerts no torque about the center of mass, the generalized non-constraint force for this system is

$$\mathbf{F}_{nc} = [0, -mg, 0]^T. \tag{4.1}$$

As gravity pulls down on the center of mass, the support plates exert normal forces on the pivot to prevent interpenetration, and frictional forces oppose the relative sliding motion. Both of these forces are continuously distributed about the contact area between the pivot and the support plates. This continuous contact area is approximated by m equally-spaced contact points, with the first point at the base of the rod with respect to the local frame, as shown in Figure 4.5. Since the pivot is designed to fit into the support plates without any gap, we assume that none of these contacts can vanish, and thus $s = 0$.

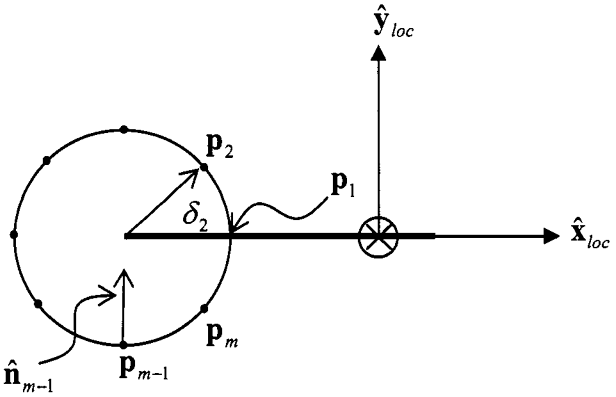

Figure 4.5. Continuous contact approximated by *m* equally-spaced points.

It follows from equation (2.9) that the equations of motion for this system are:

$$m\ddot{r}_x = \sum_{i=1}^{m} \left(\mathbf{N}_{i,x} + \mathbf{f}_{i,x} \right) = \sum_{i=1}^{m} \left(\hat{\mathbf{n}}_{i,x} - f\left(\|\mathbf{v}_i\| \right) \hat{\mathbf{v}}_{i,x} \right) N_i, \tag{4.2}$$

$$m\ddot{r}_y = -mg + \sum_{i=1}^{m} \left(\mathbf{N}_{i,y} + \mathbf{f}_{i,y} \right) = -mg + \sum_{i=1}^{m} \left(\hat{\mathbf{n}}_{i,y} - f\left(\|\mathbf{v}_i\| \right) \hat{\mathbf{v}}_{i,y} \right) N_i, \tag{4.3}$$

$$J\ddot{\theta} = \sum_{i=1}^{m} \mathbf{r}_i \times (\mathbf{N}_i + \mathbf{f}_i) = \sum_{i=1}^{m} \mathbf{r}_i \times \left(\hat{\mathbf{n}}_i - f\left(\|\mathbf{v}_i\|\right)\hat{\mathbf{v}}_i\right) N_i . \tag{4.4}$$

Figure 4.6 shows a force diagram for four contact points when θ is increasing.

In order to simplify these equations, we can use the dimensionless coordinates

$$\bar{r}_x \equiv \frac{r_x}{l}, \tag{4.5}$$

$$\bar{r}_y \equiv \frac{r_y}{l}, \tag{4.6}$$

$$\bar{\theta} \equiv \theta . \tag{4.7}$$

We also define the dimensionless parameter

$$\alpha \equiv \frac{J}{ml^2} . \tag{4.8}$$

The quantity ml^2 is the rotational inertia the pendulum would have if all its mass could be concentrated equally at both ends. Since it is physically impossible to achieve this and still maintain the rigidity of the pendulum, this is a strict upper bound on J. At the other extreme, it is also impossible to concentrate all mass into a single point, and so zero is a strict lower bound on J. Therefore, we must have $0 < \alpha < 1$.

For small oscillations in the absence of friction, the angular of the simple harmonic oscillator (SHO) solution is

$$K \equiv \sqrt{\frac{g}{l(1+\alpha)}} . \tag{4.9}$$

With this parameter we can define the dimensionless time as

$$\bar{t} \equiv Kt . \tag{4.10}$$

It can easily be shown that (4.10) measures time in multiples of the angular period of the frictionless simple harmonic oscillator (SHO). In other words, for very small oscillations a frictionless pendulum would complete one full swing in 2π dimensionless time units.

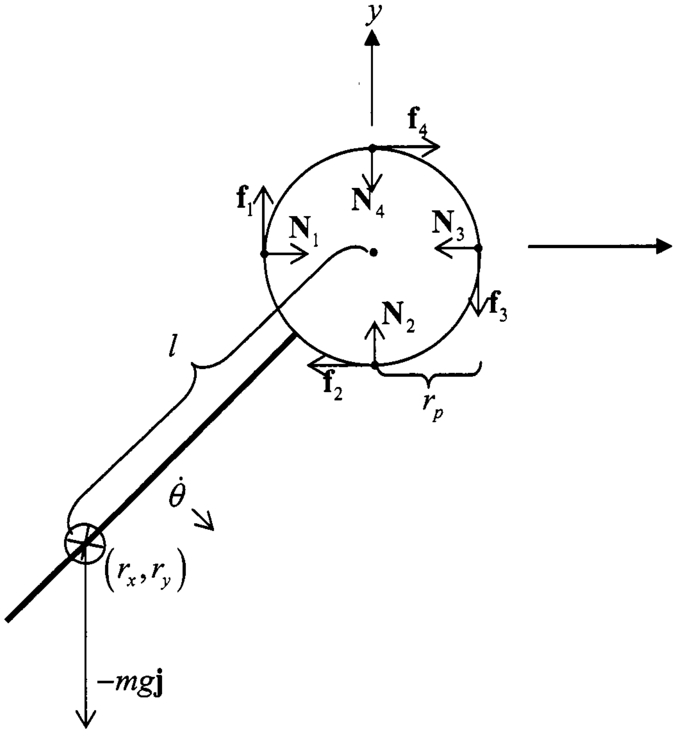

Figure 4.6. Normal and frictional forces for a four-point discretization.

From (4.10) we can compute the rate of change of dimensionless time with respect to the actual (dimensional) time:

(4.11)

$$\frac{d\bar{t}}{dt} = K .$$

It follows that

$$\dot{r}_x \equiv \frac{d}{dt}\left(r_x\right) = \frac{d\bar{t}}{dt}\frac{d}{d\bar{t}}\left(l\bar{r}_x\right) = K\frac{d}{d\bar{t}}\left(l\bar{r}_x\right) = Kl\dot{\bar{r}}_x, \tag{4.12}$$

where $\dot{\bar{r}}_x$ is taken to mean the derivative of the dimensionless variable $\bar{r}_x$ with respect to the dimensionless time $\bar{t}$. Similarly,

$$\dot{r}_y = Kl\dot{\bar{r}}_y, \tag{4.13}$$

$$\dot{\theta} = K\dot{\bar{\theta}}. \tag{4.14}$$

From these we define the dimensionless generalized position as

$$\bar{\mathbf{x}} \equiv \left[\bar{r}_x, \bar{r}_y, \bar{\theta}\right]^T. \tag{4.15}$$

Since this system is only 2-dimensional, the dimensionless generalized velocity is

$$\bar{\mathbf{u}} = \dot{\bar{\mathbf{x}}} = \left[\dot{\bar{r}}_x, \dot{\bar{r}}_y, \dot{\bar{\theta}}\right]^T. \tag{4.16}$$

Differentiating (4.12) – (4.14) once more with respect to t gives

$$\ddot{r}_x = \tfrac{g}{1+\alpha}\ddot{\bar{r}}_x, \tag{4.17}$$

$$\ddot{r}_y = \tfrac{g}{1+\alpha}\ddot{\bar{r}}_y, \tag{4.18}$$

$$\ddot{\theta} = \tfrac{g}{l(1+\alpha)}\ddot{\bar{\theta}}. \tag{4.19}$$

Substituting these back into (4.2) – (4.4), we get

$$\left(\tfrac{mg}{1+\alpha}\right)\ddot{\bar{r}}_x = \sum_{i=1}^{m}\left(\hat{\mathbf{n}}_{i,x} - f\left(\left\|\mathbf{v}_i\right\|\right)\hat{\mathbf{v}}_{i,x}\right)N_i, \tag{4.20}$$

$$\left(\tfrac{mg}{1+\alpha}\right)\ddot{\bar{r}}_y = -mg + \sum_{i=1}^{m}\left(\hat{\mathbf{n}}_{i,y} - f\left(\left\|\mathbf{v}_i\right\|\right)\hat{\mathbf{v}}_{i,y}\right)N_i, \tag{4.21}$$

$$\tfrac{g}{l(1+\alpha)}J\ddot{\bar{\theta}} = \sum_{i=1}^{m}\mathbf{r}_i \times \left(\hat{\mathbf{n}}_i - f\left(\left\|\mathbf{v}_i\right\|\right)\hat{\mathbf{v}}_i\right)N_i. \tag{4.22}$$

Multiplying (4.20) – (4.21) by $1/_{mg}$ and multiplying (4.22) by $1/_{mgl}$ gives us

$$\left(\tfrac{1}{1+\alpha}\right)\ddot{\bar{r}}_x = \sum_{i=1}^{m}\left(\hat{\mathbf{n}}_{i,x} - f\left(\|\mathbf{v}_i\|\right)\hat{\mathbf{v}}_{i,x}\right)\bar{N}_i , \tag{4.23}$$

$$\left(\tfrac{1}{1+\alpha}\right)\ddot{\bar{r}}_y = -1 + \sum_{i=1}^{m}\left(\hat{\mathbf{n}}_{i,y} - f\left(\|\mathbf{v}_i\|\right)\hat{\mathbf{v}}_{i,y}\right)\bar{N}_i , \tag{4.24}$$

$$\left(\tfrac{\alpha}{1+\alpha}\right)\ddot{\theta} = \sum_{i=1}^{m}\bar{\mathbf{r}}_i \times \left(\hat{\mathbf{n}}_i - f\left(\|\mathbf{v}_i\|\right)\hat{\mathbf{v}}_i\right)\bar{N}_i . \tag{4.25}$$

where

$$\bar{N}_i = \tfrac{1}{mg}N_i . \tag{4.26}$$

It follows that the dimensionless mass matrix for our system is

$$\mathbf{M} = \frac{1}{1+\alpha}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \alpha \end{bmatrix}, \tag{4.27}$$

and the dimensionless generalized nonconstraint force is

$$\bar{\mathbf{F}}_{nc} = \tfrac{1}{mg}\mathbf{F}_{nc} = [0,-1,0]^T . \tag{4.28}$$

In order to compute $\tilde{\mathbf{W}}$ for this contact configuration, we define the dimensionless parameter

$$\beta = \frac{r_p}{l} . \tag{4.29}$$

By design, the radius of the journal satisfies $0 < r_p < l$, and therefore $0 < \beta < 1$. As illustrated in Figure 4.5 above, the regular angle from the pendulum's local x-axis to its i^{th} contact point has been defined to be

(4.30) $$\delta_i = (i-1)\frac{2\pi}{m},\ i = 1,\dots,m.$$

Therefore, the contact normals in the local frame of the pendulum are

(4.31) $$\hat{\mathbf{n}}_{i,loc} = [-\cos\delta_i, -\sin\delta_i].$$

Multiplying these vectors by the orientation matrix

(4.32) $$\mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

transforms them into the global coordinate frame; thus, the contact normal in global coordinates is given by

(4.33) $$\hat{\mathbf{n}}_i = [-\cos\theta\cos\delta_i + \sin\theta\sin\delta_i, -\sin\theta\cos\delta_i - \cos\theta\sin\delta_i]^T.$$

By a similar transformation, the contact points with respect to the global frame are

(4.34) $$\begin{aligned}\bar{\mathbf{p}}_i = \Big[&\bar{r}_x + (\beta\cos\delta_i - 1)\cos\theta - \beta\sin\delta_i\sin\theta, \\ &\bar{r}_y + (\beta\cos\delta_i - 1)\sin\theta - \beta\sin\delta_i\cos\theta\Big]^T,\end{aligned}$$

and the vectors from the center of mass to the contact points are

(4.35) $$\bar{\mathbf{r}}_i = \Big[(\beta\cos\delta_i - 1)\cos\theta - \beta\sin\delta_i\sin\theta, (\beta\cos\delta_i - 1)\sin\theta - \beta\sin\delta_i\cos\theta\Big]^T.$$

The direction of the sliding velocity at $\mathbf{p}_i$ within the local frame of the pendulum is

(4.36) $$\hat{\mathbf{v}}_{i,loc} = \operatorname{sgn}\dot{\theta}[-\sin\delta_i, \cos\delta_i]^T.$$

Transforming this vector into the global frame gives

(4.37) $$\hat{\mathbf{v}}_i = \operatorname{sgn}\dot{\theta}[-\sin\theta\cos\delta_i - \cos\theta\sin\delta_i, \cos\theta\cos\delta_i - \sin\theta\sin\delta_i]^T.$$

The dimensionless torque on the pendulum at the i^{th} contact is

$$\bar{\boldsymbol{\tau}}_i = \bar{\mathbf{r}}_i \times \left(\hat{\mathbf{n}}_i - f\left(\|\mathbf{v}_i\|\right)\hat{\mathbf{v}}_i\right)\bar{N}_i = \left[\sin\delta_i - f\left(\|\mathbf{v}_i\|\right)\operatorname{sgn}\dot{\theta}\left(\beta - \cos\delta_i\right)\right]\bar{N}_i . \tag{4.38}$$

The dimensionless generalized contact force at each point $\bar{\mathbf{p}}_i$ is therefore

$$\begin{aligned} \bar{\mathbf{F}}_{c,i} = \big[& -\cos\theta\cos\delta_i + \sin\theta\sin\delta_i + f\operatorname{sgn}\dot{\theta}\left(\sin\theta\cos\delta_i + \cos\theta\sin\delta_i\right), \\ & -\sin\theta\cos\delta_i - \cos\theta\sin\delta_i - f\operatorname{sgn}\dot{\theta}\left(\cos\theta\cos\delta_i - \sin\theta\sin\delta_i\right), \\ & \sin\delta_i - f\operatorname{sgn}\dot{\theta}\left(\beta - \cos\delta_i\right)\big]^T \bar{N}_i \end{aligned} \tag{4.39}$$

It follows from equation (2.13) that

$$\begin{aligned} \mathbf{w}_i = \big[& \cos\theta\cos\delta_i - \sin\theta\sin\delta_i - f\operatorname{sgn}\dot{\theta}\left(\sin\theta\cos\delta_i + \cos\theta\sin\delta_i\right), \\ & \sin\theta\cos\delta_i + \cos\theta\sin\delta_i + f\operatorname{sgn}\dot{\theta}\left(\cos\theta\cos\delta_i - \sin\theta\sin\delta_i\right), \\ & -\sin\delta_i + f\operatorname{sgn}\dot{\theta}\left(\beta - \cos\delta_i\right)\big]^T . \end{aligned} \tag{4.40}$$

In order to accurately match experimental trajectories, we used the following friction model:

$$f_i\left(\|\mathbf{v}_i\|\right) = \mu + b\|\mathbf{v}_i\|, \tag{4.41}$$

which includes both a Coulomb (constant) term and a linear viscosity term. For any particular mass configuration, this friction model provides a 2-parameter family of curves with which experimental trajectories can be matched. We will see later in Section 4.3 that both coefficients are necessary to properly fit the data. Since the physically correct solution will satisfy all the constraints identically, the magnitude of the sliding velocity at each contact point is

$$\|\mathbf{v}_i\| = r_p\left|\dot{\theta}\right| = r_p K\left|\dot{\bar{\theta}}\right| . \tag{4.42}$$

This form also allows us to calibrate a dimensional value for b.

The experimental setup requires that each contact point $\mathbf{p}_i$ remain within a disk of radius β centered at the global origin, and therefore the system is subject to the inequality constraints

$$g_i\left(\bar{\mathbf{x}}\right) = \bar{p}_{i,x}^2 + \bar{p}_{i,y}^2 - \beta^2 \le 0\,, \;\; i = 1,\ldots,m+s\,. \tag{4.43}$$

Substituting (4.34) into (4.43) simplifies to

$$\begin{gathered} g_i\left(\bar{\mathbf{x}}\right) = 2\Big[\beta\left(\bar{r}_x\cos\theta + \bar{r}_y\sin\theta - 1\right)\cos\delta_i + \\ \beta\left(\bar{r}_y\cos\theta - \bar{r}_x\sin\theta\right)\sin\delta_i - \bar{r}_x\cos\theta - \bar{r}_y\sin\theta\Big] + \bar{r}_x^2 + \bar{r}_y^2 + 1\,, \end{gathered} \tag{4.44}$$

and differentiating this expression with respect to $\bar{\mathbf{x}}$ gives

$$\begin{gathered} \mathbf{g}_{i,\bar{\mathbf{x}}}\left(\bar{\mathbf{x}}\right) = 2\Big[\bar{r}_x - \cos\theta + \beta\left(\cos\theta\cos\delta_i - \sin\theta\sin\delta_i\right), \\ \bar{r}_y - \sin\theta + \beta\left(\sin\theta\cos\delta_i + \cos\theta\sin\delta_i\right), \\ \bar{r}_x\sin\theta - \bar{r}_y\cos\theta + \beta\left(\bar{r}_y\cos\theta - \bar{r}_x\sin\theta\right)\cos\delta_i - \beta\left(\bar{r}_x\cos\theta + \bar{r}_y\sin\theta\right)\sin\delta_i\Big] \end{gathered} \tag{4.45}$$

Finally, multiplying (4.45) by $\mathbf{H}\left(\bar{\mathbf{x}}\right)$ and then differentiating with respect to $\bar{t}$ gives us

$$\begin{gathered} \frac{d}{d\bar{t}}\left(\mathbf{g}_{i,\bar{\mathbf{x}}}\mathbf{H}\right) = 2\Big[\dot{\bar{r}}_x + \sin\theta\dot{\bar{\theta}}\left(1 - \beta\cos\delta_i\right) - \beta\cos\theta\dot{\bar{\theta}}\sin\delta_i, \\ \dot{r}_y - \cos\theta\dot{\bar{\theta}}\left(1 - \beta\cos\delta_i\right) - \beta\sin\theta\dot{\bar{\theta}}\sin\delta_i, \\ \dot{\bar{r}}_x\sin\theta + \bar{r}_x\cos\theta\dot{\bar{\theta}} - \dot{\bar{r}}_y\cos\theta + \bar{r}_y\sin\theta\dot{\bar{\theta}} \\ -\beta\left(\dot{\bar{r}}_x\cos\theta - \bar{r}_x\sin\theta\dot{\bar{\theta}} + \dot{\bar{r}}_y\sin\theta + \bar{r}_y\cos\theta\dot{\bar{\theta}}\right)\sin\delta_i \\ -\beta\left(\dot{\bar{r}}_x\sin\theta + \bar{r}_x\cos\theta\dot{\bar{\theta}} - \dot{\bar{r}}_y\cos\theta + \bar{r}_y\sin\theta\dot{\bar{\theta}}\right)\cos\delta_i\Big]. \end{gathered} \tag{4.46}$$

Equations (4.45) and (4.46) are used to compute $\tilde{\mathbf{N}}$ through optimization and to project **x** and **u** onto the constraint manifold at each stage.

In order to derive an approximate ordinary differential equation, we can find which constraints are active over a particular time interval and then solve for $\bar{\mathbf{N}}$. The algebra for anything more than 4 contact points is nontrivial; however, the ODE derived from a 4-point approximation can still provide some insight into the dynamics of this system. Let us assume that the pendulum is released from an angle $-\frac{\pi}{2} < \theta_0 < 0$, and thus only $\ddot{\bar{g}}_1$ and $\ddot{\bar{g}}_4$ affect the solution for $\bar{\mathbf{N}}$. From these assumptions, we compute

$$\textbf{(4.47)} \qquad \bar{N}_1 = \frac{\alpha(1+\alpha)f\cos\theta - (1+\alpha)[1+\alpha-\beta f]\sin\theta + (1+\alpha-\beta f)\dot{\bar{\theta}}^2}{(1+\alpha)\left[1+\alpha+f^2(1+\alpha)-\beta f(1+f)\right]},$$

$$\textbf{(4.48)} \qquad \bar{N}_4 = \frac{\alpha(1+\alpha)\cos\theta + (1+\alpha)\left[(1+\alpha)-\beta\right]f\sin\theta - (1+\alpha-\beta)f\dot{\bar{\theta}}^2}{(1+\alpha)\left[1+\alpha+f^2(1+\alpha)-\beta f(1+f)\right]}.$$

It is easy to verify that both $\bar{N}_1 > 0$ and $\bar{N}_4 > 0$ for $0 < \alpha < 1$, $0 < \beta < 1$, and $f \ll 1$. Since both $\bar{N}_1$ and $\bar{N}_4$ are continuous functions of $\theta(t)$ and $\dot{\bar{\theta}}(t)$, this approximation is valid for at least a short period of time.

At first this solution may seem physically unreasonable: how could it be that $\bar{N}_1 > 0$ at the moment of release? The answer to this question lies in an intuitive evaluation of Figure 4.6. An instant after the pendulum is released there is a *rightward* frictional force at contact 4 which must be overcompensated for by the normal force at contact 1 (after all, the CM is accelerating *leftward*). This simultaneously produces an *upward* frictional force at contact 1 which helps reduce the normal force at contact 4 required to prevent penetration. Fortunately, we see that $\bar{N}_1 \to 0$ as $f \to 0$, as we would expect.

It follows that

(4.49)
$$\ddot{\bar{\theta}} = \frac{-(1+\alpha)\left[1+(1-\beta)f^2-\beta f\right]\cos\theta-(1+\alpha)\beta\left(1-f^2\right)\sin\theta+\beta f(1-f)\dot{\bar{\theta}}^2}{\left[(1+\alpha)\left(1+f^2\right)-\beta f(1+f)\right]}$$

over the time interval during which both $\bar{N}_1 > 0$ and $\bar{N}_4 > 0$. We can see that as $f \to 0$ or $\beta \to 0$, equation (4.49) approaches

(4.50)
$$\ddot{\bar{\theta}} = -\cos\bar{\theta},$$

which can also be derived directly by applying the parallel axis theorem to the frictionless system. The analysis above confirms the intuition that friction loses its "grip" on the pendulum as the ratio between r_p and l approaches zero.

4.3 Experimental Results

Before calibrating friction coefficients from the experimental data, it was important to verify that the discretization in time and space was sufficient to produce accurate simulations. Figure 4.7 below compares simulations at three different error tolerances for time stepping. Even though tightening the error tolerances beyond 10^{-3} provided no significant improvement, an absolute and relative error tolerance of 10^{-6} was used for all simulations.

Due to the curvature of the contact area, it was necessary to discretize the contact area into a sufficient number of points. While it is difficult to prove that this refinement resulted in convergence to the solution of a continuous problem, numerical experiments confirmed convergence to a stationary point. Figure 4.8 below illustrates this convergence for stainless steel using 4, 8, 16, and 32 isolated contact points. Based upon these numerical experiments, it was decided that using

16 evenly-spaced discrete contact points produced sufficiently accurate simulations.

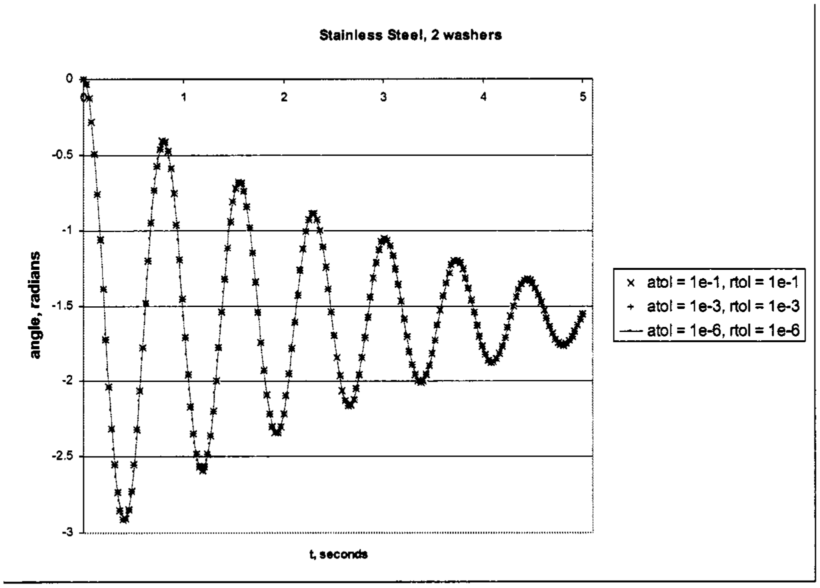

Figure 4.7. Simulated trajectories for absolute and relative error tolerances of 10^{-1}, 10^{-3}, and 10^{-6} in the time domain (16 contact points).

Since the mass and length parameters of each trajectory are known, the friction coefficients μ and b are the only free parameters in the system. These coefficients are adjusted until the simulated trajectory matches the actual data. Numerical experiments indicate that changes in μ and b have independent effects: when $\mu > 0$ and $b = 0$, the amplitude decays almost linearly, and oscillations stop in finite time. Decreasing μ and increasing b makes the amplitude decay more and more exponentially. At the other extreme, with $\mu = 0$ and $b > 0$, oscillations decay but never cease. For the same rate of attenuation, increasing b tends to shorten the period more than increasing μ does. All of these behaviors are illustrated in

Figure 4.9. Due to these independent effects, the (μ, b) pair required to match a particular trajectory was unique to within at about 10% uncertainty.

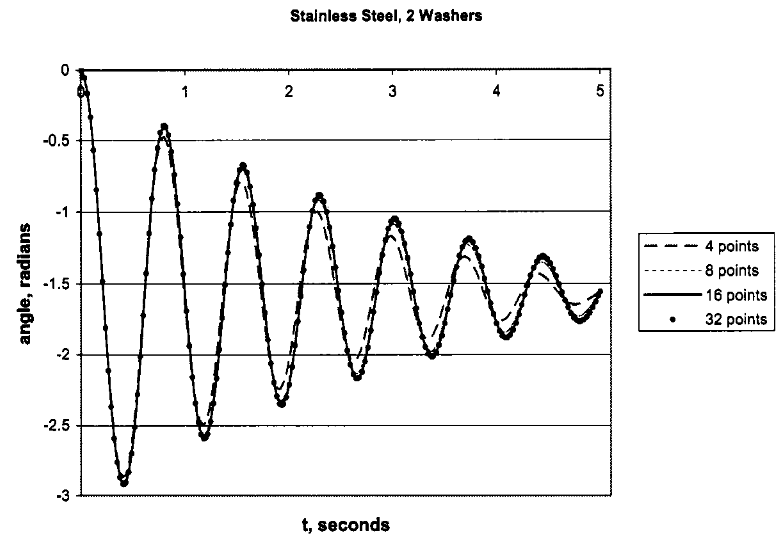

Figure 4.8. Simulated trajectories with 4, 8, 16, and 32 contact points (atol = rtol = 10^{-6}).

As an added check, our numerical solution was compared with the following empirical model:

$$\ddot{\bar{\theta}} = -\cos\bar{\theta} - \mu\,\mathrm{sgn}\,\dot{\bar{\theta}} - b r_p K \left|\dot{\bar{\theta}}\right| . \tag{4.51}$$

Strictly speaking, this should be written as an inclusion to take into account the set-valuedness of the right-hand side when $\dot{\bar{\theta}} = 0$ [39], but for the materials we tested, slip-stick transitions were observed only when the pendulum was very nearly vertical, and so Matlab's `ode45()` function could accurately integrate through the discontinuities of equation (4.51) by simply adjusting the time step.

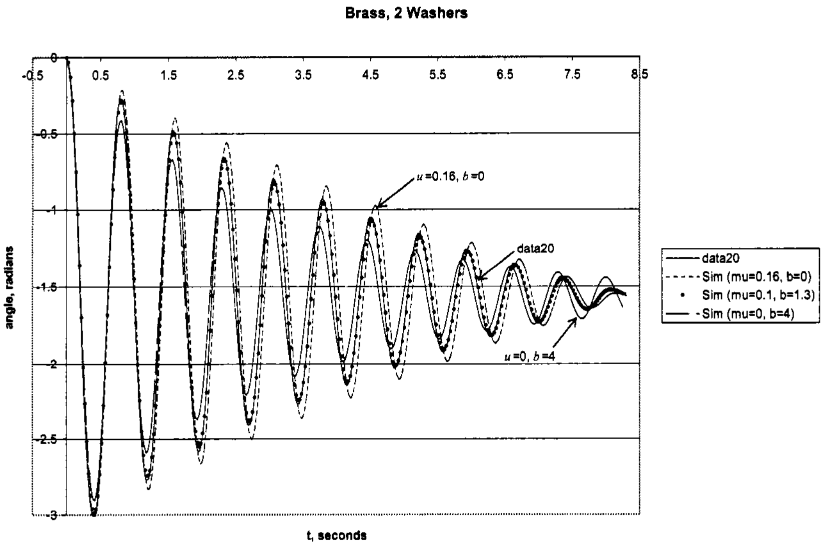

Figure 4.9. Distinct effects of μ and b on amplitude and frequency.

The first term on the right-hand side of (4.51) accounts for the gravitational force, the second term is an approximation of Coulomb friction, and the third term applies first-order velocity-dependent friction. All derivatives are taken with respect to dimensionless time (see equation (4.10)). Note that the first two terms do not take into account the pivot radius (the last term simply scales the angular velocity); therefore, this model cannot be used to calibrate physically meaningful friction coefficients. It also does not account for the variation in the normal load, which may explain its inability to match experimental trajectories as well as our load-dependent model. Nevertheless, numerical experiments confirm that both models exhibit similar behaviors when μ or b are adjusted. In the absence of friction, the two models are in perfect agreement.

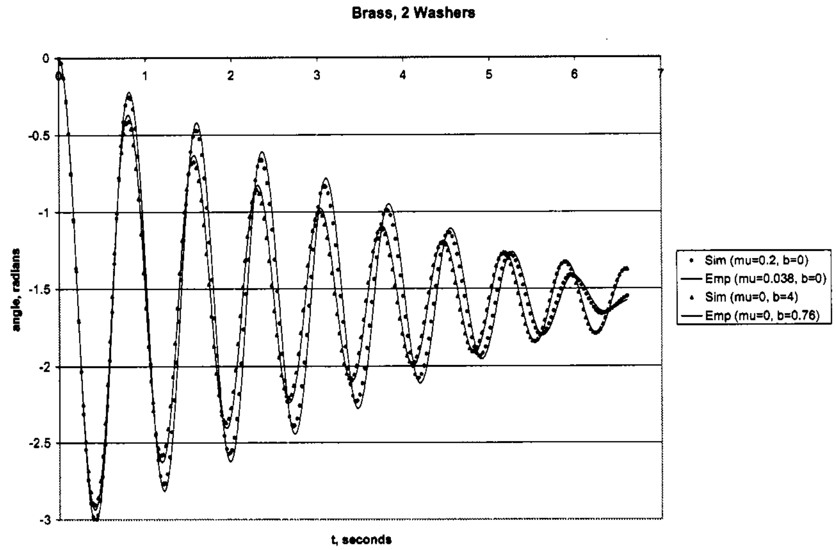

Figure 4.10. A comparison of the empirical model and the optimization-based contact model for different values of μ and b.

Note that the first two terms of the right-hand side of (4.51) do not take into account the pivot radius, and in the last term it simply scales the angular velocity. Therefore, this model cannot be used to calibrate system-independent friction coefficients. It also does not account for the variation in the normal load, which may explain the slight differences between the two models. Nevertheless, both models exhibit similar qualities when μ or b are adjusted, as shown in above in Figure 4.10.

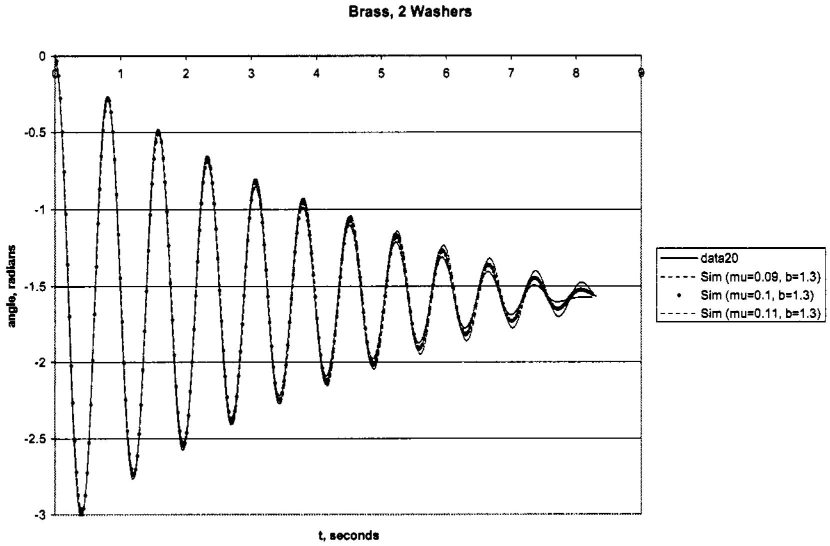

Figure 4.11. The effects of perturbations of μ and b on simulated pendulum trajectories.

In order to assess the data variability, a series of 20 consecutive trajectories was recorded for each washer configuration: a single-washer configuration (90.5 g) and a double-washer configuration (181 g). The trajectory data for the 2-washer configurations are shown below in Figure 4.12 – Figure 4.15, along with calibrated friction coefficient for the shortest and longest runs. Between each series, the pivot and support plates were cleaned with 100% ethyl alcohol and dried to remove any residue or debris. Within each series, however, the variability in the stopping time was considerable, with the stopping time tending to *increase* (i.e., friction *decreased*) with the number of oscillations. It is difficult to know for sure why this was the case, but it could be attributed to a combination of factors, such as heat buildup, abrasion, or oxidation.

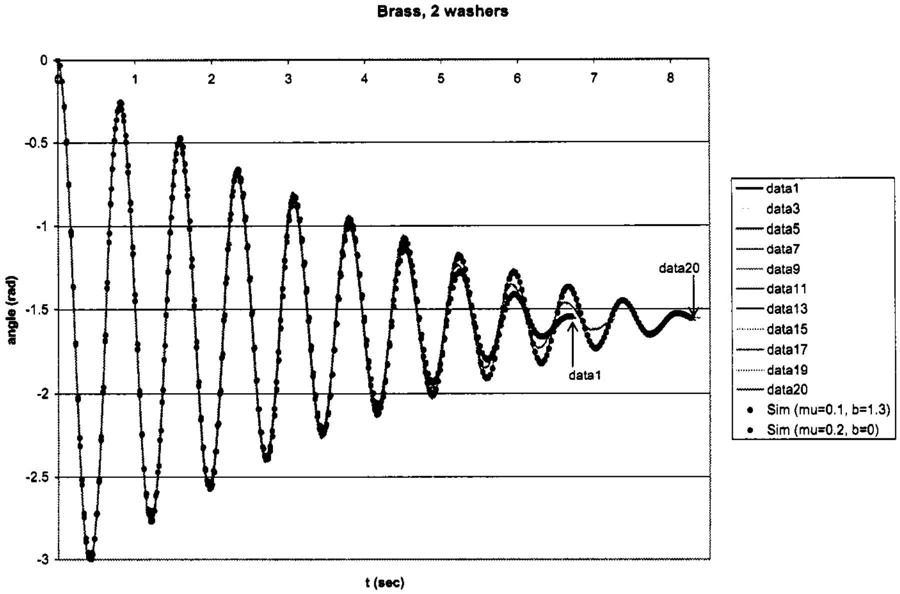

Figure 4.12. Trajectories and calibrated μ and b values for brass.

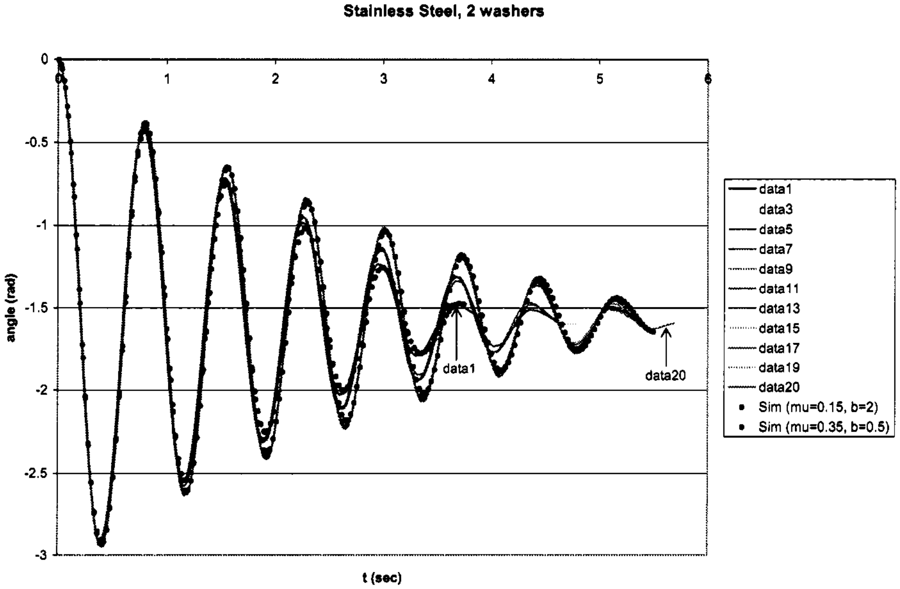

Figure 4.13. Trajectories and calibrated μ and b values for stainless steel.

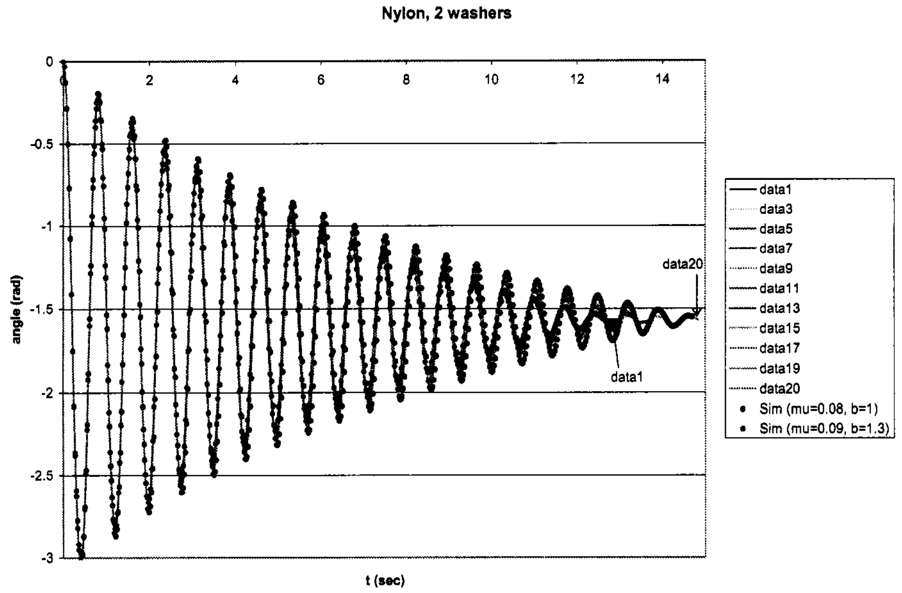

Figure 4.14. Trajectories and calibrated μ and b values for nylon.

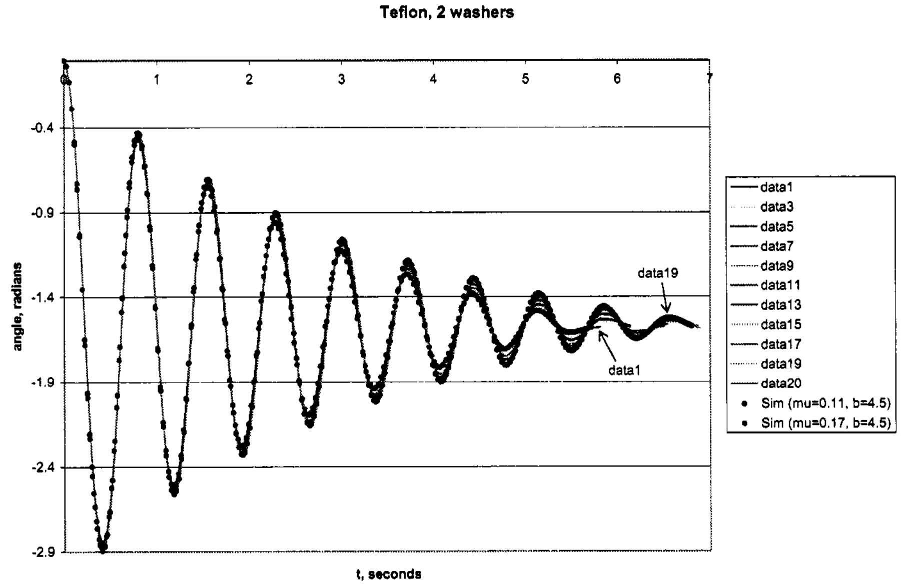

Figure 4.15. Trajectories and calibrated μ and b values for Teflon®.

We found that there was reasonable agreement between the friction coefficients calibrated in either washer configuration; however, coefficients calibrated from the single-washer configuration were generally *higher* than those calibrated from the double-washer (heavier) configuration. This could have been due to the heavier configuration generating more heat and creating more abrasion than the lighter single-washer system. The calibrated values are given below in Table 4.2. *Left* pairs correspond to the *longest* stopping time, while *right* pairs correspond to the *shortest* stopping time. Estimated uncertainty in μ is ± 0.005, while uncertainty in b is ± 0.1.

Table 4.2. Calibrated friction coefficients for 1 and 2-washer configurations.

Material	(μ, b), **1 washer**	(μ, b), **2 washers**
brass	(0.18, 0.4) – (0.29, 0.0)	(0.10, 1.3) – (0.20, 0.0)
steel	(0.22, 2.5) – (0.35, 1.5)	(0.15, 2.0) – (0.35, 0.5)
nylon	(0.13, 0.8) – (0.155,0.8)	(0.08, 1.0) – (0.09, 1.3)
Teflon®	(0.08, 5.0) – (0.18, 5.0)	(0.11, 4.5) – (0.17, 4.5)

When we fix $b = 0$ and use only μ to match the stopping time, the values we get are higher and somewhat more in line with commonly accepted values (see Table 4.4 below); however, we can see in Figure 4.9 that the data cannot be satisfactorily matched with $b = 0$. As before, *lower* values correspond to the *longest* stopping time, and *higher* values correspond to the *shortest* stopping time. Estimated uncertainty in μ is ± 0.005.

Table 4.3. Coulomb coefficients ($b = 0$) calibrated for 1- and 2-washer configurations.

Material	μ, 1 washer	μ, 2 washers
brass	0.20 – 0.29	0.16 – 0.20
steel	0.34 – 0.37	0.24 – 0.37
nylon	0.17 – 0.20	0.12 – 0.15
Teflon®	0.26 – 0.39	0.27 – 0.33

Commonly accepted Coulomb coefficients for some similar materials are given in the table below for comparison [11]:

Table 4.4. Coulomb coefficients for some material interfaces.

Material	μ
Aluminum on mild steel	0.48
Brass on mild steel	0.44
Brass on cast iron	0.30
Teflon on steel	0.04

It is also interesting to examine how this model predicts the distribution of the normal load as a function of the pendulum angle. Intuitively, we would expect the load of a stationary frictionless pendulum to be distributed symmetrically over the lower half of the pivot, while the upper half of the pivot experiences no normal load. Our contact simulation confirms this intuition, as shown in Figure 4.16 below (arrow lengths are in proportion to force magnitude, and no arrow indicates zero load).

Numerical simulations produce intuitive results in the frictionless dynamic case, as well (see Figure 4.17 below). When the pendulum is released from the horizontal, the normal load is distributed symmetrically about the lower half of the pivot. As the pendulum rotates downward, the load increases as it must counteract gravity and provide the centripetal force necessary to enforce nonpenetration. At the bottom of the trajectory, the normal load reaches its peak and becomes symmetrically distributed about the rod. As it swings back up, the normal load is the mirror image of what it was on the downswing. In the absence of friction there is no energy dissipation, so this behavior will oscillate back and forth without end.

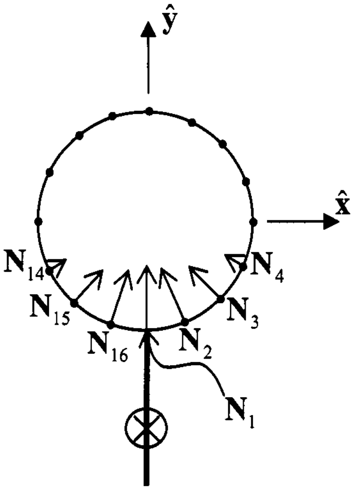

Figure 4.16. Predicted distribution of normal load for a stationary pendulum without friction (brass, 2 washers).

When friction is included in the model, however, our numerical simulations produce somewhat surprising results. At the instant the pendulum is released from the horizontal, the force distribution is nearly identical to the frictionless system. But as the motion continues, the normal load becomes concentrated over a smaller and smaller area. At the bottom of the trajectory, the normal load density (force per unit length) is roughly twice that of the frictionless system, and the center of the distribution lags behind the pendulum rod, as shown in Figure 4.18 below.

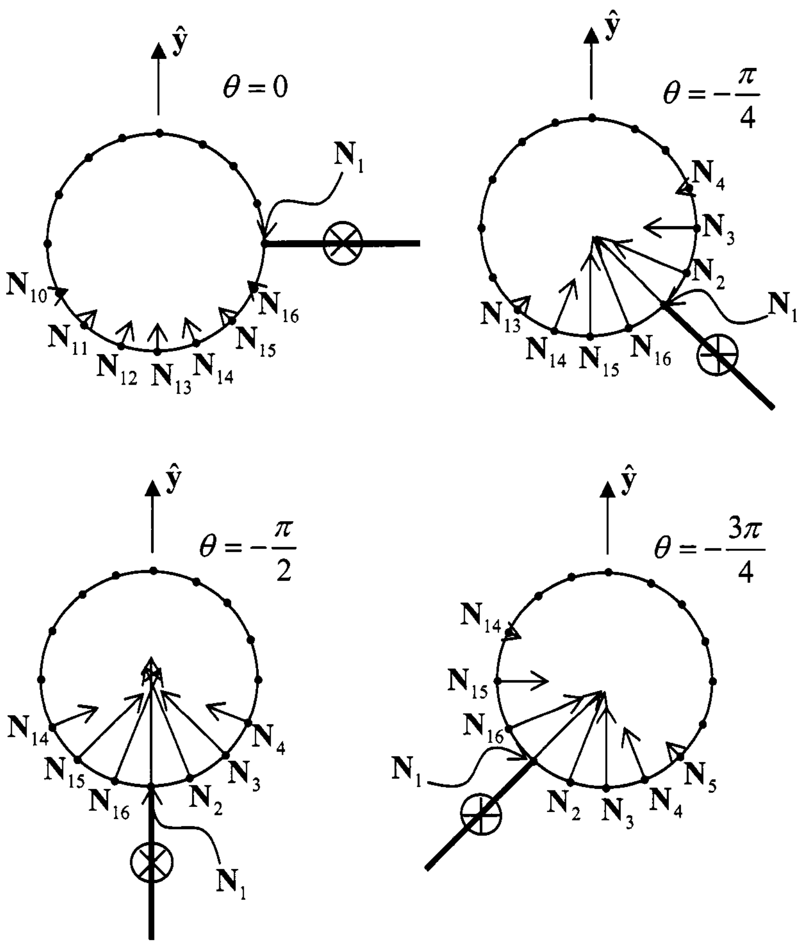

Figure 4.17. Predicted distribution of normal load for a dynamic pendulum without friction (brass, 2 washers).

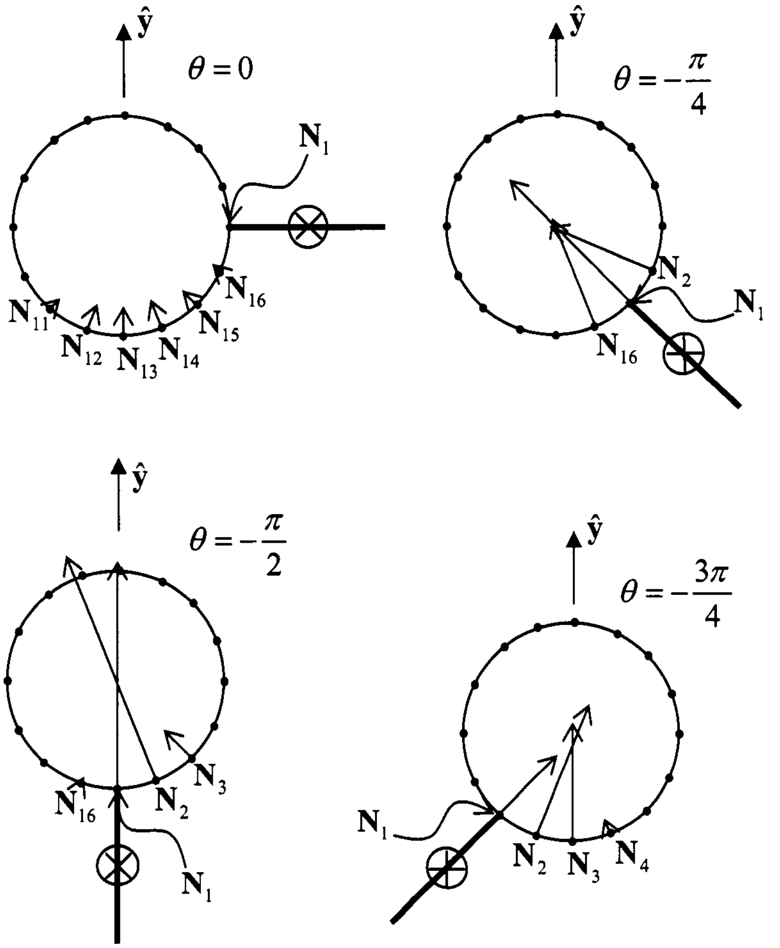

Figure 4.18. Predicted distribution of normal load for a dynamic pendulum with $\mu = 0.115$ and $b = 1.3$ (brass, 2 washers).

We can make some sense of these results by qualitatively analyzing the force distribution in the quasistatic case: at the bottom of its trajectory when the pendulum is in its vertical orientation, the velocity of the center of mass is at its maximum (or nearly so), and so the acceleration $\dot{\mathbf{u}}$ is nearly zero. The net force on the pendulum must therefore be very close to zero. If the normal forces were symmetrically distributed about the y-axis, then the frictional forces would exert a net *rightward* force on the pendulum. Since symmetrically distributed normal forces could only exert a net upward force, there would be no other force to counteract the frictional force, which contradicts our assumption of zero net force. By shifting the distribution of normal forces to the right of the pendulum, we can achieve force balance.

To be sure, the analysis above is not precise, and numerical simulations predict that the maximum rotational velocity is not achieved until slightly after the pendulum has reached its vertical orientation. Nevertheless, for the low friction coefficients of the materials we tested, the difference was slight, and the argument should hold for slight perturbations from the vertical.

To our knowledge, there is no closed-form description of the continuously distributed normal load in the presence of dry friction which could be used to validate this result. It might be possible to check these results numerically using finite-element modeling or check them experimentally using localized micro-pressure measurements. If these predictions are accurate, it could have significant implications for analyzing fatigue and fracturing in rigid materials.

Numerical simulations of hydrodynamic journal bearings produce similar asymmetry in the normal load distribution; however, the displacement in these systems *leads* the journal rotation (instead of lagging behind it as in our system) because the pressure buildup is due to a thin film surrounding the journal within the bearing (see [66] p. 28-1 or [12] p. 616).

Finally, we note that solutions other than the minimum Euclidean norm solution may contain nonzero contact forces on the side *opposite* the pendulum washers, which makes no physical sense. This provides some evidence that the minimum Euclidean norm solution is a physically reasonable choice; however, there may be other non-Euclidean minimum norm solutions that produce reasonable force distributions as well.

4.4 Summary and Future Research

We have presented an optimality formulation for computing the contact forces of a rigid mechanical system with multiple unilateral contacts and kinetic friction. This formulation applies Gauss' Principle of Least Constraint to formulate the contact forces as the minimum Euclidean norm solution of a convex quadratic program. The primary advantage of this approach is that when the problem is feasible the solution space is *convex*, making it tractable to compute the minimum norm solution. This is in contrast to the LCP formulation, which may have a nonconvex solution space for certain configurations.

In order to extend Gauss' Principle to discrete systems with redundant constraints, we have chosen to use the minimum Euclidean norm solution of the convex quadratic program. This allows us to analyze discretizations of continuously distributed contact forces. Examining numerically predicted force distributions for our pendulum device without friction seems to validate this as a physically reasonable choice. While the numerically predicted force distribution in the presence of friction may be somewhat counter-intuitive, these results can be explained using force balance in the quasistatic configuration. Further experimentation is necessary in order to establish this as the physically correct choice.

We have also demonstrated how this formulation can be easily used to construct explicit Runge-Kutta methods of arbitrary order. Our numerical scheme (a 5^{th}-order embedded method with automatic time step adjustment) was used to accurately predict the motion of an experimental pendulum with an unlubricated axle of nonzero radius. Friction coefficients were calibrated for four different material interfaces and were found to be in reasonable agreement between the single and double-washer configurations.

If exceptionally high-precision were necessary, it seems easy to extend our approach to explicit methods with embedded extrapolation formulas. In order to simulate mechanical systems with a high degree of stiffness, it would also seem necesary to employ an implicit scheme. Since an optimization problem must be solved at each stage, devising an implicit Runge-Kutta method seems nontrivial; however, it might be possible to devise a projected implicit Runge-Kutta (PIRK) method for these types of problems [9]. It might also be possible to devise a velocity-impulse based time-stepping scheme (as is done in [7, 22, 59, 64]), solving a convex QP at each time step rather than an LCP.

In order to model static friction, many authors have applied the Maximum Dissipation Principle [50] in conjunction with a polyhedral approximation of the friction cone to form a mixed LCP. It seems likely that a similar approach could be applied to augment the quadratic program in our Least Constraint formulation. For our particular experimental system, our numerical method accurately predicted the angle at which the pendulum came to rest in most cases. In many systems, however, the coefficient of static friction can be appreciably greater than the kinetic friction coefficient, in which case stick-slip transitions would be more pronounced.

On a final note, it remains to prove that a solution of Gauss' Principle is also a solution of the contact LCP in equation (2.36). This inclusion is trivially satisfied in our experimental device, and we have shown it to be *strict* for Painleve's example. Whether or not this inclusion holds for all problems or just a subset of contact problems remains to be explored.

Bibliography

[1] S. Abbott, *Understanding Analysis*, Springer-Verlag, New York, 2001.

[2] M. Anitescu, F. Potra, "Formulating Dynamic Multi-Rigid-Body Contact Problems with Friction as Solvable Linear Complementarity Problems," *Nonlinear Dynamics*, **14**: pp. 231 – 247, 1997.

[3] M. Anitescu, F. Potra, D.E. Stewart, "Time-stepping for Three-Dimensional Rigid Body Dynamics," *Computer Methods in Applied Mechanics and Engineering*, **177**: pp. 183 – 197, 1997.

[4] M. Anitescu, F. Potra, "A Time-Stepping Method for Stiff Multibody Dynamics with Contact and Friction," *International Journal for Numerical Methods in Engineering*, **55**: pp. 753 – 784, 2002.

[5] M. Anitescu, "A Fixed Time-Step Approach for Multibody Dynamics with Contact and Friction," in *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3725 – 3731, 2003.

[6] M. Anitescu, G.D. Hart, "A Fixed-Point Iteration Approach for Multibody Dynamics with Contact and Small Friction," *Mathematical Programming, Series B*, **101**: pp. 3 – 32, 2004.

[7] M. Anitescu, "Optimization-based Simulation of Nonsmooth Rigid Multibody Dynamics," *Mathematical Programming, Series A*, **105**: pp. 113 – 143, 2006.

[8] B. Armstrong-Helouvry, P. Dupont, C. Canudas de Wit, "A Survey of Models, Analysis Tools, and Compensation Methods for the Control of Machines with Friction," *Automatica*, **30**(7): pp. 1083 – 1138, 1994.

[9] U. Ascher, L. Petzold, "Projected Implicit Runge-Kutta Methods for Differential-Algebraic Equations," *SIAM Journal on Numerical Analysis*, **28**(4): pp. 1097 – 1120, 1991.

[10] U. Ascher, L. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, Philadelphia, 1998.

[11] Avallone, Baumeister, eds, *Mark's Standard Handbook for Mechanical Engineers*, 10th Edition, McGraw-Hill, New York, 1996.

[12] R.G. Budynas, J.K. Nisbett, eds., *Shigley's Mechanical Engineering Design*, 8th Edition, McGraw-Hill, New York, 2008.

[13] B. Brogliato, *Nonsmooth Mechanics*, 2nd Edition, Springer-Verlag, London, 1999.

[14] B. Brogliato, A.A. ten Dam, L. Paoli, F. Genot, M. Abadie, "Numerical Simulation of Finite Dimensional Multibody Nonsmooth Mechanical Systems," *Applied Mechanics Reviews*, **55**(2), pp. 107 – 150, 2002.

[15] K.E. Brennan, S.L. Campbell, L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, 2nd edition, SIAM, Philadelphia, 1996.

[16] J.R. Cash, A.H. Karp, "A Variable Order Runge-Kutta Method for Initial Value Problems with Rapidly Varying Right-Hand Sides." *ACM Transactions on Mathematical Software (TOMS)*, **16**(3), pp. 201 – 222, 1990.

[17] T.L. Chow, *Classical Mechanics*, John Wiley & Sons, New York, 1995.

[18] F.H. Clarke, *Optimization and Nonsmooth Analysis*, 2nd edition, SIAM, Philadelphia, 1990.

[19] E.A. Coddington, N. Levinson, *Theory of Ordinary Differential Equations*, Krieger Publishing, Malabar, 1984.

[20] E.A. Coddington, *An Introduction to Ordinary Differential Equations*, Dover Publications, New York, 1989.

[21] R.W. Cottle, J.S. Pang, R.E. Stone, *The Linear Complementarity Problem*, Academic Press, San Diego, 1992.

[22] A. Curnier, "Unilateral Contact – Problem Formulation," in *Multibody Dynamics with Unilateral Contacts*, F. Pfeiffer, Ch. Glocker, eds. CISM, Udine, 2000.

[23] H.F. Davis, A.D. Snider, *Introduction to Vector Analysis*, 6th edition, Wm. C. Browne Publishers, Dubuque, 1991.

[24] K.F. Gauss, "Uber ein neues allgemeines grundgesatz der mechanic," *Journal fur die Reine und Angewandte Mathematik*, **4**, pp. 232 – 235, 1829.

[25] C.W. Gear, G.K. Gupta, B. Leimkuhler, "Automatic Integration of Euler-Lagrange Equations with Constraints," *Journal of Computational and Applied Mechanics*, **12-13**, pp. 77 – 90, 1985.

[26] F. Genot, B. Brogliato, "On the Well-Posedness of Painleve's Example," in *IUTAM Symposium on Unilateral Multibody Contacts*, F. Pfeiffer and Ch. Glocker, eds., Kluwer Academic, Dordrecht, Boston, 1998.

[27] J.W. Gibbs, "On the Fundamental Formulae of Dynamics," *American Journal of Mathematics*, **2**(1), pp. 49 – 64, 1879.

[28] Ch. Glocker, "Formulation of Spatial Contact Situations in Rigid Multibody Systems," *Computer Methods in Applied Mechanics and Engineering*, **177**, pp. 199–214, 1999.

[29] Glocker, Ch., "Scalar Force Potentials in Rigid Multibody Systems," in *Multibody Dynamics with Unilateral Contacts*, F. Pfeiffer and Ch. Glocker, eds., Springer-Verlag, New York, 2000.

[30] Glocker, Ch., *Set-Valued Force Laws: Dynamics of Nonsmooth Systems*, F. Pfeiffer, ed., Springer, Berlin, 2001

[31] H. Goldstein, C.P. Poole, J.L. Safko, *Classical Mechanics*, 3rd edition, Pearson Education, Boston, 2001.

[32] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.

[33] E. Hairer, Ch. Lubich, M. Roche, "Error of Runge-Kutta methods for stiff problems studied via differential algebraic equations," *BIT*, **28**(3): pp. 678–700, 1988.

[34] E. Hairer, Ch. Lubich, M. Roche, *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*, A. Dold, B. Eckman, F. Takens, eds., Springer-Verlag, Berlin, 1989.

[35] E. Hairer, S.P. Norsett, G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, 2nd edition, Springer-Verlag, Berlin, 1993.

[36] E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Systems*, 2nd edition, Springer-Verlag, Berlin, 1996.

[37] R.A. Horn, C.R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[38] R.A. Horn, C.R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991.

[39] Lamarque, C.-H., Bastien, J., "Numerical Study of a Forced Pendulum with Friction," *Nonlinear Dynamics*, **23**(12), pp. 335–352, 2000.

[40] C. Lanczos, *The Variational Principles of Mechanics*, 4th edition, Dover, New York, 1986.

[41] R.J. Leveque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, SIAM, Philadelphia, 2007.

[42] P. Lotstedt, "Coulomb Friction in Two-dimensional Rigid-body Systems," *Z. Angew. Math. Mech.*, **61**, pp. 605–615, 1981.

[43] P. Lotstedt, "Mechanical Systems of Rigid Bodies Subject to Unilateral Constraints," *SIAM J. Appl. Math.*, **42**, pp. 281–296, 1982.

[44] Ch. Lubich, "On Projected Runge-Kutta Methods for Differential-Algebraic Equations," *BIT*, **31**, pp. 545–550, 1991.

[45] Ch. Lubich, "Extrapolation Integrators for Constrained Multibody Systems," *Impact of Computing in Science and Engineering*, 3, pp. 213–234, 1991.

[46] Ch. Lubich, Ch. Engstler, U. Nowak, U. Pohle, "Numerical Integration of Constrained Mechanical Systems using MEXX," *Mechanics of Structures and Machines*, **23**(4), pp. 473–495, 1995.

[47] E. Meyer, R.M. Overney, K. Dransfeld, T. Gyalog, *Nanoscience: Friction and Rheology on the Nanometer Scale*, World Scientific, Singapore, 1998.

[48] J.J. Moreau, "Les Liaisons Unilatérales et le Principe de Gauss," *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences*, **256**(4), pp. 871–874, 1963.

[49] J.J. Moreau, "Quadratic Programming in Mechanics: Dynamics of One-sided Constraints," *SIAM Journal on Control*, **4**(1), pp. 153–158, 1966.

[50] J.J. Moreau, "Sur les Lois de Frottement, de Plasticité et de Viscosité," *Comptes Rendus, série A*, **271**, pp. 608–611, 1970.

[51] J.J. Moreau, "Liaisons Unilatérales sans Frottement et Chocs Inélastiques," *Comptes Rendus des Séances de l'Académie des Sciences, Série II*, **296**(19), pp. 1473 - 1476, 1983.

[52] J.J. Moreau, "Une Formulation du Contact a Frottement Sec: Application au Calcul Numérique," *Comptes Rendus des Séances de l'Académie des Sciences, Série II*, **302**(13), pp. 799–801, 1986.

[53] J.J. Moreau, "Une Formulation de la Dynamique Classique," *Comptes Rendus des Séances de l'Académie des Sciences, Série II*, **304**(5), pp. 191–194, 1987.

[54] J.J. Moreau, "Numerical Aspects of the Sweeping Process," Computer Methods in Applied Mechanics and Engineering, **177**, pp. 329–349, 1999.

[55] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.

[56] P. Painleve, "Sur les Lois du Frottement de Glissement," *Comptes Rendus de l'Acade'mie des Sciences*, **121**, pp. 112–115, 1895.

[57] J.G. Papastavridis, *Analytical Mechanics: A Comprehensive Treatise on the Dynamics of Constrained Systems for Engineers, Physicists, and Mathematicians*, Oxford University Press, Oxford, 2002.

[58] F. Pfeiffer, Ch. Glocker, *Multibody Dynamics with Unilateral Contacts*, Wiley, New York, 1996.

[59] F.A. Potra, M. Anitescu, B. Gavrea, J. Trinkle, "A Linearly Implicit Trapezoidal Method for Integrating Stiff Multibody Dynamics with Contact, Joints, and Friction," *International Journal for Numerical Methods in Engineering*, **66**(7), pp. 1079–1124, 2006.

[60] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C++: The Art of Scientific Computing*, 2nd Edition, Cambridge University Press, Cambridge, 2002.

[61] R.T. Rockafellar, *Fundamentals of Optimization*, course notes, 2007, available online at http://www.math.washington.edu/~rtr/fundamentals.pdf.

[62] P. Song, P. Kraus, V. Kumar, P. Dupont, "Analysis of Rigid-Body Dynamic Models for Simulation of Systems with Frictional Contacts", *Trans. ASME J. Appl. Mech.*, **68**, pp. 118–128, 2001.

[63] D.E. Stewart, J. Trinkle, "An Implicit Time-stepping Scheme for Rigid-Body Dynamics with Inelastic Collisions and Coulomb Friction," *International Journal for Numerical Methods in Engineering*, 39, pp. 2673 – 2691, 1996.

[64] D.E. Stewart, "Convergence of a Time-Stepping Scheme for Rigid-Body Dynamics and Resolution of Painleve's Problem," *Arch. Rational Mech. Anal*, 145, pp. 215 – 260, 1998.

[65] D.E. Stewart, "Rigid-Body Dynamics with Friction and Impact." *SIAM Review*, **42**(1), pp. 3–39, 2000.

[66] J.E. Shigley, C.R. Mischke, eds., *Standard Handbook of Machine Design*, McGraw-Hill, New York, 1986.

[67] L. Trefethen, D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[68] J. Trinkle, J. Pang, S. Sudarsky, G. Lo. "On Dynamic Multi-Rigid-Body Contact Problems with Coulomb Friction," *Zeitschrift fur Angewandte Mathematik und Mechanik*, vol. 77, no. 4, pp. 267-279. 1997.

[69] J. Trinkle, J. Tzitzouris, J. Pang. "Dynamic Multi-Rigid-Body Systems with Concurrent Distributed Contacts: Theory and Examples," *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 359 (1789) pp. 2575-2593. 2001.

[70] J. Trinkle. "Formulation of Multibody Dynamics as Complementarity Problems," in *Proceedings of the ASME 2003 Design Engineering Technical Conference and Computers and Information in Engineering Conference*, Chicago, IL. Sept. 2-6, 2003.

[71] J.J. Tuma, R.A. Walsh, *Engineering Mathematics Handbook*, 4th Edition, McGraw Hill, New York, 1998.

[72] F.E. Udwadia, R.E. Kalaba, *Analytical Dynamics: a New Approach*, Cambridge University Press, Cambridge, UK, 1996.

[73] F.E. Udwadia, "New General Principle of Mechanics and its Application to General Nonideal Nonholonomic Systems," *Journal of Engineering Mechanics*, 131(4), pp. 444 – 450, 2005.

[74] E.V. Wilms, H. Cohen, "Planar Motion of a Rigid Body with a Friction Rotor," *Journal of Applied Mechanics*, 48, pp. 205 – 206, 1981.

[75] E.V. Wilms, H. Cohen, "Limitations of the Coulomb friction model for two elementary dynamical systems," *International Journal of Mechanical Engineering Education*, 11(2), pp. 107 – 112, 1982.

[76] E.V. Wilms, H. Cohen, "The Occurrence of Painleve's Paradox in the Motion of a Rotating Shaft," *Journal of Applied Mechanics*, v. 64, pp. 1008-1010. December 1997.

[77] P. Wriggers, *Computational Contact Mechanics*. John Wiley & Sons Ltd. 2002.

Appendix A : Code Listings

A.1 C++ Listings

The following C++ code was used to extract the pendulum angle as a function of time from an ASCII file of the 2-channel optical encoder analog signal:

```
///////////////////////////////////////////////////////////////////////////////
//
// Copyright 2006-2008, Miguel J. Gomez
//
// Convert raw data from 2-channel optical scanner into theta(t) for the
// pendulum experimental device.
//
#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include <cmath>

using namespace std;

//fixed-size data types
#ifdef WIN32

    //integers
    typedef char          int8;
    typedef unsigned char uint8;
    typedef short         int16;
    typedef unsigned short uint16;
    typedef int                int32;
    typedef unsigned int  uint32;

    //floating point numbers
    typedef float         float32;
    typedef double             float64;

#endif   //WIN32


//constants
const float64 PI = 4.0 * atan(1.0);

///////////////////////////////////////////////////////////////////////////////
//Function Prototypes
///////////////////////////////////////////////////////////////////////////////

///////////////////////////////////////////////////////////////////////////////
void write_theta_t_file
(
    const char*   in_file_path, //name of raw data file
    const float64 freq          //sample frequency, kHz
);


///////////////////////////////////////////////////////////////////////////////
void main( int32 argc, char* argv[] )
```

```
{
    //if init file was not specified, warn user
    if( 2 != argc )
    {
        cerr << "ERROR:  Specify exactly one initialization file path.";

        return;
    }

    //initialization file name
    const char* init_file_name = argv[1];
    //open the init file
    ifstream init_file;

    init_file.open( init_file_name );

    //if file could not be opened, warn user
    if( !init_file )
    {
        cerr << "ERROR:  Could not open initialization file:\n\""
                 << init_file_name << "\".";

        return;
    }

    //read each lines in the file
    while( !init_file.eof() )
    {
        //character buffer
        char buf[2048];

        //read line
        init_file.getline( buf, 2048 );

        //ignore empty lines
        if( !strcmp( "", buf ) )
            continue;

        //if first two letters of this word are "//",
        //then it's a single-line comment
        if( '/'==buf[0] && '/'==buf[1] )
            continue;

        //unparsed file line
        stringstream line( buf );
        //data file path
        string path;
        //sample frequency in kHz
        float64 freq;

        //NOTE:  Assume the line was formatted correctly
        line >> path >> freq;

        //if frequency was not valid, warn user
        if( freq<1 || 50<freq )
        {
            cerr << "ERROR:  Sample frequencies must be between 1 and 50 kHz.";

            //close file and bail out
            init_file.close();
            return;
        }
```

```
        //convert the raw voltage data from channels 1 and 2 to theta(t)
        write_theta_t_file( path.c_str(), freq );
    }
}


//////////////////////////////////////////////////////////////////////////////
void write_theta_t_file
(
    const char*   in_file_path, //name of raw data file
    const float64 freq      //sample frequency, kHz
)
{
    //raw data in_file
    ifstream in_file;

    in_file.open( in_file_path );

    if( !in_file )
    {
        cerr <<"ERROR:  Could not open in_file \"" << in_file_path << "\".";
        return;
    }

    //analog voltage from channels 1 and 2
    vector<uint8> ch1, ch2;

    //read until end of file
    while( !in_file.eof() )
    {
        uint32 c1, c2;

        in_file >> c1 >> c2;

        ch1.push_back( (uint8)c1 );
        ch2.push_back( (uint8)c2 );
    }

    //close the in_file
    in_file.close();

    //make sure each channel had same amount of data
    if( ch1.size() != ch2.size() )
    {
        cerr << "ERROR:  File \"" << in_file_path << "\"\n";
        cerr <<
             "does not have the same amount of samples from channels 1 and 2.";
        return;
    }

    //look for channel transitions and
    //motion reversals to infer theta(t)

    //angle increment between channel transitions, in radians
    //500 counts, 2 channels -> 1000 transitions per revolution
    float64 dtheta = -(2*PI) / 1000;
    //initial angle in radians
    float64 theta = 0;   //perfectly horizontal
    //time per sample, sec
    const float64 dt = 1e-3 / freq;
    //starting time, sec
```

```
float64 t = 0;
//total number of samples
const uint32 count = (uint32)ch1.size();
//allocate time and theta arrays
vector<float64> times, thetas;

     times.reserve( count );
     thetas.reserve( count );

     //initial values
     times.push_back( t );
     thetas.push_back( theta );

//previous channel values
uint8 c1_prev = ch1[0];
uint8 c2_prev = ch2[0];
//true if channel 1 was the
//last channel to transition
bool bCh1 = true;
//flag for finding when motion actually starts
bool bFirstTransition = true;

//start at 2nd data point
for( uint32 i=1 ; i<count ; i++ )
{
     //voltages from channels 1 and 2
     const uint8 c1 = ch1[i];
     const uint8 c2 = ch2[i];

     //increment time
     t += dt;

     //check if channel 1 is transitioning (uptick or downtick)
     if( (c1>0 && c1_prev<1) || (c1<1 && c1_prev>0) )
     {
          if( bFirstTransition )
          {
               //start time
               t = 0;

               //increment theta by 1/2 tick angle
               theta += 0.5*dtheta;

               //unset this flag
               bFirstTransition = false;
          }
          //if channel 1 was the previous channel to
          //transition, as well, then the pendulum
          //direction has reversed
          else if( bCh1 )
          {
               //negate the angle increment
               dtheta = -dtheta;
          }
          else
          {
               //increment the angle
               theta += dtheta;

               //add new data point
               times.push_back( t );
               thetas.push_back( theta );
```

```
        }

        //set this flag AFTER the above if statements
        bCh1 = true;
    }
    else if( (c2>0 && c2_prev<1) || (c2<1 && c2_prev>0) )
    {
                //channel 2 is transitioning
        if( bFirstTransition )
        {
            //restart the timer
            t = 0;

            //increment theta by 1/2 tick angle
            theta += 0.5*dtheta;

            //unset this flag
            bFirstTransition = false;
        }
        //if channel 2 was the previous channel to
        //transition, as well, then the pendulum
        //direction has reversed
        else if( !bCh1 )
        {
            //negate the angle increment
            dtheta = -dtheta;
        }
        else
        {
            //increment the angle
            theta += dtheta;

            //add new data point
            times.push_back( t );
            thetas.push_back( theta );
        }

        //set this flag AFTER the above if statements
        bCh1 = false;
    }

    //store previous values
    c1_prev = c1;
    c2_prev = c2;
}

//output file path
char out_file_path[1024];

strcpy( out_file_path, in_file_path );

char* p = strchr( out_file_path, '.' );

//terminate the string at this point
p[0] = 0;

//extract the file name without the path
char name[1024];

strcpy( name, out_file_path );

//look for the last '\' or '/' character
```

```
    const char bs = '\\';
    const char fs = '/';
    const int sc = (int)strlen(name) - 1;

    for( int i = sc ; i>=0 ; i-- )
    {
        if( (bs == name[i]) || (fs == name[i]) )
        {
            //copy string
            strcpy( name, name+(i+1) );

            break;
        }
    }

    //add new extension to path
    strcat( out_file_path, "_out.txt" );

    //open output file
    ofstream out_file( out_file_path );

    //write column headers
    out_file << "Time(sec)\t" << name << "\n";

    for( uint32 i=0 ; i<times.size() ; i++ )
    {
        //write out each data point
        out_file << times[i] << "\t" << thetas[i] << "\n";
    }

    //close file
    out_file.close();
}

//EOF
////////////////////////////////////////////////////////////////////////////////
```

A.2 Matlab Listings

The following Matlab code was used to simulate the motion of the pendulum and compare numerical results to the nonlinear pendulum with load-independent friction:

Pend.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Copyright 2005-2008, Miguel J. Gomez
%
% Simulate pendulum device for several different materials.
%
% NOTE:  Dimensionless equations of motion are solved, but input parameters
% and output data are in SI (meters-kg-sec).
%

%clear workspace
%NOTE:  'clear all' removes breakpoints!
```

```
clear;

%show double precision
format long g;

%grav accel, m/sec^2
global g;
g = 9.80665;

%number of contact points
global m_s;

%NOTE:  16 contact points seems to be sufficient.
m_s = 16;

%radius of pivot
global rp;
rp = 0.01; %1 cm

%height of borehole where rod screws into pivot
hh = 2*rp;

%"height" of pivot cylinder
hp = 0.0826;  %8.26 cm

%radius of the rod (to which washers are fastened)
rr = 0.0048;  %0.48 cm

%masses of each pivot
% mp = 0.1960;   %196 g, brass
% mp = 0.0275;   % 27.5 g, nylon
 mp = 0.1880;   %188 g, steel (stainless)
% mp = 0.0520;   % 52 g, teflon

%moment of inertia of the pivot about its radial axis, taking
%into account the borehole where the rod screws in
% Jp = 0.5*mp*rp^2;
Jp = 0.5*mp*(hp*rp^4 - (1.0/6)*hh*rr^2*(3*rr^2 + hh^2)) / (hp*rp^2 - hh*rr^2);
%NOTE:  Subtracting out the hollow space where the
%rod screws in does not make a noticeable difference.

%heights of lower (threaded) and
%upper (unthreaded) portions of rod
hrt = 0.0375;  %threaded,    3.75 cm
hr = 0.1470; %total height of rod, 14.70 cm
hru = hr - hrt; %unthreaded

%mass of the rod (1018 steel)
mr = 0.074; %74 g
%moment of the rod about its CM
%(cylinder about its transverse axis)
Jr = 0.25*mr*(rr^2) + (1.0/12)*mr*(hr^2);

%inner/outer radius of each washer
rwi = rr;
rwo = 0.015;  %1.5 cm

%number of washers (1 or 2)
% n = 1;
 n = 2;
%height of n stacked washers
hw = 0.0187 * n;  %1.87 cm per washer
```

```
%mass of n washers
mw = 0.0905 * n; %90.5 g per washer
Jw = (1.0/12)*mw*(3*(rwi^2 + rwo^2) + hw^2);

%dist from z-axis to CM of pivot along -y
lp = 0;

%dist from z-axis to CM of rod along -y
lr = 0.5*hr;%NOTE:  True if and only if the rod extends half-way
            %through the pivot cylinder, which it currently does.

%CM of n stacked washers along -y direction
lw = hru + 0.5*hw;

%total mass
global mass;
mass = mp + mr + mw;

%dist from pivot axis to CM
global l;
l = (lp*mp + lr*mr + lw*mw) / mass;

%total moment of inertia about CM
%(Parallel axis theorem)
global J;
J = (Jp + mp*(l-lp)^2) + (Jr + mr*(l-lr)^2) + (Jw + mw*(l-lw)^2);

%dimensionless parameters
alpha = J / (mass*l^2);
global beta;
beta = rp / l;

%dimensionless mass matrix, [ mass ; J ]
mJ = [ 1/(1+alpha) ; alpha/(1+alpha) ];

%approximate sliding friction coefficients
%of various materials on aluminum support
%plates
global mu;
% mu = 0.0;    %no friction
% mu = 0.16;    %brass range [0.115, 0.29]
% mu = 0.145;   %nylon range [0.08, 0.15]
 mu = 0.37;    %steel range [0.14, 0.35]
% mu = 0.33;    %teflon range [0.08, 0.18]

%linear viscosity term
global b;
% b = 0;  %no friction
% b = 0.0;  %brass range [1.3, 0.0]
% b = 0.8;  %nylon range [0.8, 1.3]
 b = 2.5;  %steel range [1.5, 2.5]
% b = 5.0;  %teflon range [4.5, 5.0]

%starting time
t0 = 0;

%initial angle
theta0 = 0;     %horizontal
%theta0 = -pi/2; %straight down

%consistent initial generalized position (dimensionless)
```

```
x0 = [ cos(theta0) ; sin(theta0) ; theta0 ];

%initial angular velocity
w0 = -1e-9; %HACK:  ensure that sign(v) = -1
            %when released from theta = 0

%initial CM velocity
v0 = [0,0]';%BUGBUG:  CM is not stationary if w0!=0

%initial generalized velocity
u0 = [ v0 ; w0 ];

%absolute and relative local truncation error tolerances
atol = 1e-6;
rtol = 1e-6;
%atol = 1e-1;%ensure dt_max is used
%rtol = 1e-1;

%initial and max time step (dimensionless)
dt0 = 1e-2;
%dt0 = 1e-6;
dt_max = 0.25; %max time step
%dt_max = 0.01; %use this to monitor N[i] transitions
%dt_max = 0.001; %use this to monitor N[i] transitions

%SHO frequency
global K;
K = sqrt( g / (l * (1+alpha)) );
%simulation time (seconds)
T = 2.75 * K;

%function handles
eval_H = @SingleBody2D_H;
eval_G = @Pend_G;
eval_Gx = @Pend_Gx;
eval_dGxH_dt = @Pend_dGxH_dt;
eval_W = @Pend_W;
eval_Fnc = @Pend_Fnc;
eval_M = @MassMatrix2D;
eval_Minv = @MassMatrix2D_Inv;

%integrate motion using...

%...My solver
[pT, pX, pU, pN, pF, pG, pGxHu] = RK54_CashKarp( atol, rtol,...
                                            T, dt0, dt_max,...
                                            t0, x0, u0,...
                                            eval_H, eval_G,...
                                            eval_Gx, eval_dGxH_dt,...
                                            eval_W, eval_Fnc,...
                                            eval_M, eval_Minv, mJ );

%...Matlab's ode45() solver
% y0 = [ theta0, w0 ];
% options = odeset( 'RelTol', rtol, 'AbsTol', atol );
% [t,y] = ode45( @Pend_f_ode45, [0,T], y0, options );
% th_ode45 = y(:,1);

% %...Analytic SHO solution for small deviations from theta = -pi/2
% A = theta0 + pi/2;
% th_sho = A*cos( K * pT ) - pi/2;
```

```
%convert back to dimensional variables
T = (1/K)*T;            %duration, sec
pT = (1/K)*pT;          %time, seconds
pX(:,1) = l*pX(:,1);    %rx, meters
pX(:,2) = l*pX(:,2);    %ry, meters
pN = mass*g*pN;         %normal force, Newtons
pF = mass*g*pF;         %friction force, Newtons
pG = l*pG;              %penetration depth, meters

%title string
tstr = ['Pendulum Angle, \it{\mu} = ' num2str(mu)...
        ', \it{b} = ' num2str(b) ];

%plot theta(t)
figure(1);
%just my solution
plot( pT, pX(:,3), '-o' );
%compare to Matlab ode45()
% plot( t, th_ode45, '-+', pT, pX(:,3), '-o' );
% legend( 'Matlab ode45()','My Solution' );
%compare to analytic SHO and nonlinear soln from Matlab ode45()
% plot( t, th_ode45, '-+', pT, pX(:,3), '-o', pT, th_sho, '-x' );
% legend( 'Matlab ode45()','My Solution', 'Analytic SHO Solution' );
xlabel( '\it{t}, sec' );
ylabel( '\it{\theta}(\it{t}), rad/pi' );
title( tstr );

%save theta(t) to an ASCII file
%for comparing to actual data
file = fopen( 'sim_output.txt', 'w' );
fprintf( file, ['Time(sec)\tSim (mu=' num2str(mu) ', b=' num2str(b) ')\n'] );
%my solution
for i=1:length(pT)
    fprintf( file, '%f\t%f\n', pT(i), pX(i,3) );
end
fclose( file );
%Matlab ode45()
% for i=1:length(t)
%     fprintf( file, '%f\t%f\n', t(i), th_ode45(i) );
% end

%plot window
figure(2);

%plot theta(t)
subplot( 4,1, 1 );
plot( pT, pX(:,3), '-+' );
xlabel( 't, sec' );
ylabel( '\theta(t), rad/\pi' );
title( tstr );

%plot dtheta/dt
subplot( 4,1, 2 );
plot( pT, pU(:,3), '-+' );
xlabel( 't, sec' );
ylabel( 'd\theta(t)/dt, rad/(\pi*sec)' );

%plot G(x(t))
subplot( 4,1, 3 );
plot( pT, pG, '-' );
title( 'Constraint error' );
xlabel( 't, sec' );
```

```
ylabel( 'G(x(t)), m' );

%plot G_x*H*u(t)
subplot( 4,1, 4 );
plot( pT, pGxHu, '-' );
title( 'Velocity constraint error' );
xlabel( 't, sec' );
ylabel( 'G_xH*u(t), m/s' );

%plot N1, N2, N3, and N4 vs. time
%to see if it looks reasonable
figure(3);

subplot( 8,1, 1 );
plot( pT, pN(:,1), '-+' );
xlabel( 't, sec' );
ylabel( 'N_1, Nt' );

subplot( 8,1, 2 );
plot( pT, pN(:,2), '-+' );
xlabel( 't, sec' );
ylabel( 'N_2, Nt' );

subplot( 8,1, 3 );
plot( pT, pN(:,3), '-+' );
xlabel( 't, sec' );
ylabel( 'N_3, Nt' );

subplot( 8,1, 4 );
plot( pT, pN(:,4), '-+' );
xlabel( 't, sec' );
ylabel( 'N_4, Nt' );

subplot( 8,1, 5 );
plot( pT, pN(:,5), '-+' );
xlabel( 't, sec' );
ylabel( 'N_5, Nt' );

subplot( 8,1, 6 );
plot( pT, pN(:,6), '-+' );
xlabel( 't, sec' );
ylabel( 'N_6, Nt' );

subplot( 8,1, 7 );
plot( pT, pN(:,7), '-+' );
xlabel( 't, sec' );
ylabel( 'N_7, Nt' );

subplot( 8,1, 8 );
plot( pT, pN(:,8), '-+' );
xlabel( 't, sec' );
ylabel( 'N_8, Nt' );

%save N(theta) to an ASCII file at various times to
%examine how normal force is distributed about axle
ts = [1e-6, 0.14, 0.21, 0.28];
n_data = length(pT);
file = fopen( 'N_vs_theta.txt', 'w' );
%parameters
fprintf( file, 'Brass (mu=%f, b=%f)\n', mu, b );
%time of snapshot
t = ts(4);
```

```
for index=1:n_data;
    if t<=pT(index)
        t = pT(index);%sim time
        break;
    end
end
%angle at time of snapshot
theta = pX(index,3);
fprintf( file, 't=%f, theta=%f\n\n', t, theta );
%column headers
fprintf( file, 'theta[i], N[i]\n' );
%theta[i] vs. N[i]
dtheta = (2*pi)/m_s;
pThi = [];
pNi = [];
for i=1:m_s
    th_i = theta + (i-1)*dtheta;
    %wrap
%     if th_i>(pi-0.01)
%         th_i = th_i - 2*pi;
%     end
    N_i = pN(index,i);
    pThi = [pThi, th_i];
    pNi = [pNi, N_i];
    fprintf( file, '%f\t%f\n', th_i, N_i );
end
fclose( file );

%test plot:
figure(4);
plot( pThi, pNi, '.', 'MarkerSize', 30 );
xlabel( '{\it\theta}, radians', 'FontSize', 20 );
ylabel( 'Nt' );
title( ['t = ' num2str(t) ' sec'], 'FontSize', 20 );

%EOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

RK54_CashKarp.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Copyright 2005-2008, Miguel J. Gomez
%
% Integate the motion of a set of rigid bodies with contacts
% experiencing sliding friction using a 6-stage 5(4) embedded
% Runge-Kutta formula developed by Cash and Karp.
%
% State variables:
%
% x = [ rx, ry, rz, Rxx, Rxy, ... , Rzy, Rzz ]^T, 12x1 configuration vector
%
% u = [ drx/dt, dry/dt, drz/dt, wx, wy, wz ]^T, 6x1 generalized velocity
% vector
%
% Velocity transfer matrix:
%
% H(x) = [ I; -Rx*; -Ry*; -Rz* ], 12x3 matrix, where Rx* is the cross-product
% matrix of local x-axis, Rx, and so on.
%
```

```
function [pT, pX, pU, pN, pF, pG, pGxHu ] = RK54_CashKarp(  atol, rtol,...
                                                         T, dt, dt_max,...
                                                         t0, x0, u0,...
                                                         eval_H,...
                                                         eval_G,...
                                                         eval_Gx,...
                                                         eval_dGxH_dt,...
                                                         eval_W,...
                                                         eval_Fnc,...
                                                         eval_M,...
                                                         eval_Minv,...
                                                         mJ )

%number of inequality constraints, m + s
G0 = eval_G(x0);
ms = length(G0);

%length of generalized state and velocity vectors
nx = length(x0);
n = length(u0);

%initial values
pT = t0;
pX = x0';
pU = u0';
pG = 0;    %assumed consistent at t0
pGxHu = 0; %assumed consistent at t0

%normal force
N0 = zeros( ms, 1 );
%HACK:  N0 can't be 0, or else first call to optimization
%routine gives grad(f)=0, which makes it stall at that point.
N0(2) = 1e-9;
pN = N0';

%friction x,y,theta components
F0 = zeros( 3, 1 );
pF = F0';

% Cash-Karp 6-stage, 5(4) embedded Runge-Kutta
% (Numerical Recipes in C, 1992), (Cash-Carp, 1985)
%
%constants:
%for k's
b21 = 1/5.0;

b31 = 3/40.0;
b32 = 9/40.0;

b41 = 3/10.0;
b42 = -9/10.0;
b43 = 6/5.0;

b51 = -11/54.0;
b52 = 5/2.0;
b53 = -70/27.0;
b54 = 35/27.0;

b61 = 1631/55296.0;
b62 = 175/512.0;
b63 = 575/13824.0;
b64 = 44275/110592.0;
```

```
b65 = 253/4096.0;

%4th order estimate
cs1 = 2825/27648.0;
cs2 = 0;
cs3 = 18575/48384.0;
cs4 = 13525/55296.0;
cs5 = 277/14336.0;
cs6 = 1/4.0;

%5th order estimate
c1 = 37/378.0;
c2 = 0;
c3 = 250/621.0;
c4 = 125/594.0;
c5 = 0;
c6 = 512/1771.0;

%initial time
t = t0;

%start at 2 so we can reference
%previous time step
i = 2;

while t<T

    %current state
    t = pT(i-1);
    x = pX(i-1,:)';
    u = pU(i-1,:)';
    N0 = pN(i-1,:)';
    F0 = pF(i-1,:)';

    %make sure that the while loop
    %below executes at least once
    err = 2;

    while err > 1

        %K1:
        [v1,a1,N1,F1] = RK_Stage(  eval_H, eval_G, eval_Gx, eval_dGxH_dt,...
                                eval_W, eval_Fnc, eval_M, eval_Minv, mJ,...
                                x, u, N0 );

        %K2:
        dx = b21*v1*dt;
        du = b21*a1*dt;

        %NOTE:  Project intermediate solution onto constraint manifold so
        %that the optimization subroutine has a consistent state vector.
        [x1,u1] = ProjectConstraint( eval_H, eval_G, eval_Gx, x+dx, u+du );

        [v2,a2,N2,F2] = RK_Stage(  eval_H, eval_G, eval_Gx, eval_dGxH_dt,...
                                eval_W, eval_Fnc, eval_M, eval_Minv, mJ,...
                                x1, u1, N1 );

        %K3:
        dx = (b31*v1 + b32*v2)*dt;
        du = (b31*a1 + b32*a2)*dt;

        [x2,u2] = ProjectConstraint( eval_H, eval_G, eval_Gx, x+dx, u+du );
```

```
        [v3,a3,N3,F3] = RK_Stage(  eval_H, eval_G, eval_Gx, eval_dGxH_dt,...
                                eval_W, eval_Fnc, eval_M, eval_Minv, mJ,...
                                x2, u2, N2 );

        %K4:
        dx = (b41*v1 + b42*v2 + b43*v3)*dt;
        du = (b41*a1 + b42*a2 + b43*a3)*dt;

        [x3,u3] = ProjectConstraint( eval_H, eval_G, eval_Gx, x+dx, u+du );

        [v4,a4,N4,F4] = RK_Stage(  eval_H, eval_G, eval_Gx, eval_dGxH_dt,...
                                eval_W, eval_Fnc, eval_M, eval_Minv, mJ,...
                                x3, u3, N3 );

        %K5:
        dx = (b51*v1 + b52*v2 + b53*v3 + b54*v4)*dt;
        du = (b51*a1 + b52*a2 + b53*a3 + b54*a4)*dt;

        [x4,u4] = ProjectConstraint( eval_H, eval_G, eval_Gx, x+dx, u+du );

        [v5,a5,N5,F5] = RK_Stage(  eval_H, eval_G, eval_Gx, eval_dGxH_dt,...
                                eval_W, eval_Fnc, eval_M, eval_Minv, mJ,...
                                x4, u4, N4 );

        %K6:
        dx = (b61*v1 + b62*v2 + b63*v3 + b64*v4 + b65*v5)*dt;
        du = (b61*a1 + b62*a2 + b63*a3 + b64*a4 + b65*a5)*dt;

        [x5,u5] = ProjectConstraint( eval_H, eval_G, eval_Gx, x+dx, u+du );

        [v6,a6,N6,F6] = RK_Stage(  eval_H, eval_G, eval_Gx, eval_dGxH_dt,...
                                eval_W, eval_Fnc, eval_M, eval_Minv, mJ,...
                                x5, u5, N5 );

        %normal force estimate
        N = N6;   %HACK:  Use the last estimate for visualization, since N1, N2,
                  %etc., will probably less accurately reflect the final state of
                  %N.
%        N = c1*N1 + c2*N2 + c3*N3 + c4*N4 + c5*N5 + c6*N6;

        %friction
        F = c1*F1 + c2*F2 + c3*F3 + c4*F4 + c5*F5 + c6*F6;

        %4th order estimate
        xs1 = x + dt*( cs1*v1 + cs2*v2 + cs3*v3 + cs4*v4 + cs5*v5 + cs6*v6 );
        us1 = u + dt*( cs1*a1 + cs2*a2 + cs3*a3 + cs4*a4 + cs5*a5 + cs6*a6 );

        %5th order estimate
        x1 = x + dt*( c1*v1 + c2*v2 + c3*v3 + c4*v4 + c5*v5 + c6*v6 );
        u1 = u + dt*( c1*a1 + c2*a2 + c3*a3 + c4*a4 + c5*a5 + c6*a6 );

        %NOTE:  only care about error in x (position and orientation),
        %error formula taken from Hairer, Norsett, Wanner vol. I, p. 168

        %max component among both x0 and x1
        x_max = norm( [x;x1], inf );

        %error metric [35]:
        sc = atol + rtol*x_max;
        err = norm( x1-xs1 ) / (sc * sqrt(nx) );
```

```
        %if [x1,u1] is acceptable, then record t now before
        %dt is adjusted for the following time step
        if err<1
            t = t + dt;
        end

        %adjust time step, but no larger than dt_max
        if err>0
            dt = min( dt_max, 0.75 * dt * (1/err)^0.2 );
        else
            dt = dt_max;
        end

        %don't overshoot T
        if t+dt>T
            if err<1 %[x1,u1] will be accepted
                dt = T - t; %last time step
            end
        end

    end %while err > 1

    %Project the final solution so that all the constraints are satisfied
    [x1,u1] = ProjectConstraint( eval_H, eval_G, eval_Gx, x1, u1 );

    %state variables
    pT = [pT ; t];
    pX = [pX ; x1'];
    pU = [pU ; u1'];

    %generalized normal force
    pN = [pN ; N'];

    %generalized frictional force
    pF = [pF ; F'];

    %eval spatial and velocity constraint errors
    G = eval_G( x1 );

    Gx = eval_Gx( x1 );
    H = eval_H( x1 );
    GxHu = Gx*H*u1;

    %max errors over all spatial and velocity constraints
    err_G = norm( G, inf );
    err_GxHu = norm( GxHu, inf );
    pG = [pG ; err_G];
    pGxHu = [pGxHu ; err_GxHu];

    %increment index
    i = i + 1;

end %while t<T

%EOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

RK_Stage.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Copyright 2005-2008, Miguel J. Gomez
%
% Compute the instantaneous v = dx/dt, acceleration a = du/dt, and normal
% force N for a system of bodies with frictional unilateral contact using
% Gauss' Principle of Least Constraint to find N.  This requires us to
% solve the nonlinear optimization problem:
%
% min 0.5*N'*S*N + eps*||N||
%
% subject to:
%
% b - A*N <= 0  %linear inequality constraints
% N >= 0        %range constraints
%
% where
%
% S = W*Minv*W'
% A = GxH*Minv*W'
%
% The initial value N0 is used to accelerate convergence of the
% optimization routine.
%
function [v,a,N,f] = RK_Stage(eval_H, eval_G,...
                            eval_Gx, eval_dGxH_dt,...
                            eval_W, eval_Fnc,...
                            eval_M, eval_Minv,...
                            mJ, x, u, N0 )

%length of generalized position and velocity
nx = length( x );
n = length( u );

%number of ineq cstrts, m + s
ms = length( N0 );

%velocity transfer matrix
H = eval_H( x );

%generalized mass matrix
M = eval_M( mJ, x );

%its inverse
Minv = eval_Minv( mJ, x );

%NOTE:  "sqg_" stands for "squiggle", the
%unreduced constraints with a '~' on top.

%wrench matrix
[sqg_W,sqg_Wf] = eval_W( x, u );

%constraint Jacobian matrix
sqg_Gx = eval_Gx( x );

%matrix product
sqg_GxH_Minv = sqg_Gx*H*Minv;

%generalized non-constraint force
Fnc = eval_Fnc( mJ, x, u );

%d(Gx*H)/dt
```

```
sqg_dGxH_dt = eval_dGxH_dt( x, u );

%First, solve the optimization problem for sqg_N
%NOTE:  Matlab's fmincon solves the following nonlinear optimization
%problem:
%
% min f(x)
%
% subject to:
%
% c(x) <= 0         %nonlinear ineq cstrts
% ceq(x) = 0        %nonlinear equality cstrts
% A*x <= b          %linear ineq cstrts
% Aeq*x = beq       %linear equality cstrts
% LB <= x <= UB     %range constraints
%

%Hessian mtx of quadratic term of obj_fn()
global S;
S = sqg_W * Minv * sqg_W';

A = -sqg_GxH_Minv * sqg_W'; %inequality cstrts
b = -sqg_GxH_Minv * Fnc - sqg_dGxH_dt * u;

Aeq = [];   %equality cstrts (none)
beq = [];
LB = zeros( ms, 1 );  %N >= 0
UB = inf( ms, 1 );    %no upper bound
nlc = [];   %nonlinear cstrts (empty)
%NOTE:  Set tolerances same as N_tol.
optns = optimset('LargeScale', 'off',...
                'Display', 'off',...
                'GradObj','on',...
                'TolFun', 1e-10,...
                'TolCon', 1e-10,...
                'TolX', 1e-10 );
%NOTE:  obj_fn (see bottom) is nonsmooth at N=0, but the cstrts bound the
%solution away from origin in most cases.  Matlab's algorithm seems to
%choose an intelligent descent direction and trust region even in the
%neighborhood of the origin.
[N,fval,flag,out,lam] = fmincon( @obj_fn, N0,...
                                A, b, Aeq, beq,...
                                LB, UB, nlc, optns );

%Next, compute v and a
v = H*u;                    %dx/dt
a = M \ (Fnc - sqg_W'*N);   %du/dt

%used for plotting net friction force
f = - sqg_Wf' * N;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [f,gf] = obj_fn( y )

%symmetric, positive semi-definite Hessian, S = W*Minv*W'
global S;

%quadratic term
ySy = y'*S*y;
norm_y = norm(y);   %Euclidean norm
```

```
%penalty term forces Matlab %to home in on the min norm solution
eps = 1e-6;
eny = eps*norm_y;

%return:
f = 0.5*ySy + eny;    %nonsmooth only at y=0

%analytic gradient
if nargout>1
    if norm_y ~= 0
        gf = S*y + eps*y*(1/norm_y);
    else
        %More accurately, 0 is an element of the subgradient
        gf = 0;
    end
end

%EOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

ProjectConstraint.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Copyright 2005-2008, Miguel J. Gomez
%
% GGL-inspired projection method:
%
% First, solve the non-linear system
%
% I*(x-x1) + (Gx(x1))^T*eta = 0
% G(x) = 0
%
% for z = [x,eta] by the Newton iteration
%
% x + Gx(x1)^T*eta = x1
% Gx(x1)*x = Gx(x1)*x1 - G(x1)
%
% REFERENCES:
% "Solving Ordinary Differential Equations II:  Stiff and Differential
% Algebraic Systems", Hairer, Wanner, 1998
%
function [x1,u1] = ProjectConstraint( eval_H, eval_G, eval_Gx, x, u )

%lengths of generalized state and velocity vectors
nx = length( x );
n = length( u );

%NOTE:  "sqg_" stands for "squiggle", the
%unreduced matrices with a '~' on top.

%initial constraint error (max value)
sqg_G   = eval_G( x );
E   = norm( sqg_G, inf );

%number of ineq cstrts, m + s
ms = length( sqg_G );

m = ms;
```

```
%reduced matrices (no "squiggles")
G = zeros( m, 1 );
Gx = zeros( m, nx );

%m x m zero matrix
O = zeros( m );

%initial estimates
x1  = x;
eta1 = zeros( m,1 );

%iteration count
ctr = 0;

%HACK:  Hardcode the error threshold for the Newton iteration
while E > 1e-10,

    %a single Newton iteration estimates z for F(z)=0
    %by solving F_z(z0)*z1 = F_z(z0)*z0 - F(z0) for z1

    %constraint Jacobian matrix
    sqg_Gx = eval_Gx( x1 );

    G = sqg_G;
    Gx = sqg_Gx;

    %solve linear system
    A   = [ eye(nx), Gx' ; Gx, O ];
    b   = [ x1 ; Gx*x1 - G ];
    %Moore-Penrose generalized inverse
    z = pinv(A) * b;
    x1  = z( 1 : nx );

    %constraint error
    sqg_G   = eval_G( x1 );
    E   = norm( G, inf );

    %HACK:  to prevent infinite loops,
    %do at most 10 iterations
    ctr = ctr + 1;
    if ctr>10
        break;
    end%if
end


% Next, make u satisfy
%
% Gx*H*u = 0
%
% by solving the linear system
%
% I*(u-u0) + (Gx*H)^T*lam = 0
% Gx*H*u                  = 0
%
% for z = [u,lam].  This removes any component of u
% in the subspace spanned by the rows of Gx*H.

%eval cstrt Jacobian matrix at x1
sqg_Gx = eval_Gx( x1 );

Gx = sqg_Gx;
```

```
%velocity transfer matrix
H   = eval_H( x1 );

%matrix product
GxH = Gx*H;

A   = [ eye(n), GxH' ; GxH, 0 ];
b   = [ u ; zeros(m,1) ];
z = pinv(A) * b;

u1   = z( 1 : n );
%lam1 = z( n+1 : n+m );

%EOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Pend_Fnc.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Copyright 2005-2008, Miguel J. Gomez
%
% Compute the generalized non-constraint force Fnc for the simple harmonic
% oscillator.
%
function [Fnc] = Pend_Fnc( mJ, x, u )

%global variables
global g;   %gravitational acceleration

%inertial properties
mass = mJ(1);

%return:
%generalized non-constraint force
Fnc = [ 0, -mass*g, 0 ]';

%dimensionless non-constraint force
Fnc = [ 0, -1, 0 ]';

%EOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Pend_W.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Copyright 2005-2008, Miguel J. Gomez
%
% Compute the (unreduced) wrench matrix sqg_W for the 2D pendulum.
%
function [W,Wf] = Pend_W( x, u )

%frictional terms
global mu;
global b;
global beta;    %rp/l
```

```
global rp;       %rad of pivot
global K;        %angular frequency of undamped SHO

B = beta;
rx = x(1);
ry = x(2);
theta = x(3);
dth_dt = u(3);          %d(theta)/dt
c = cos( theta );
s = sin( theta );
v = rp*K*abs(dth_dt);   %dimensional contact speed
sgn = sign( dth_dt );   %sign( d(theta)/dt )

%Coulomb + (Linear Drag)
f = (mu + b*v)*sgn;

r2 = sqrt(2);
r3 = sqrt(3);
r2_2 = r2 / 2;
r2_B = r2 * B;
r3_2 = r3 / 2;

%number of contact points
global m_s;

%angle increment for m+s contacts
delta = (2*pi)/m_s;

%start with empty matrix
Wf = [];
W = [];

%construct matrix row by row
for i=1:m_s
    sd = sin( (i-1)*delta );
    cd = cos( (i-1)*delta );
    win =   [  c*cd - s*sd, s*cd + c*sd, -sd ];
    wif = f*[ -s*cd - c*sd, c*cd - s*sd, B - cd ];
    wi = win + wif;
    Wf = [Wf ; wif];
    W = [W ; wi];
end

%EOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Pend_G.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Copyright 2005 - 2008, Miguel J. Gomez
%
% Contact constraints for pendulum.
%
function [G] = Pend_G( x )

%dist from contact to CM
%global l;

%the ratio rp/l
```

```
global beta;

B = beta;

%state variables
rx = x(1);
ry = x(2);
theta = x(3);

c = cos( theta );
s = sin( theta );
r2 = sqrt(2);
r3 = sqrt(3);
r2_2 = r2 / 2;
rxcB = rx*c*B;
rxsB = rx*s*B;
rysB = ry*s*B;
rycB = ry*c*B;
rys_rxc = -ry*s - rx*c;
rxcB_rysB_B = rxcB + rysB - B;
rycB_rxsB = rycB - rxsB;
rx2_ry2_1 = rx^2 + ry^2 + 1;

%number of contact points
global m_s;

%angle increment for m+s contacts
delta = (2*pi)/m_s;

%start with empty matrix
G = [];

%construct matrix row by row
for i=1:m_s
    sd = sin( (i-1)*delta );
    cd = cos( (i-1)*delta );
    gi = 2*(rys_rxc + rxcB_rysB_B*cd + rycB_rxsB*sd) + rx2_ry2_1;
    G = [G ; gi];
end

%EOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Pend_Gx.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Copyright 2005 - 2008, Miguel J. Gomez
%
% Constraint Jacobian matrix for the 2D pendulum.
%
function [G_x] = Pend_Gx( x )

%the ratio rp/l
global beta;

B = beta;

%x = [rx, ry, theta]
rx = x(1);
```

```
ry = x(2);
theta = x(3);

c = cos( theta );
s = sin( theta );
rx_c = rx - c;
ry_s = ry - s;
cB = c*B;
sB = s*B;
rxsB = rx*sB;
rxcB = rx*cB;
rysB = ry*sB;
rycB = ry*cB;
rxs_ryc = rx*s - ry*c;
rycB_rxsB = rycB - rxsB;
rxcB_rysB = rxcB + rysB;

%number of contact points
global m_s;

%angle increment for m+s contacts
delta = (2*pi)/m_s;

%start with empty matrix
G_x = [];

%construct matrix row by row
for i=1:m_s
    sd = sin( (i-1)*delta );
    cd = cos( (i-1)*delta );
    gix = 2*[ rx_c + cB*cd - sB*sd,...
              ry_s + sB*cd + cB*sd,...
              rxs_ryc + rycB_rxsB*cd - rxcB_rysB*sd ];
    G_x = [G_x ; gix];
end

%EOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Pend_dGxH_dt.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Copyright 2005-2008, Miguel J. Gomez
%
% Compute d(GxH)/dt for the nonlinear pendulum.
%
function [dGxH_dt] = Pend_dGxH_dt( x, u )

%angular frequency
global K;

%the ratio rp/l
global beta;

B = beta;

%rx, ry, and theta
rx = x(1);
ry = x(2);
```

```
theta = x(3);

%drx/dt, dry/dt, and d(theta)/dt
drx = u(1);
dry = u(2);
dth = u(3);

c = cos( theta );
s = sin( theta );
sdth = s*dth;
cdth = c*dth;
sdthB = sdth*B;
cdthB = cdth*B;
drx_sdth = drx + sdth;
dry_cdth = dry - cdth;
drxs_rxcdth_dryc_rysdth = drx*s + rx*c*dth - dry*c + ry*s*dth;
drxcB_rxsdthB_drysB_rycdthB = (-drx*c + rx*s*dth - dry*s - ry*c*dth)*B;
drxsB_rxcdthB_drycB_rysdthB = (-drx*s - rx*c*dth + dry*c - ry*s*dth)*B;

%number of contact points
global m_s;

%angle increment for m+s contacts
delta = (2*pi)/m_s;

%start with empty matrix
dGxH_dt = [];

%construct matrix row by row
for i=1:m_s
    sd = sin( (i-1)*delta );
    cd = cos( (i-1)*delta );
    ddtgix = 2*[ drx_sdth - sdthB*cd - cdthB*sd,...
                 dry_cdth + cdthB*cd - sdthB*sd,...
                 drxs_rxcdth_dryc_rysdth...
                 + drxcB_rxsdthB_drysB_rycdthB*sd...
                 + drxsB_rxcdthB_drycB_rysdthB*cd ];
    dGxH_dt = [dGxH_dt ; ddtgix];
end

%EOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

SingleBody2D_H.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Copyright 2005-2008, Miguel J. Gomez
%
% Velocity transfer matrix H for a single body in 2 dimensions.
%
function [H] = SingleBody2D_H( x )

%return:
H = eye(3);

%EOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

MassMatrix2D.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Copyright 2005-2008, Miguel J. Gomez
%
% Evaluate the generalized mass matrix M(x) for a single rigid-body in 2
% dimensions.
%
function [M] = MassMatrix2D( mJ, x )

%inertial properties
mass = mJ(1);
J = mJ(2);

%return:
M  = diag( [ mass, mass, J ] );

%EOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

MassMatrix2D_Inv.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Copyright 2005-2008, Miguel J. Gomez
%
% Compute the inverse of the generalized mass matrix M(x) for a single
% rigid-body in 2 dimensions.
%
function [Minv] = MassMatrix2D_Inv( mJ, x )

%mass inverse
m_inv = 1/mJ(1);

%moment of inertia inverse
J_inv = 1/mJ(2);

%return:
Minv  = diag( [m_inv, m_inv, J_inv] );

%EOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

VITA

Miguel J. Gomez was born in San Francisco, California, on September 3, 1972. In 1995, he completed a B.S. in Physics at the University of Washington in Seattle. While working as a software engineer, he reentered the University of Washington in 2000 and completed his Ph.D. in Applied Mathematics in 2008. He currently resides in Redmond, Washington, with his wife and two children.