USING SIMULATION FOR PLANNING AND DESIGN OF ROBOTIC SYSTEMS WITH INTERMITTENT CONTACT

By

Stephen George Berard

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: COMPUTER SCIENCE

Approved by the
Examining Committee:

Jeffrey Trinkle, Thesis Adviser

Kurt Anderson, Member

John Mitchell, Member

Barbara Cutler, Member

Rensselaer Polytechnic Institute
Troy, New York

April 2009
(For Graduation May 2009)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

I am grateful to my advisor, Jeff Trinkle. Without his guidance this thesis would not have been possible. I am grateful to the department's staff for making the logistical aspects of this process as easy as possible. A special thanks to Shannon for putting up with my idle bantering. I would also like to thank my fellow lab mates and collaborators that I've had the pleasure of working with over the years. I especially thank Eric for being a great friend during these past six years, and wish him the best of luck in his future endeavors.

On a personal level, I'd like to thank my parents, Jane and Donald, for their unwavering support and encouragement. Thank you, Mom and Dad. In addition, I'd like to thank my brother for his support and for keeping my brain sharp with all those consulting calls. Thank you Mark. Most of all, I'd like to thank Katie for her enduring motivation and emotional support during this long process. Thank you Blondie!

ABSTRACT

For robotic systems to automatically plan and execute manipulation tasks involving intermittent contact, one must be able to accurately predict the motions of the manipulated objects. Not surprisingly, many of the important manipulation problems that could yield to closed-form analysis have been solved and studied thoroughly. Problems characterized by intermittent contact are one particularly important type of robotics problem for which research must rely on simulation techniques.

Due to the intermittency of contact and the presence of stick-slip frictional behavior, dynamic models of such multibody systems are inherently (mathematically) nonsmooth, and are thus difficult to integrate accurately or quickly. Commercially available software tools have a difficult time simulating systems with unilateral constraints, that is constraints where touching bodies are allowed to touch or separate, but not interpenetrate. Users expect to spend considerable effort in a trial-and-error search for good simulation parameters to obtain believable, not necessarily accurate, results. Even the seemingly simple problem of a sphere rolling on a horizontal plane under only the influence of gravity is challenging for commercial simulators. The correct handling of unilateral contact constraints is one of the most difficult challenges left for many commercial simulation software packages.

This thesis relates to the use of simulation for planning and design of robotics systems with intermittent contact. As previously noted, such systems arise in many applications, including automated manufacturing, health care, and personal robotics. The relationship between these seemingly unrelated application areas is forged by the desire of interactive robotic devices situated in an unstructured world. A better understanding of the dynamics, especially the contact dynamics, will allow us to improve the autonomy of these systems. In the first part of this work, we introduce four new time-stepping methods, which were constructed for a variety of reasons, including accuracy, performance, and design. Next, we developed a simulation software package (dubbed daVinci Code) that implements these new methods. This software tool facilitates the simulation, analysis, and virtual design of multibody systems with intermittent frictional unilateral contact. Next, a study on the applicability of our time-stepping method is presented. We performed a numerical study on the accuracy of our methods, and experimentally validated our time-stepper on a system composed of a vibrating rigid plate and interacting part. With the accuracy of our time-stepper verified for this system, we were able to study the inverse problem of designing new plate motions to generate a desired part motion. Lastly, we present our initial results of a new non-recursive nonlinear filter using our model of these systems. This filter allows us to estimate the system's parameters, which is a necessary requirement for using simulation for planning and design. The filtering problem is particularly challenging, since the underlying mathematical model is nonsmooth.

1. Introduction

For robotic systems to automatically plan and execute manipulation tasks involving intermittent contact, one must be able to accurately predict the motions of the manipulated objects. Not surprisingly, many of the important manipulation problems that could yield to closed-form analysis have been solved and studied thoroughly. Problems characterized by intermittent contact are one particularly important type of robotics problem for which research must rely on simulation techniques [24, 18, 99, 111, 55, 90]. Examples of robotics applications with contact for which special-purpose simulations have been developed include the design of local controllers for self-organizing systems [61], the synthesis and execution of grasping and manipulation [26, 75, 114, 124], and parts feeding and assembly by vibrating plates [96, 103, 119, 49].

Due to the intermittency of contact and the presence of stick-slip frictional behavior, dynamic models of such multibody systems are inherently (mathematically) nonsmooth, and are thus difficult to integrate accurately or quickly. Commercially available software systems have a difficult time simulating any system with unilateral constraints, that is constraints where touching bodies are allowed to touch or separate, but not interpenetrate. Users expect to spend considerable effort in a trial-and-error search for good simulation parameters to obtain believable, not necessarily accurate, results. Even the seemingly simple problem of a sphere rolling on a horizontal plane under only the influence of gravity is challenging for commercial simulators [24]. The correct handling of unilateral contact constraints is one of the most difficult challenges left for many commercial simulation software packages.

Further evidence of the need for simulation can be found in the recent trend of robotics researchers (those studying grasping, assembly, and dexterous manipulation problems) using Open Dynamic Engine (ODE) [98]. ODE was developed for computer game applications, which has led its developers to trade physical accuracy for simulation speed while trying to achieve believability. Researchers that we have communicated with on the topic of simulation cite ODEs poor accuracy and ask if we know of better simulation tools [77].

For an example of why we need simulation tools with improved accuracy, we repeated, in simulation, an earlier grasping experiment performed by Brost and Christensen [20] at Sandia National Labs. The authors were interested in developing manipulation strategies that reliably accomplished desired goals, in the presence of uncertainty. The particular task they chose was to grasp a circular lock piece with a parallel-jaw gripper. The goal of their experiment was to estimate the probability of a successful grasp as a function of the starting position of the grippers. In pursuit of their motivation, there was uncertainty in the gripper closing speed and the coefficient of friction between the surface and part. They used two gripper closing speeds, normal and fast (twice normal) and the surface was either clean or purposefully contaminated with sand, affecting the coefficient of surface friction.

The idea behind their experiment was straight forward. They defined a box of initial positions for the gripper, and executed a grasp for each point in the box (sampled at 1-mm intervals). For each point in the box, the grasp was executed 30 times, and based on the result of the trial a success or failure was recorded. They then estimated the success probability for each point as $N/30$, where N was the number of observed successes for the point. It took them over 100,000 trials to collect all the data. The cost was high, since expensive equipment and a technician was dedicated to these experiments for more than a week. Performing this experiment in simulation has the advantage of being faster and cheaper, but is useless if the simulation is not accurate enough. In addition, simulation also opens the possibility of designing grasp strategies while simultaneously computing the success probabilities.

1.1 Contributions

My research contributions relate to the use of simulation for planning and design of robotics systems with intermittent contact. As previously noted, such systems arise in many applications, including automated manufacturing, health care, and personal robotics. The relationship between these seemingly unrelated application areas is forged by the desire of interactive robotic devices situated in an unstructured world. A better understanding of the dynamics, especially the contact dynamics, will allow us to improve the autonomy of these systems. The research contributions of this thesis are:

- Four new time-stepping methods
    - Study on accuracy
    - Experimental validation
- Software package (dVC) that implements these methods
- Parameter identification for these nonsmooth nonlinear systems

In Chapter 3, four new time-stepping formulations are presented (see Figure 1.1 for a graphical representation). These formulations were constructed for a variety of reasons, including accuracy, performance, and design. The first method correctly handles non-penetration constraints for non-convex corners (the problem occurring in figure 4.8) [36]. The next formulation is a method that adds surface friction to a 2D body allowing for an approximation of surface friction effects without the associated costs of modeling the entire 3D body [18]. Next is a time-stepping formulation for multi-body quasi-static systems [112]. Despite being more restrictive than the dynamic model, it includes a large number of important tasks such as low-speed assembly, static grasping, and walking using tripods of support. Lastly, a fully implicit time-stepping scheme is presented [24, 23], which we believe is currently the most accurate time-stepping formulation available. This formulation won best student paper at the Robotics: Science and Systems conference.

In Chapter 4, we present *daVinci Code* (dVC), which is a new software tool we designed and implemented to facilitate simulation, analysis, and virtual design of multibody systems with intermittent frictional unilateral contact [18]. dVC uses the state-of-the-art time-stepping methods we developed to capture the nonsmooth phenomena (stick-slip transitions and contact loss and formation) without regularization. These methods are numerically stable and provably convergent.

In Chapter 5, we present the results of a study on the applicability of our time-stepping method to a system composed of a rigid plate and a single rigid

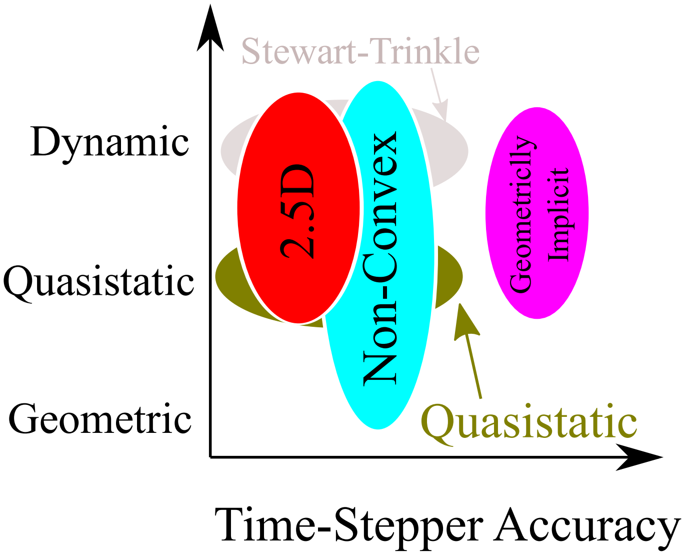


Figure 1.1: The four new time-stepping formulations are displayed along with the existing Stewart-Trinkle time-stepper. The y axis is the model space, and the x axis is the time-stepper accuracy. The new methods are not mutually exclusive, for example it is possible to combine the 2.5D, non-convex, and quasi-static steppers together.

body [17]. The plate followed a prescribed periodic motion of small amplitude and high frequency, such that the net force applied to the part appeared to be from a time-independent, position-dependent velocity field in the plane of the plate. Theoretical results obtained by Vose *et al.* were found to be in good agreement with simulation results obtained with our time-stepping method. In addition, simulations were found to agree with the qualitative experimental results of Vose *et al.* After such verification of the simulation method, additional numerical studies were done that would have been impossible to carry out analytically. For example, we were able to demonstrate the convergence of the method with decreasing step size (as predicted theoretically by Stewart [109]).

In Chapter 6, we again looked at the system composed of a vibrating rigid plate and interacting part. With the accuracy of our time-stepper verified for this system, we were able to study the inverse problem of designing new plate motions to generate a desired part motion. This is done through an optimization framework, where a simulation of the part interacting with the plate (including the full dynamics of the system) is performed, and based on the results of the simulation the motion of the plate is modified. Vose kindly ran our learned plate motion on his experimental device, and without any tuning (of the simulation parameters or device parameters) verified our learned (by simulation) plate motion produced the desired part motion on the actual device.

In Chapter 7, we present our initial results of a new nonrecursive nonlinear filter for systems with complementarity [29] constraints. We formulate the estimation problem as a mathematical program with equilibrium constraints (MPEC) [70] and present some initial results for the dual estimation problem on simulated and experimental data.

2. Background

Rigid multi-body dynamics with intermittent frictional contact is one of the most challenging physical events to simulate. It is an active area of research in robotics, gaming, and 3D graphics. This background chapter begins with a review of rigid body motion. Next, it incorporates forces into the study of motion, leading to the famous Newton-Euler equations of motion. Afterwards, a discussion on the complementarity problem, and how one solves this problem is provided. Next, traditional integrate-detect-restart methods for solving the constrained equations of motion is shown. Lastly, the complementarity based time-stepping formulation of the dynamics is presented.

2.1 Rigid Body Kinematics

Kinematics describes the motion of a body (more specifically, the motion of a frame attached to a point on the body) solely as a function of time. The response of a body subject to forces (dynamics) will be discussed in section 2.4. A rigid body is an idealization of a physical object. Without providing the details, this common modeling decision simplifies the mathematical description of the object's motion by allowing us to study the motion of a "reference frame" attached to the body. Given two points in a rigid body, their relative distance is fixed for all time. Enforcing this modeling decision requires that rigid bodies cannot be inter-penetrated or deformed (though, current research does allow for local deformations).

2.1.1 Position and Orientation

An unconstrained rigid body has six degrees of freedom: three translational and three rotational. The configuration (*i.e.* position and orientation) of the body can be represented by an orthonormal reference frame attached to a point on the body (see figure 2.1). For convenience (when we get to dynamics), this frame is usually attached to the center of mass of the body, but this is not a requirement. By convention, we also assume that the frame is right-handed.

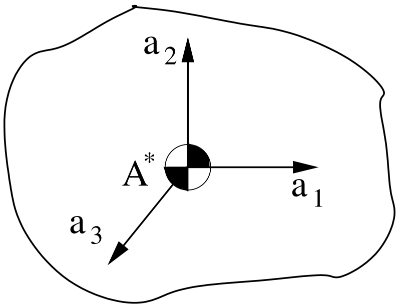


Figure 2.1: Position and orientation of a rigid body specified by a body fixed reference frame $\{A\}$ with origin at the center of mass of body A, denoted A^*.

A frame $\{A\}$ is orthonormal and right-handed if its 3 basis vectors, $\mathbf{a}_1,\ \mathbf{a}_2,\ \mathbf{a}_3$ satisfy:

1. $||\mathbf{a}_i|| = 1,\ i = 1, 2, 3$

2. $\mathbf{a}_i \cdot \mathbf{a}_j = 0,\ i, j = 1, 2, 3, i \neq j$

3. $\mathbf{a}_1 \times \mathbf{a}_2 = \mathbf{a}_3$

The first condition is the normality constraint, the second condition is the orthogonality constraint, and the final condition is the right-hand rule.

Let ${}^{A}\mathbf{R} = [\mathbf{a}_1\ \mathbf{a}_2\ \mathbf{a}_3]$ be a 3×3 matrix. This matrix can be considered an orthonormal basis for $\mathbb{E}^3$, where $\mathbb{E}^3$ is the 3-dimensional Euclidean space. This allows one to uniquely represent a vector $\mathbf{v} \in \mathbb{E}^3$ by a vector in $\mathbb{R}^3$:

$$\mathbf{v} = {}^{A}\mathbf{R}^{T\,A}\mathbf{v} \tag{2.1}$$

where ${}^{A}\mathbf{v} \in \mathbb{R}^3$ is the representation of $\mathbf{v}$ in the coordinate frame $\{A\}$.

A vector $\mathbf{v}$ can be represented in various frames:

$$\mathbf{v} = {}^{A}\mathbf{R}^{T\,A}\mathbf{v} = {}^{B}\mathbf{R}^{T\,B}\mathbf{v} \tag{2.2}$$

where $\{B\}$ is an orthonormal frame and ${}^{B}\mathbf{v}$ is the coordinates of $\mathbf{v}$ in frame $\{B\}$.

The two representations are related by (since ${}^{A}\mathbf{R}$ is orthogonal, ${}^{A}\mathbf{R}{}^{A}\mathbf{R}^{T} = \mathbf{I}$):

$$ {}^{A}\mathbf{v} = {}^{A}\mathbf{R}{}^{B}\mathbf{R}^{T}{}^{B}\mathbf{v} \tag{2.3} $$

This 3×3 matrix ${}^{A}\mathbf{R}{}^{B}\mathbf{R}^{T}$ that converts the representations between frames is often called a rotation matrix (or direction cosine matrix), and is denoted as ${}^{A}_{B}\mathbf{R}$.

The position of the rigid body is described by the position of the body fixed frame with respect to a fixed (also known as inertial or Newtonian) world frame. The orientation of the rigid body can represented by the rotation matrix associated with the body fixed frame and the fixed world frame. There are, however, many possible ways to parameterize the rotation, and the next section briefly discusses some of them.

2.1.2 Rotation Group

One interpretation of the rotation matrix, and the reason for its name, is that it rotates frames:

$$ {}^{A}\mathbf{R} = {}^{A}_{B}\mathbf{R}{}^{B}\mathbf{R} \tag{2.4} $$

This leads directly to the definition that $\mathbf{R}$ is a rotation if for any orthonormal frame $\{A\}$, $\mathbf{R}{}^{A}\mathbf{R}$ is also an orthonormal frame. All rotation matrices (orthonormal matrices with positive determinant) form a group called the *special orthogonal group* of dimension 3, denoted $SO(3)$:

$$ SO(3) = \left\{ \mathbf{R} \mid \mathbf{R} \in \mathbb{R}^{3\times3},\ \mathbf{R}^{T}\mathbf{R} = \mathbf{R}\mathbf{R}^{T} = \mathbf{I},\ \det(\mathbf{R}) > 0 \right\} \tag{2.5} $$

The condition $\det(\mathbf{R}) > 0$ enforces the right-hand rule constraint, and is the reason for the name *special* in special orthogonal group.

However, there are other possible representations of $SO(3)$ in addition to rotation matrices [72, 123], for example Euler angles, Gibbs-Rodrigues' representation, Euler parameters etc. We recommend using Euler parameters (unit quaternions) for parameterizing orientation since they are compact (4 elements) and do not suffer from singularities (singularities are discussed in section 2.1.4.1). Euler proved that general motion of a rigid body with a fixed point is a rotation about a fixed axis.

This allows one to use a single rotation about a fixed axis instead of decomposing the motion into three simple rotations as is commonly done (Euler angles). The drawback of this representation is that instead of requiring three parameters, one for each simple rotation angle, it requires 4 parameters, one for the angle of rotation and three for the axis of rotation. Let the vector $\hat{\mathbf{b}}$ denote a unit vector parallel to the axis of rotation and θ the angle of rotation about this fixed axis. We can now define the 4 parameters in terms of $\hat{\mathbf{b}}$ and θ as

$$\begin{aligned} \mathbf{e}_v &= \hat{\mathbf{b}} \sin\left(\frac{\theta}{2}\right) \\ e_w &= \cos\left(\frac{\theta}{2}\right) \end{aligned}$$

where $\mathbf{e}_v = [e_x \; e_y \; e_z]^T$ is the vector component of the quaternion and e_w is the scalar component. Taken together, $(e_w, \mathbf{e}_v)$ is a four parameter representation of $SO(3)$ with one constraint, $e_w^2 + ||\mathbf{e}_v||^2 = 1$.

If the Euler parameters are a different representation of $SO(3)$, it should be possible to transform them to a corresponding rotation matrix. This is indeed possible, and given $(e_w, \mathbf{e}_v)$, the corresponding rotation matrix $\mathbf{R}$ is [123]:

$$\begin{bmatrix} 1 - 2e_y^2 - 2e_z^2 & 2e_xe_y - 2e_we_z & 2e_xe_z + 2e_we_y \\ 2e_xe_y + 2e_we_z & 1 - 2e_x^2 - 2e_z^2 & 2e_ye_z - 2e_we_x \\ 2e_xe_z - 2e_we_y & 2e_ye_z + 2e_we_x & 1 - 2e_x^2 - 2e_y^2 \end{bmatrix} \tag{2.6}$$

2.1.3 Euclidean Group

The group of rotations, $SO(3)$, can be extended to allow both rotations and translations. For a body A, let $\{A\}$ be the orthonormal frame attached to the body representing its orientation and let $\mathbf{r}^{O_A}$ be a vector from a reference point to the origin of the reference frame $\{A\}$ (see figure 2.2). Taken together, $(\mathbf{r}^{O_A},\ {}^{A}\mathbf{R})$ represent the position and orientation of body A by the following homogeneous transformation matrix:

$${}^{A}\mathbf{T} = \begin{bmatrix} {}^{A}\mathbf{R} & \mathbf{r}^{O_A} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.7}$$

The set of all such matrices that represent rigid body displacements is called the *special Euclidean group* of order 3, denoted $SE(3)$:

$$SE(3) = \left\{ \mathbf{T} \mid \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix},\ \mathbf{R} \in SO(3),\ \mathbf{r} \in \mathbb{R}^3 \right\} \tag{2.8}$$

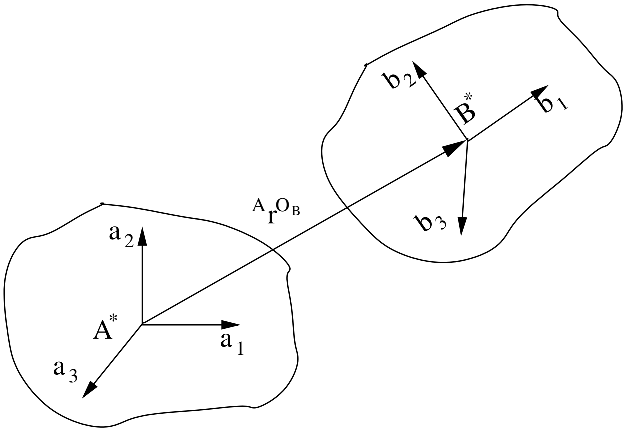


Figure 2.2: Displacement of body A to body B.

We can construct a transformation matrix representing the displacement from body A to body B (given in A's frame) as:

$$ {}^{A}_{B}\mathbf{T} = \begin{bmatrix} {}^{A}_{B}\mathbf{R} & {}^{A}\mathbf{r}^{O_B} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \tag{2.9}$$

where ${}^{A}\mathbf{r}^{O_B} = {}^{A}_{F}\mathbf{R}(O_B - O_A)$ and ${}^{A}_{B}\mathbf{R}$ is a rotation matrix that transforms from $\{B\}$ to $\{A\}$.

The composition of two displacements, first from $\{A\}$ to $\{B\}$, and second from

$\{B\}$ to $\{C\}$ can be achieved by matrix multiplication of ${}^A_B\mathbf{T}$ and ${}^B_C\mathbf{T}$:

$$\begin{aligned}
{}^A_C\mathbf{T} &= \begin{bmatrix} {}^A_C\mathbf{R} & {}^A\mathbf{r}^{O_C} \\ \mathbf{0} & 1 \end{bmatrix} \\
&= \begin{bmatrix} {}^A_B\mathbf{R} & {}^A\mathbf{r}^{O_B} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^B_C\mathbf{R} & {}^B\mathbf{r}^{O_C} \\ \mathbf{0} & 1 \end{bmatrix} \\
&= \begin{bmatrix} {}^A_B\mathbf{R}{}^B_C\mathbf{R} & {}^A_B\mathbf{R}{}^B\mathbf{r}^{O_C} + {}^A\mathbf{r}^{O_B} \\ \mathbf{0} & 1 \end{bmatrix}
\end{aligned} \tag{2.10}$$

The group $SO(3)$ is a subgroup of $SE(3)$, where the displacement vector $\mathbf{r} = 0$. It can be thought of as the group of all spherical displacements:

$${}^A_B\mathbf{T} = \begin{bmatrix} \mathbf{R} & 0 \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.11}$$

2.1.4 Velocity of a Rigid Body

The first derivative of the body's motion gives an expression for the velocity. Considering the two components, position and orientation, of the body's configuration separately, we can define two velocity components: *linear* velocity $\mathbf{v} = [v_x \ v_y \ v_z]^T$ and *angular* velocity $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$. The linear velocity is the time derivative of the translational position vector

$$\mathbf{v} = \dot{\mathbf{r}} \tag{2.12}$$

where the $\dot{()}$ notation is shorthand for the derivative with respect to time, *e.g.* $\dot{\mathbf{r}} = \dfrac{d\mathbf{r}}{dt}$. The vector $\mathbf{v}$ is the velocity at the origin of the moving frame attached to the body, which may or may not be the body's center of mass.

Similar to linear velocity, angular velocity is the time rate of change of the body's orientation. However, unlike linear velocity there is no notion of an orientation vector, so the definition of angular velocity is not the time derivative of a vector. The angular velocity of a reference frame $\{B\}$ in a reference frame $\{A\}$, denoted ${}^A\boldsymbol{\omega}^B$, is defined[1] as [58]

$$ {}^A\boldsymbol{\omega}^B \triangleq \mathbf{b_1}\frac{{}^A d\mathbf{b_2}}{dt}\cdot\mathbf{b}_3 + \mathbf{b_2}\frac{{}^A d\mathbf{b_3}}{dt}\cdot\mathbf{b}_1 + \mathbf{b_3}\frac{{}^A d\mathbf{b_1}}{dt}\cdot\mathbf{b}_2 \tag{2.13}$$

Using this definition of angular velocity, we can obtain the time derivative of any vector fixed in the body. Let ${}^B\mathbf{v} = [v_x\ v_y\ v_z]^T$ be any vector fixed in body B's frame, then

$$\frac{{}^A d\mathbf{v}}{dt} = {}^A\boldsymbol{\omega}^B \times {}^B\mathbf{v} = \begin{bmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{2.14}$$

For a physical interpretation of the angular velocity, the magnitude of $\boldsymbol{\omega}$ is the rate the body is rotating, and the direction of $\boldsymbol{\omega}$ is the axis of rotation.

2.1.4.1 Kinematic Update

Given an angular velocity of a rigid body, the orientation is determined by integrating the differential equations for one of the many possible parameterizations of orientation (described in section 2.1.2). In general, if $\mathbf{q} \in \mathbb{R}^k$ is a representation of $SO(3)$, the goal is to find the representation Jacobian $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{k\times 3}$ such that

$$\frac{d\mathbf{q}}{dt} = \mathbf{G}(\mathbf{q})\boldsymbol{\omega} \tag{2.15}$$

If $\mathbf{G}$ loses rank (called a singularity of the representation), then $\mathbf{q}$ cannot be solved for. These singularities are purely mathematical and a function of the specific representation, not a function of a physical process.

We recommend using Euler parameters for parameterizing $SO(3)$, and as such now present the differential kinematics for them. For Euler parameters, $\mathbf{q} =$

[1] This is not the frequently given definition of angular velocity as the limit of $\frac{\delta\theta}{\delta t}$ as δt goes to 0. However, this definition makes the relationship between the time-derivative of a vector and the cross-product multiplication easier to show.

$[e_w\ e_x\ e_y\ e_z]$, the representation Jacobian is given by [27, 104]:

$$\mathbf{G}(\mathbf{q}) = \frac{1}{2}\begin{bmatrix} -e_x & -e_y & -e_z \\ e_w & e_z & -e_y \\ -e_z & e_w & e_x \\ e_y & -e_x & e_w \end{bmatrix} \tag{2.16}$$

This Jacobian can never become singular, which is why we prefer the use of Euler parameters for representing orientation.

2.1.4.2 Velocity of a Point on a Rigid Body

Given a point p on a rigid body B, the position of p in the body fixed frame $\{B\}$, denoted ${}^B\mathbf{r}^p$, is constant. However, the position of p in another frame $\{A\}$, denoted ${}^A\mathbf{r}^p(t)$, is a function of time since the position of body B in A's frame, ${}^A\mathbf{r}^{O_B}(t)$, is a function of time (see figure 2.3).

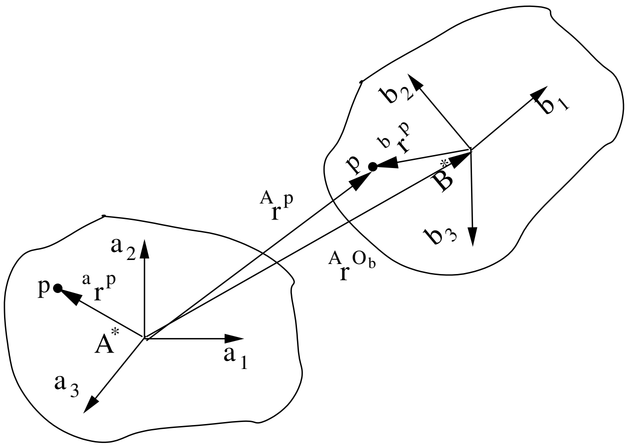


Figure 2.3: Velocity of a point p on a rigid body.

Using the Euclidean frame and homogeneous coordinates for rigid body displacement, the position of point p in frame A is:

$$
\begin{aligned}
{}^{A}\mathbf{r}^{p} &= {}^{A}_{B}\mathbf{T}\begin{bmatrix} {}^{B}\mathbf{r}^{p} \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} {}^{A}_{B}\mathbf{R} & {}^{A}\mathbf{r}^{O_B} \\ \mathbf{0} & 1 \end{bmatrix}\begin{bmatrix} {}^{B}\mathbf{r}^{p} \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} {}^{A}_{B}\mathbf{R}\,{}^{B}\mathbf{r}^{p} + {}^{A}\mathbf{r}^{O_B} \\ 1 \end{bmatrix}
\end{aligned}
\tag{2.17}
$$

Differentiating equation (2.17) with respect to time and using the definition of angular velocity (equation 2.13) we can compute the velocity of the point p in A's frame, ${}^{A}\mathbf{v}^{p}$, as

$$
\begin{aligned}
{}^{A}\mathbf{v}^{p} &= \frac{d}{dt}({}^{A}_{B}\mathbf{R}\,{}^{B}\mathbf{r}^{p}) + \frac{d}{dt}({}^{A}\mathbf{r}^{O_B}) \\
&= {}^{A}\boldsymbol{\omega}^{B} \times {}^{B}\mathbf{r}^{p} +^{A} \mathbf{v}^{B}
\end{aligned}
\tag{2.18}
$$

which can be written more compactly as

$$
{}^{A}\mathbf{v}^{p} =^{A} \mathbf{v}^{B} + {}^{A}\widetilde{\boldsymbol{\omega}}^{B}\,{}^{B}\mathbf{r}^{p} \tag{2.19}
$$

where ${}^{A}\widetilde{\boldsymbol{\omega}}^{B}$ is a skew-symmetric matrix operator with the form:

$$
\begin{bmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{bmatrix}
$$

2.1.4.3 Twists

A twist is a compact representation for expressing the displacement or motion (often called velocity) of a rigid body. A twist, $\boldsymbol{\nu} \in \mathbb{R}^6$, consists of the linear and angular velocity components concatenated[2] into a single vector:

$$\boldsymbol{\nu} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \tag{2.20}$$

where $\mathbf{v}$ is the linear velocity of the frame and $\boldsymbol{\omega}$ is the angular velocity of the frame.

Using homogeneous coordinates and twists allows us to compactly represent the motion of rigid bodies. Looking back at the equation for the velocity of a point (equation (2.19)), we can rewrite it as,

$$^{A}\mathbf{v}^{p} = {}^{A}_{B}\mathbf{H} \begin{bmatrix} ^{B}\mathbf{r}^{p} \\ 1 \end{bmatrix} \tag{2.21}$$

where the ${}^{A}_{B}\mathbf{H} \in \mathbb{R}^{4\times4}$ is a matrix operator defined as:

$${}^{A}_{B}H = \begin{bmatrix} ^{A}\widetilde{\boldsymbol{\omega}}^{B} & ^{A}\mathbf{v}^{B} \\ \mathbf{0}_{1\times3} & 0 \end{bmatrix} \tag{2.22}$$

2.2 Kinetics

Where kinematics describes the motion of a particle, kinetics looks at how the motion is related to the forces acting upon it. This section describes a general representation of forces and then present Newton's second law using this representation.

In general a system of forces and moments acting on a body cannot be reduced to a single resultant force or a single resultant moment. Instead, the forces are added into a single resultant force and the moments are added about the body's center of mass (this is why the frame's origin is usually placed at the center of mass) into a resultant moment. For example, if a body was being acted upon by f forces, $\mathbf{F}_1, \ldots, \mathbf{F}_f$ and m moments $\mathbf{M}_1, \ldots, \mathbf{M}_m$, the resultant force $\mathbf{F}$ and moment $\mathbf{M}$ are

[2] In many algebra texts, a twist is defined with the angular component first, followed by the linear elements. Since classical mechanics traditionally represents position and velocity with the linear terms first, we have swapped the order but will continue to call it a twist.

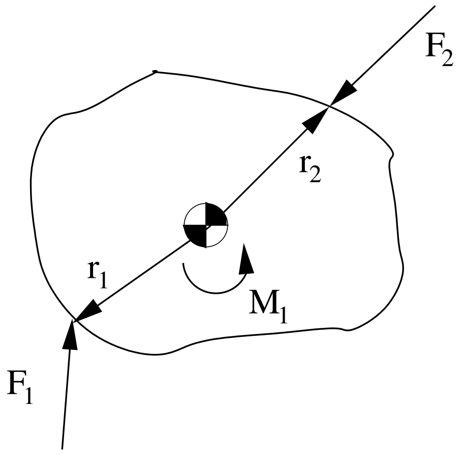


Figure 2.4: Forces and moments acting on a rigid body.

given by:

$$\begin{aligned}\mathbf{F} &= \sum_{i=1}^{f} \mathbf{F}_i \\ \mathbf{M} &= \sum_{i=1}^{m} \mathbf{M}_i + \sum_{i=1}^{f} \mathbf{r}_i \times \mathbf{F}_i\end{aligned} \tag{2.23}$$

where $\mathbf{r}_i$ is a vector from center of mass to the point where the force $\mathbf{F}_i$ is acting (see figure 2.4).

2.2.1 Wrenches

We just showed that a system of external forces and moments can be combined into a single resultant force and resultant moment. If we combine these two resultant values into a single vector $\boldsymbol{\lambda}$, we arrive at the definition of a wrench,

$$\boldsymbol{\lambda} = \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} \tag{2.24}$$

In practice, it is important to remember that the wrenches are written with respect to the basis vectors of a body frame, the force $\mathbf{F}$ is applied to the origin of the body frame, and the moment $\mathbf{M}$ is acting about the origin of the body frame.

2.2.2 Transforming Twists and Wrenches

Similar to how the 4×4 transformation matrix, ${}^{A}_{B}\mathbf{T} = \begin{bmatrix} {}^{A}_{B}\mathbf{R} & {}^{A}\mathbf{r}^{O_B} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}$, transforms vectors between different frames, we can construct a transformation matrix to transform twists and wrenches between different frames [78]. Given twists ${}^{A}\boldsymbol{\nu}$ and ${}^{B}\boldsymbol{\nu}$ represented in frames $\{A\}$ and $\{B\}$ respectively, they are related by:

$$ {}^{A}\boldsymbol{\nu} = {}^{A}_{B}\mathbf{Ad}{}^{B}\boldsymbol{\nu} \tag{2.25} $$

where ${}^{A}_{B}\mathbf{Ad}$ is:

$$ {}^{A}_{B}\mathbf{Ad} = \begin{bmatrix} {}^{A}_{B}\mathbf{R} & {}^{A}_{B}\mathbf{R} \\ 0 & [{}^{A}\widetilde{\mathbf{r}}^{O_B}]{}^{A}_{B}\mathbf{R} \end{bmatrix} \tag{2.26} $$

The inverse of the adjoint transform that transforms a twist from frame $\{A\}$ to frame $\{B\}$ is the adjoint that transforms a twist from frame $\{B\}$ to frame $\{A\}$:

$$ {}^{A}\boldsymbol{\nu} = {}^{A}_{B}\mathbf{Ad}{}^{B}\boldsymbol{\nu} = {}^{B}_{A}\mathbf{Ad}^{-1}{}^{B}\boldsymbol{\nu} \tag{2.27} $$

Wrenches are related by the transpose of the adjoint matrix,

$$ {}^{A}\boldsymbol{\lambda} = {}^{A}_{B}\mathbf{Ad}^{T}{}^{B}\boldsymbol{\lambda} \tag{2.28} $$

2.2.3 Equations of Motion

Describing the equations of motion for a rigid body begins by considering the body to be a collection of particles. The distance between the particles is fixed and the mass is integrated over the collection [72]. In this fashion, the center of mass of the body behaves as if it was a particle, and we can apply the Newton-Euler equations of motion to the center of mass.

To describe the dynamic model mathematically, we first introduce some notation. Let $\mathbf{q}_j$ be the position and orientation of a frame attached to body j's center of mass in an inertial frame and $\boldsymbol{\nu}_j$ be the velocity twist of the origin of the frame.

Newton's 2nd law relates the sum of the forces acting on the body to the acceleration of the body:

$$\mathbf{F}(\mathbf{q}_j, \dot{\mathbf{q}}_j, t) = m(\mathbf{q}_j, t)\ddot{\mathbf{q}}_j \tag{2.29}$$

where $m(\mathbf{q}_j, t)$ the mass of body j, and $\mathbf{F}(\mathbf{q}_j, \dot{\mathbf{q}}_j, t)$ the sum of all forces acting on body j. As explained in section 2.1.4.1, the dimension of $\mathbf{q}_j$ is not necessarily the same as the dimensions of $\dot{\mathbf{q}}_j$ and $\ddot{\mathbf{q}}_j$.

Similar to Newton's law, Euler's equation relates the resultant moment applied to the body to the angular momentum:

$$\boldsymbol{\tau}(\mathbf{q}_j, \dot{\mathbf{q}}_j, t) = \mathcal{I}(\mathbf{q}_j, t)\dot{\boldsymbol{\omega}}_j + \boldsymbol{\omega}_j \times \mathcal{I}(\mathbf{q}_j, t)\boldsymbol{\omega}_j \tag{2.30}$$

where $\boldsymbol{\tau}(\mathbf{q}_j, \dot{\mathbf{q}}_j, t)$ denotes the resultant moment acting on body j and $\mathcal{I}(\mathbf{q}_j, t)$ the inertia tensor for body j. Typically, $\mathcal{I}$ is known in the body fixed frame $\{B\}$, ${}^{B}\mathcal{I}$, and must be converted to the fixed frame $\{F\}$, $\mathcal{I} = {}^{F}_{B}\mathbf{R}^{B}\mathcal{I}^{F}_{B}\mathbf{R}^{T}$.

2.3 Complementarity Problem

As we will show in section 2.4, one natural way to model the dynamics of a system with frictional intermittent contact is as a differential complementarity problem (DCP) [85]. In this section we provide the basic definitions of various forms of complementarity problems, all of which will be seen in later sections.

Definition 1 (Differential Complementarity Problem)**.** Let $g(\mathbf{u}, \mathbf{v}) : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{n_1}$ and $f(\mathbf{u}, \mathbf{v}) : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{n_2}$ be given vector functions of $\mathbf{u} \in \mathbb{R}^{n_1}$ and $\mathbf{v} \in \mathbb{R}^{n_2}$, with $n_1 + n_2 = n$. Find $\mathbf{u}$ and $\mathbf{v}$ satisfying

$$\begin{aligned} \dot{\mathbf{u}} = g(\mathbf{u}, \mathbf{v}), &\qquad \mathbf{u}, \text{free} \\ 0 \leq \mathbf{v} \ \perp \ f(\mathbf{u}, \mathbf{v}) \geq 0 & \end{aligned}$$

Typically, the DCP is not solved directly but instead a time-stepping scheme is employed and the resulting (possible nonlinear) mixed complementarity problem [29, 85] is solved at each time step.

Definition 2 (Mixed Complementarity Problem)**.** Let $g(\mathbf{u}, \mathbf{v}) : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{n_1}$ and $f(\mathbf{u}, \mathbf{v}) : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{n_2}$ be given vector functions of $\mathbf{u} \in \mathbb{R}^{n_1}$ and $\mathbf{v} \in \mathbb{R}^{n_2}$, with $n_1 + n_2 = n$. Find $\mathbf{u}$ and $\mathbf{v}$ satisfying

$$\begin{aligned} 0 &= g(\mathbf{u}, \mathbf{v}), \qquad \mathbf{u}, \text{free} \\ 0 \leq \mathbf{v} \;\perp\; & f(\mathbf{u}, \mathbf{v}) \geq 0 \end{aligned}$$

In the special case that the functions g and f are linear, we arrive at the definition for a mixed linear complementarity problem (MiLCP) [29]:

Definition 3 (Mixed Linear Complementarity Problem)**.** Let $\mathbf{A} \in \mathbb{R}^{n_1 \times n_1}$, $\mathbf{B} \in \mathbb{R}^{n_2 \times n_2}$, $\mathbf{C} \in \mathbb{R}^{n_1 \times n_2}$, and $\mathbf{D} \in \mathbb{R}^{n_2 \times n_1}$ be given matrices. Let $\mathbf{a} \in \mathbb{R}^{n_1}$ and $\mathbf{b} \in \mathbb{R}^{n_2}$ be given vectors. Find $\mathbf{u}$ and $\mathbf{v}$ satisfying

$$\begin{aligned} 0 &= \mathbf{a} + \mathbf{A}\mathbf{u} + \mathbf{C}\mathbf{v} \\ 0 \leq \mathbf{v} \perp \mathbf{b} &+ \mathbf{D}\mathbf{u} + \mathbf{B}\mathbf{v} \geq 0 \end{aligned} \tag{2.31}$$

If the matrix $\mathbf{A}$ is non-singular in the mixed LCP (2.31), the mixed LCP can be converted in the standard LCP [29] by letting $\mathbf{q} = \mathbf{b} - \mathbf{D}\mathbf{A}^{-1}\mathbf{a}$ and $\mathbf{M} = \mathbf{B} - \mathbf{D}\mathbf{A}^{-1}\mathbf{C}$.

Definition 4 (Linear Complementarity Problem)**.** Let $\mathbf{M} \in \mathbb{R}^{n \times n}$ and $\mathbf{q} \in \mathbb{R}^{n}$ be given vectors. Find $\mathbf{z} \in \mathbb{R}^{n}$ satisfying

$$0 \leq \mathbf{z} \perp \mathbf{q} + \mathbf{M}\mathbf{z} \geq 0 \tag{2.32}$$

2.3.1 Finding Solutions of Complementarity Problems

There is no guaranteed solution method for general LCPs except for enumeration, which has a cost of $O(2^n)$ where n is the size of the LCP matrix. It is well known that a LCP$(\mathbf{q}, \mathbf{M})$ can be solved in polynomial time if $\mathbf{M}$ is a positive semi-definite matrix [80]. There are a number of algorithms available for finding solutions of LCPs and they generally fall into one of two categories, direct or iterative.

Direct methods are the most accurate methods for solving LCPs and are guaranteed to terminate in finite time. These solvers include Lemke's algorithm [80, 65], the Dantzig-Cottle method [30], Murty's method [79], and numerically improved Lemke [59]. However, Lemke's method solves the largest class of problems, including being one of the most robust algorithms available for solving LCPs arising from the complementarity based formulation of frictional multi-rigid-body dynamics (section 2.6). Mathematically, Lemke's method will solve an LCP($\mathbf{M}, \mathbf{q}$) if $\mathbf{M}$ is a copositive plus matrix. Such matrices are defined as:

$$\mathbf{u}^T\mathbf{M}\mathbf{u} \geq 0 \quad \text{for all } \mathbf{u} \geq 0, \tag{2.33}$$

$$\text{if } \mathbf{u}^T\mathbf{M}\mathbf{u} = 0 \text{ and } \mathbf{u} \geq 0, \text{ then } (\mathbf{M} + \mathbf{M}^T)\mathbf{u} = 0 \tag{2.34}$$

where the first equation is the definition of copositive matrices, and the second equation is the restriction for being plus. However, when solving mixed LCPs, Lemke's algorithm must convert the mixed LCP into a pure LCP, which is usually done with Schur complements (*i.e.* pivot out the equation and then perform Lemke's method on the reduced problem). This matrix operation is expensive, removes the sparcity of the matrices, and when the matrix becomes degenerate or ill-conditioned can even sometimes fail. An active area of research is in improving Lemke's method for these ill-conditioned LCPs, *e.g.* [59]. In addition, Lemke's method is not amendable to warm starting with previous solutions, which leads to practical inefficiencies. In practice, algorithms based on Lemke's method exhibit an average runtime of $O(n^3)$, where n is the rank of the matrix and, therefore, do not scale as well as iterative methods.

Indirect methods, or iterative methods, have the advantages of being much easier to implement and they can scale to much larger problems, but they do not have a finite time termination guarantee like the direct methods. In fact, it has been shown in [40] that iterative methods require a large number of iterations to converge to both accurate contact constraint enforcement and accurate friction response. Most of the iterative algorithms for solving LCPs are specializations of nonlinear programming algorithms applied to quadratic programs that are equivalent to the original LCP [80] (*i.e.* the solution of the LCP is the equilibrium point of an associated quadratic program). However, in the LCPs for frictional contact that arise in section 2.6, the coupling of the normal force with the frictional forces means the LCP cannot be reformulated as a quadratic program. In fact, it does not correspond to a minimization or even a saddle point problem.

Iterative schemes for LCPs of this form reply upon splitting methods. One type of splitting method is known as matrix splitting [84]. Consider the LCP $(\mathbf{q}, \mathbf{M})$ of size n. If $\mathbf{M} = \mathbf{B} + \mathbf{C}$, it is said to be a splitting of the matrix $\mathbf{M}$. For matrix splitting methods to be useful, the LCP$(\mathbf{p}, \mathbf{B})$ should be easily solved, *e.g.* $\mathbf{B}$ triangular with positive diagonals, which corresponds to the frictionless LCP.

Another splitting technique is known as operator splitting, where the original LCP problem is broken apart into two problems. The solution to the original problem is found by finding a solution that satisfies both subproblems simultaneously. The solution process becomes, find a solution to subproblem 1, use that solution as input to subproblem 2, repeat until convergence satisfied. For this method to be practical, it relies on each of the subproblems being easier to solve than the original problem, *e.g.* each subproblem can be reformulated as a convex quadratic program. This is a popular solution method since it is amendable to warm starting, scales well, is easy to implement, and appears to converge in practice in a few iterations; however there are no convergence guarantees. Papers based on this method include, [51, 56, 35, 8, 40, 60].

2.4 Complementarity formulation of Dynamics

There are several requirements the complementarity formulation of dynamics must satisfy. Firstly, it must satisfy the Newton-Euler equations of motion (section 2.2) and the kinematic update law (section 2.1.4.1). Additionally, it must allow for joint constraints, which constrain the relative positions of two bodies, which in turn impose velocity and acceleration constraints. There are also several requirements on how to handle contact constraints. The model must prevent interpenetration of the bodies at the contacts, must enforce that the contact forces are compressive (*i.e.* contact forces cannot pull the bodies together), must enforce a friction law, and lastly must have an impact resolution law.

The Newton-Euler equations and kinematic update law form a system of ordinary differential equations. The joint constraints will be discussed in section 2.4.1 and add a system of (nonlinear) algebraic equations. The contact conditions will be discussed in section 2.4.2, and are given by a system of complementarity constraints.

Considering all parts together, the dynamic model is a differential complementarity problem (DCP) as defined in definition 1.

To describe the dynamic model mathematically, we re-introduce some notation. Let $\mathbf{q}_j$ be the position and orientation of a frame attached to body j's center of mass in an inertial frame and $\boldsymbol{\nu}_j$ be the velocity twist of the origin of the frame. The generalized coordinates, $\mathbf{q}$, and generalized velocity, $\boldsymbol{\nu}$ of the whole system are formed by concatenating $\mathbf{q}_j$ and $\boldsymbol{\nu}_j$ respectively.

The Newton-Euler equations (2.29)–(2.30) can be compactly written in matrix form as:

$$\mathbf{M}(\mathbf{q}, t)\dot{\boldsymbol{\nu}} = \boldsymbol{\lambda}_{\text{vp}}(\mathbf{q}, \dot{\mathbf{q}}, t) + \boldsymbol{\lambda}_{\text{app}}(\mathbf{q}, t) \tag{2.35}$$

where

$$\mathbf{M} = \begin{bmatrix} m_1\mathbf{I} & 0 & & 0 & 0 \\ 0 & \boldsymbol{\mathcal{I}}_1 & & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & & m_n\mathbf{I} & 0 \\ 0 & 0 & & 0 & \boldsymbol{\mathcal{I}}_n \end{bmatrix}$$

Additionally, we split apart the applied wrenches into those dependent on configuration, $\boldsymbol{\lambda}_{\text{app}}$, and those dependent on configuration and velocity, $\boldsymbol{\lambda}_{\text{vp}}$. For example, the force due to gravity would appear in $\boldsymbol{\lambda}_{\text{app}}$, whereas the velocity product term in the Euler equation ($\boldsymbol{\omega} \times \mathcal{I}\boldsymbol{\omega}$) would appear in $\boldsymbol{\lambda}_{\text{vp}}$.

The kinematic update law (section 2.1.4.1) can also be written compactly for all bodies:

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\boldsymbol{\nu} \tag{2.36}$$

where

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} \mathbf{G}(\mathbf{q}_1) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{G}(\mathbf{q}_n) \end{bmatrix}, \quad \mathbf{G}(\mathbf{q}_i) = \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}(\mathbf{q}_i) \end{bmatrix}$$

and $\mathbf{J}(\mathbf{q}_i) \in \mathbb{R}^{4\times 3}$ was defined in equation 2.16 and relates the change in the ith body's Euler parameters to its angular velocity.

Taken together, equations (2.35) and (2.36) are enough to model unconstrained rigid body motion subject to applied wrenches. Simulation of a set of unconstrained rigid bodies is insufficient for use in studying robotics. Robotic systems are generally composed of links that are connected by simple joints constraining the relative motion of the connected bodies. The links may move either in free space (*e.g* moving a robot arm to an object) or in contact with an object, (*e.g* grasping the target object). This section explains how we model these constraints mathematically.

The two types of constraints considered in this thesis, joints and contact, are mathematically modeled differently. Joint constraints fall into a class of constraints known as *bilateral* constraints and are mathematically modeled with algebraic equations. Contact constraints kinematically fall into a class of constraints known as *unilateral* constraints, which are mathematically modeled as algebraic inequalities. However, in the dynamical analysis of contact, there are relationships involving forces that are mathematically modeled with complementarity conditions.

2.4.1 Equality Constraints

Joints constrain the relative positions of two bodies, which in turn impose velocity and acceleration constraints. Mathematically, joint constraints are modeled as algebraic equations involving the unknowns of the system,

$$\Phi_i(\mathbf{q}, t) = 0 \tag{2.37}$$

where Φ_i is the constraint function for the particular joint i and $\mathbf{q}$ is the vector of all bodies' configurations concatenated together.

The joint constraint function can be differentiated to obtain the constraint at the velocity level:

$$\frac{d\Phi_i}{dt} = \frac{\partial \Phi_i(\mathbf{q}, t)}{\partial \mathbf{q}}\dot{\mathbf{q}} + \frac{\partial \Phi_i(\mathbf{q}, t)}{\partial t}\dot{t} = 0 \tag{2.38}$$

Using equation (2.36) to substitute $G(\mathbf{q})\boldsymbol{\nu}$ for $\dot{\mathbf{q}}$ and the fact that $\frac{dt}{dt} = 1$, the velocity level constraint simplifies to:

$$0 = \mathbf{W}_{i\mathrm{b}}^{\mathrm{T}}(\mathbf{q}, t)\boldsymbol{\nu} + \frac{\partial \Phi_i(\mathbf{q}, t)}{\partial t} \tag{2.39}$$

where $\mathbf{W}_{i\mathrm{b}}(\mathbf{q},t)^{\mathrm{T}} = \dfrac{\partial \Phi_i(\mathbf{q},t)}{\partial \mathbf{q}} G(\mathbf{q})$ is known as the constraint Jacobian[3] matrix. The constraint Jacobian $\dfrac{\partial \Phi_i(\mathbf{q},t)}{\partial \mathbf{q}}$ takes on different forms dependent on the desired joint [28, 39].

Similarly, to express the kinematic constraint at the acceleration level one would take the derivative one more time.

Let there be n_b bilateral constraints and let $\mathbf{W}_b^{\mathrm{T}} = \begin{bmatrix} \mathbf{W}_{1\mathrm{b}}^{\mathrm{T}} \\ \vdots \\ \mathbf{W}_{n_b\mathrm{b}}^{\mathrm{T}} \end{bmatrix}$ be the concatenated vector of all the bilateral wrenches. The bilateral constraints for the entire system, specified at the velocity level, can be written as:

$$0 = \mathbf{W}_{\mathrm{b}}^{\mathrm{T}}(\mathbf{q},t)\boldsymbol{\nu} + \frac{\partial \Phi(\mathbf{q},t)}{\partial t} \tag{2.40}$$

Since the joint constraints are typically a function of one or two bodies, the matrix $\mathbf{W}_b^{\mathrm{T}}$ tends to be very sparse.

There are numerical difficulties that arise when specifying the position level constraints at a velocity or acceleration level, most noticeable being drift. Later sections will describe various techniques available for stabilizing these constraints.

2.4.1.1 Bilaterally Constrained Dynamics Formulation

We can reuse the bilateral constraint Jacobian $\mathbf{W}_{\mathrm{b}}^{\mathrm{T}}$ to relate the joint wrenches to the center of mass of each body. Each column of $\mathbf{W}_{\mathrm{b}}^{\mathrm{T}}$ is the wrench corresponding to a unit force applied to the body's center of mass. In other words, each column of $\mathbf{W}_{\mathrm{b}}^{\mathrm{T}}$ is a basis vector for the constraint force. This allows us to solve for a vector of constraint force magnitudes, $\boldsymbol{\lambda}_{\mathrm{b}}$ coupled to the system. With these new constraint forces, we can augment equations (2.35) and (2.36) with the new constraint forces and the constraint equation arriving at the following system of mixed differential-

[3] In robotics, the Jacobian is sometimes referred to as a "Wrench" matrix, since the rows of the matrix are unit wrenches.

algebraic equations (DAE) of index[4] 3:

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{G}(\mathbf{q})\boldsymbol{\nu} \\ \mathbf{M}(\mathbf{q},t)\dot{\boldsymbol{\nu}} &= \mathbf{W}_{\mathrm{b}}(\mathbf{q},t)\boldsymbol{\lambda}_{\mathrm{b}} + \boldsymbol{\lambda}_{\mathrm{app}}(\mathbf{q},t) + \boldsymbol{\lambda}_{\mathrm{vp}}(\mathbf{q},\boldsymbol{\nu},t) \\ \Phi(\mathbf{q},t) &= 0\end{aligned} \tag{2.41}$$

where $\boldsymbol{\lambda}_{\mathrm{b}}$ is the vector of bilateral constraint force magnitudes and $\mathbf{W}_{\mathrm{b}}$ is the transpose of the bilateral constraint Jacobian.

2.4.2 Contact

Contact constraints arise between pairs of bodies and serve two roles, prevent the bodies from inter-penetrating and effect a friction law at the point (or patch) of contact. There are two common approaches for handling contacts, *Rigid* and *Visco-Elastic* methods.

2.4.2.1 Rigid Contact

There are difficulties inherent with rigid contact constraints, the most difficult being that an impact between rigid bodies is a non-smooth event. During this impact phase, there is an instantaneous change in the velocity of the bodies to prevent penetration. This instantaneous change in the velocity is caused by an impulsive force and makes integrating through an impact event problematic. Another difficulty is that contacts are intermittent, they appear when two bodies touch, and disappear when the bodies separate. A contact law must allow for both the transition from no contact to contact, and vice versa. Lastly, friction laws can lead to non-unique solutions or no solutions to the system [107].

When two bodies j and k ($j \neq k$) are in (potential) contact (figure 2.5), we label the contact point as i, and consider it uniquely associated with the pair (j,k). Let n_c denote the total number of contact points at the current time t. Each contact point i of bodies j and k defines the origin of a contact frame Λ_i. Let $\hat{\mathbf{n}}_i$ denote the unit contact "normal" vector, that is the normalized vector from the contact

[4] the index is one plus the number of differentiations of the constraints that are needed in order to eliminate the Lagrange multipliers

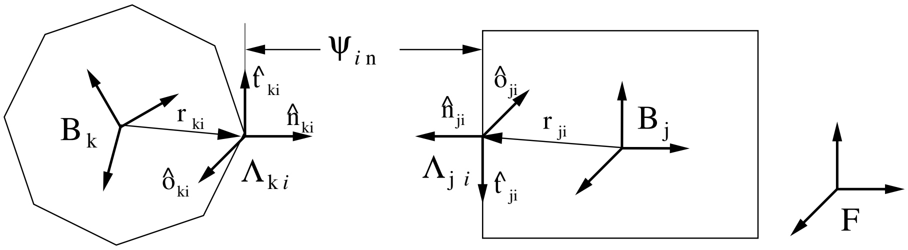


Figure 2.5: Coordinate system for contact constraints

point on j to the contact point on k. Let ψ_{in} denote the signed distance between the contact points on bodies j and k (by definition of the normal vector, ψ_{in} is the distance in the normal direction). The other 2 axes[5] of the contact frame Λ_i, denoted $\hat{\mathbf{t}}_i$ and $\hat{\mathbf{o}}_i$, span the contact tangent plane.

When modeling the contact constraints, it becomes convenient to break them into their two components: the *normal* and *tangential* constraints.

Normal Contact (Nonpenetration) Constraints The normal contact constraint prevents inter-penetration of the bodies, but must also allow for separation. Physically, the contact forces are compressive, meaning the contact force cannot pull the two bodies together. For each contact, we there is signed distance function $\psi_{in}(\mathbf{q}, t)$, which equals 0 when bodies j and k are touching at contact i, the positive distance when the two bodies are not touching at i, and returns negative penetration distance[6] when the bodies are inter-penetrating at i. Equation (2.36) provides a connection between the distance functions and the matrix $\mathbf{W}_\mathrm{n}$ as follows: $\mathbf{W}_\mathrm{n}^T = \frac{\partial \boldsymbol{\psi}_\mathrm{n}}{\partial \mathbf{q}}\mathbf{G}$. Note that one can define analogous (local) tangential displacement functions $\boldsymbol{\psi}_\mathrm{t}$ and $\boldsymbol{\psi}_\mathrm{o}$ with elements $\psi_{i\mathrm{t}}$ and $\psi_{i\mathrm{o}}$ for which the following hold: $\mathbf{W}_\mathrm{t}^T = \frac{\partial \boldsymbol{\psi}_\mathrm{t}}{\partial \mathbf{q}}\mathbf{G}$ and $\mathbf{W}_\mathrm{o}^T = \frac{\partial \boldsymbol{\psi}_\mathrm{o}}{\partial \mathbf{q}}\mathbf{G}$.

Analogous to before, we stack all the active contact gap functions into a single vector, $\boldsymbol{\psi}_\mathrm{n}$ obtaining the constraint:

$$\boldsymbol{\psi}_\mathrm{n}(\mathbf{q}, t) \geq 0 \tag{2.42}$$

[5] In 2D systems there is only one other axis. We would remove either the **o** or the **t** axis.

[6] The distance function must be able to return a negative distance if two bodies are overlapping for numerical stability. Otherwise, if a small penetration occurs from numerical errors and the distance at contact i did not become negative, the penetration will continue to get worse.

Let $\lambda_{i\mathrm{n}}$ be the magnitude of the normal contact force at the ith contact point and let $\boldsymbol{\lambda}_\mathrm{n}$ be the concatenated vector of all normal contact force magnitudes. Unlike the joint constraints, the normal contact forces at a contact cannot pull the bodies together:

$$\boldsymbol{\lambda}_\mathrm{n} \geq 0 \tag{2.43}$$

Additionally, at each contact there is a naturally occurring disjunctive relationship between the normal gap, $\psi_{i\mathrm{n}}$, and normal contact force, $\lambda_{i\mathrm{n}}$. Namely, if the contact is producing a normal contact force ($\lambda_{i\mathrm{n}} \geq 0$) then the normal distance between the two bodies must be zero ($\psi_{i\mathrm{n}} = 0$). Conversely, if there is a gap between the two bodies ($\psi_{i\mathrm{n}} \geq 0$), then the normal contact force must be zero ($\lambda_{i\mathrm{n}} = 0$). This final constraint can be written as:

$$\boldsymbol{\psi}_\mathrm{n}(\mathbf{q}, t)^\mathrm{T} \boldsymbol{\lambda}_\mathrm{n} = 0 \tag{2.44}$$

Equations (2.42), (2.43), and (2.44) taken together represent the nonpenetration constraint, which is a complementarity constraint and is often written using a more compact notation:

$$0 \leq \boldsymbol{\psi}_\mathrm{n}(\mathbf{q}, t) \perp \boldsymbol{\lambda}_\mathrm{n} \geq 0 \tag{2.45}$$

Tangential Contact Constraints (Friction) The contact forces in the tangential plane must satisfy a given friction model. In this section, Coulomb's dry friction law is presented. Coulomb's law is the most common model of friction available, derived in the 18th century after many experimental trials. There are many variants on Coulomb's original law, *e.g.* [115], and while the basic law still stirs up controversy, it has been successfully used in practice and makes a good law for presentation.

Coulomb's friction law requires that the contact force remain within a cone defined by the magnitude of the normal force, $\boldsymbol{\lambda}_\mathrm{n}$ and the coefficient of friction μ. When the contact is rolling, the magnitude of the friction force is bounded by $\mu\boldsymbol{\lambda}_\mathrm{n}$ with no constraints on the direction; when the contact point is sliding, the direction of the friction force must be opposite the sliding direction, and the magnitude must be exactly $\mu\boldsymbol{\lambda}_\mathrm{n}$.

This law can be written mathematically using the maximum dissipation principle, which states that given a normal force magnitude and relative slipping at the contact, the frictional force is the one that maximizes the rate of energy dissipation. Let $\lambda_{in} \geq 0$, let $\mathcal{F}_i(\mu_i, \lambda_{i\mathrm{n}})$ denote the friction cone at contact i:

$$\mathcal{F}_i(\mu_i, \lambda_{i\mathrm{n}}) = \{(\lambda_{i\mathrm{t}}, \lambda_{i\mathrm{o}}) : \; \mu_i^2 \lambda_{i\mathrm{n}}^2 - \lambda_{i\mathrm{t}}^2 - \lambda_{i\mathrm{o}}^2 \geq 0\} \tag{2.46}$$

where μ_i is the coefficient of friction acting at contact i. Next, define orthogonal sliding velocity components $v_{i\mathrm{t}}$ and $v_{i\mathrm{o}}$. The vectors of sliding velocities for all the contacts are: $\mathbf{v}_\mathrm{t} = \mathbf{W}_\mathrm{t}^T \boldsymbol{\nu} + \frac{\partial \boldsymbol{\psi}_\mathrm{t}}{\partial t}$ and $\mathbf{v}_\mathrm{o} = \mathbf{W}_\mathrm{o}^T \boldsymbol{\nu} + \frac{\partial \boldsymbol{\psi}_\mathrm{o}}{\partial t}$ with i^th elements $v_{i\mathrm{t}} = \mathbf{W}_{i\mathrm{t}}^T \boldsymbol{\nu} + \frac{\partial \psi_{i\mathrm{t}}}{\partial t}$ and $v_{i\mathrm{o}} = \mathbf{W}_{i\mathrm{o}}^T \boldsymbol{\nu} + \frac{\partial \psi_{io}}{\partial t}$, respectively. Then using the maximum dissipation principle, Coulomb's law at contact i may be written as follows:

$$(\lambda_{i\mathrm{t}}, \lambda_{i\mathrm{o}}) \in \underset{(\lambda_{i\mathrm{t}}, \lambda_{io}) \in \mathcal{F}_i}{\arg\max} \left(-\lambda_{i\mathrm{t}} v_{i\mathrm{t}} - \lambda_{io} v_{io} \right), \tag{2.47}$$

The equations that the solution of this optimization problem must satisfy cannot be determined from the Karush-Kuhn-Tucker (KKT) necessary conditions [16] because there are no applicable regularity conditions at a solution with $\lambda_{i\mathrm{n}} = 0$ or $\mu_i = 0$. To obtain the optimality conditions, one must use the Fritz–John conditions [16], and for completeness we present the derivation as it has appeared elsewhere in the literature [113, 103]. The Fritz–John optimality conditions that the friction forces at contact i must satisfy are:

$$0 = u_{i0} v_{i\mathrm{t}} + u_i 2 \lambda_{i\mathrm{t}} \tag{2.48}$$

$$0 = u_{i0} v_{i\mathrm{o}} + u_i 2 \lambda_{i\mathrm{o}} \tag{2.49}$$

$$0 \leq u_i \perp \mu_i^2 \lambda_{i\mathrm{n}}^2 - \lambda_{i\mathrm{t}}^2 - \lambda_{i\mathrm{o}}^2 \geq 0 \tag{2.50}$$

$$u_{i0} \geq 0, \qquad (u_{i0}, u_i) \neq 0 \tag{2.51}$$

where u_{i0} and u_i are Lagrange multipliers arising from the conversion.

Unlike the KKT conditions, the Fritz–John conditions require an extra multiplier that must be solved for, u_{i0}. In the literature, u_{i0} is replaced with $\mu_i \lambda_{i\mathrm{n}}$, leaving only one multiplier to solve for, u_i. Obviously $\mu_i\lambda_{in} \geq 0$, and when $\mu_i\lambda_{\text{in}} = 0$, a non-zero u_i trivially satisfies the Fritz–John conditions. The constraints in equation (2.51) can now be removed, and the substitution also provides a physical interpretation to the u_i Lagrange multiplier, it is half the slip speed at the ith contact point.

To see this and to also arrive at the commonly shown equations we will now show the algebraic steps to making this substitution. First, if $u_i = 0$, this implies that $u_{i0} > 0$. Otherwise, if $u_i > 0$ we can solve for λ_{it} and λ_{io} from equations (2.48) and (2.49), respectively:

$$\lambda_{it} = \frac{u_{i0}v_{\text{it}}}{2u_i} \qquad\qquad \lambda_{io} = \frac{u_{i0}v_{\text{io}}}{2u_i}. \tag{2.52}$$

and recognize that the right hand side of the complementarity constraint in equation (2.50) must now equal 0. Substituting λ_{it} and λ_{io} into the right hand side of the complementarity constraint, which must now be an equality condition, results in:

$$\mu_i^2\lambda_{\text{in}}^2 = \left(\frac{u_{i0}v_{\text{it}}}{2u_i}\right)^2 - \left(\frac{u_{i0}v_{\text{io}}}{2u_i}\right)^2 \tag{2.53}$$

and solving for u_{i0}

$$u_{i0} = \frac{2u_i\mu_i\lambda_{\text{in}}}{\sqrt{v_{\text{it}}^2 + v_{\text{io}}^2}} \tag{2.54}$$

Substituting u_{i0} into equations (2.48) and (2.49) we arrive at

$$0 = \frac{2u_i\mu_i\lambda_{\text{in}}}{\sqrt{v_{\text{it}}^2 + v_{\text{io}}^2}}v_{\text{it}} + u_i2\lambda_{it} \tag{2.55}$$

$$0 = \frac{2u_i\mu_i\lambda_{\text{in}}}{\sqrt{v_{\text{it}}^2 + v_{\text{io}}^2}}v_{\text{io}} + u_i2\lambda_{\text{io}} \tag{2.56}$$

which simplifies to:

$$0 = \mu_i\lambda_{\text{in}}v_{\text{it}} + \sigma_i\lambda_{\text{it}} \tag{2.57}$$

$$0 = \mu_i\lambda_{\text{in}}v_{\text{it}} + \sigma_i\lambda_{io} \tag{2.58}$$

where $\sigma_i = \sqrt{v_{\text{it}}^2 + v_{\text{io}}^2}$, the slip speed at contact i.

We have now replaced u_{i0} with $\mu_i\lambda_{in}$ and u_i with σ_i in equations (2.48) and (2.49). Replacing u_i with σ_i in the complementarity constraint (equation (2.50)) also works. When $u_1 > 0$, the right hand side of equation (2.54) is positive, and therefore $\sigma_i^2 > 0 \rightarrow \sigma_i > 0$. When $u_1 = 0$, this implies $u_{i0} > 0$, and from equations (2.48) and (2.49) $v_{i\text{t}} = 0$ and $v_{i\text{o}} = 0$, implying that $\sigma_i = 0$.

We have now shown through algebraic manipulation that after replacing u_{i0} with $\mu\lambda_{in}$ and using σ_i as the remaining Lagrange multiplier equations (2.48)– (2.50) still hold, and we can now remove equation (2.51). Hence, we arrive at the formulation commonly seen in the literature:

$$0 = \mu_i\lambda_{in}v_{it} + \lambda_{it}\sigma_i \tag{2.59}$$

$$0 = \mu_i\lambda_{in}v_{io} + \lambda_{io}\sigma_i \tag{2.60}$$

$$0 \leq \sigma_i \perp \mu_i^2\lambda_{in}^2 - \lambda_{it}^2 - \lambda_{io}^2 \geq 0 \tag{2.61}$$

Compactly, Coulomb's law for all contacts is:

$$0 = (\mathbf{U}\boldsymbol{\lambda}_{\text{n}})\circ(\mathbf{v}_{\text{t}}) + \boldsymbol{\lambda}_{\text{t}}\circ\boldsymbol{\sigma} \tag{2.62}$$

$$0 = (\mathbf{U}\boldsymbol{\lambda}_{\text{n}})\circ(\mathbf{v}_{\text{o}}) + \boldsymbol{\lambda}_{\text{o}}\circ\boldsymbol{\sigma} \tag{2.63}$$

$$0 \leq \boldsymbol{\sigma} \perp (\mathbf{U}\boldsymbol{\lambda}_{\text{n}})\circ(\mathbf{U}\boldsymbol{\lambda}_{\text{n}}) - \boldsymbol{\lambda}_{\text{t}}\circ\boldsymbol{\lambda}_{\text{t}} - \boldsymbol{\lambda}_{\text{o}}\circ\boldsymbol{\lambda}_{\text{o}} \geq 0 \tag{2.64}$$

where $\mathbf{U}$ is the diagonal matrix with i^{th} diagonal element equal to μ_i and $\circ$ connotes the Hadamard product, *i.e.* $\mathbf{u} \circ \mathbf{v} = [u_1v_1\ u_2v_2 \ldots u_nv_n]^T$.

Equations (2.62)–(2.64) are nonlinear (quadratic) in the unknowns, so their direct use in a time-stepping scheme would require the solution of a mixed nonlinear complementarity problems. In order to obtain a scheme based on mixed LCPs [110], a piecewise linear approximation of the quadratic friction cone with nonnegative force variables is needed (see figure 2.6). Let n_{d} friction force direction vectors d_j be chosen such that they positively span the space of possible friction forces, and let $(\lambda_{i\text{f}})_j$ be the friction force components in those directions. Also, let $(\psi_{i\text{f}}(\mathbf{q}, t))_j$ be the corresponding (local) tangential displacement function.

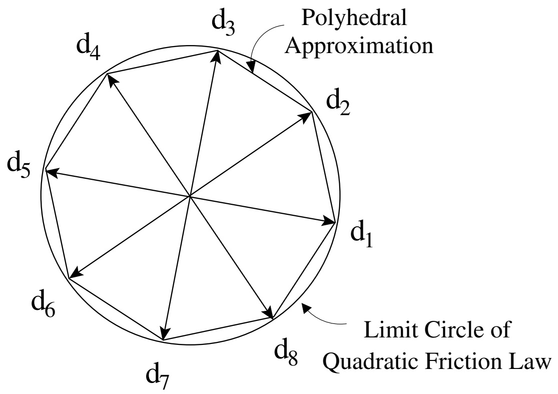


Figure 2.6: Friction cone approximated by an eight-sided pyramid defined by friction direction vectors d_j.

The frictional force can be approximated as $\mathbf{W}_\mathrm{f}(\mathbf{q})\boldsymbol{\lambda}_\mathrm{f}$, where $\boldsymbol{\lambda}_\mathrm{f} \in \mathbb{R}^{n_\mathrm{c}n_\mathrm{d}}$ has n_c elements $\boldsymbol{\lambda}_{i\mathrm{f}} \in \Re^{n_\mathrm{d}}$ with elements $(\lambda_{i\mathrm{f}})_j$, the vector $\boldsymbol{\psi}_\mathrm{f} \in \mathbb{R}^{n_\mathrm{c}n_\mathrm{d}}$ is defined analogously, and $\mathbf{W}_\mathrm{f}^T = \frac{\partial \boldsymbol{\psi}_\mathrm{f}}{\partial \mathbf{q}}\mathbf{G}$.

The approximate friction cone can be represented as:

$$\overline{\mathcal{F}}_i(\mu_i, \lambda_{i\mathrm{n}}) = \{\boldsymbol{\lambda}_{i\mathrm{f}} \mid \mu_i\lambda_{i\mathrm{n}} - \mathbf{e}^T\boldsymbol{\lambda}_{i\mathrm{f}} \geq 0,\ \boldsymbol{\lambda}_{i\mathrm{f}} \geq 0\} \tag{2.65}$$

where $\mathbf{e} \in \mathbb{R}^{n_\mathrm{d}}$ is vector of ones. Let $\mathbf{v}_{i\mathrm{f}} = [(v_{i\mathrm{f}})_1 \ \ldots \ (v_{i\mathrm{f}})_{n_\mathrm{d}}]^T = \frac{\partial \psi_{i\mathrm{f}}}{\partial \mathbf{q}}\mathbf{G}\boldsymbol{\nu} = \mathbf{W}_{i\mathrm{f}}^T\boldsymbol{\nu}$ be the vector of components of the sliding velocity at contact i in the friction directions. The approximate version of the dissipation condition becomes:

$$\boldsymbol{\lambda}_{i\mathrm{f}} \in \arg\max_{\boldsymbol{\lambda}_{i\mathrm{f}} \in \overline{\mathcal{F}}_i} \left(-\boldsymbol{\lambda}_{i\mathrm{f}}^T\mathbf{W}_{i\mathrm{f}}^T\boldsymbol{\nu}\right). \tag{2.66}$$

Reusing the slack variable σ_i (with slightly different meaning now), a useful equivalent LCP formulation of the maximum dissipation condition for the approximate friction cone is:

$$0 \leq \boldsymbol{\lambda}_{i\mathrm{f}} \perp \mathbf{W}_{i\mathrm{f}}^T\boldsymbol{\nu} + \mathbf{e}\sigma_i + \frac{\partial \boldsymbol{\psi}_{i\mathrm{f}}}{\partial t} \geq 0 \tag{2.67}$$

$$0 \leq \sigma_i \perp \mu_i\lambda_{i\mathrm{n}} - \mathbf{e}^T\boldsymbol{\lambda}_{i\mathrm{f}} \geq 0, \tag{2.68}$$

where now σ_i *approximates* the sliding speed at contact i. Maximum dissipation for all contacts can be written compactly as:

$$0 \leq \boldsymbol{\lambda}_{\mathrm{f}} \perp \mathbf{W}_{\mathrm{f}}^{T}\boldsymbol{\nu} + \mathbf{E}\boldsymbol{\sigma} + \frac{\partial \boldsymbol{\psi}_{\mathrm{f}}}{\partial t} \geq 0 \tag{2.69}$$

$$0 \leq \boldsymbol{\sigma} \perp \mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}} - \mathbf{E}^{T}\boldsymbol{\lambda}_{\mathrm{f}} \geq 0 \tag{2.70}$$

where $\mathbf{E}$ is the block diagonal matrix with i^{th} block on the main diagonal given by $\mathbf{e}$.

2.4.3 Instantaneous Formulation of Constrained Dynamics

If the quadratic form of Coulomb's friction law is used, we arrive at a nonlinear DCP formulation of the dynamics. If instead the discretized version of the law is used, we arrive at a linear DCP formulation. Both forms are presented below.

2.4.3.1 Nonlinear DCP Formulation

Using equations (2.41), (2.42)–(2.44), and (2.62)–(2.64), the nonlinear DCP can be written as:

$$\begin{aligned}
\mathbf{M}(\mathbf{q})\dot{\boldsymbol{\nu}} &= \mathbf{W}_{\mathrm{n}}(\mathbf{q})\boldsymbol{\lambda}_{\mathrm{n}} + \mathbf{W}_{\mathrm{t}}(\mathbf{q})\boldsymbol{\lambda}_{\mathrm{t}} + \mathbf{W}_{\mathrm{o}}(\mathbf{q})\boldsymbol{\lambda}_{\mathrm{o}} + \boldsymbol{\lambda}_{\mathrm{app}}(\mathbf{q},t) + \boldsymbol{\lambda}_{\mathrm{vp}}(\mathbf{q},\boldsymbol{\nu},t)\\
\dot{\mathbf{q}} &= \mathbf{G}(\mathbf{q})\boldsymbol{\nu}\\
0 &= \Phi(\mathbf{q},t)\\
0 &= (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}})\circ(\mathbf{v}_{\mathrm{t}}) + \boldsymbol{\lambda}_{\mathrm{t}}\circ\boldsymbol{\sigma}\\
0 &= (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}})\circ(\mathbf{v}_{\mathrm{o}}) + \boldsymbol{\lambda}_{\mathrm{o}}\circ\boldsymbol{\sigma}\\
0 \leq \boldsymbol{\lambda}_{\mathrm{n}} &\perp \boldsymbol{\psi}_{\mathrm{n}}(\mathbf{q},t) \geq 0\\
0 \leq \boldsymbol{\sigma} &\perp (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}})\circ(\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}) - \boldsymbol{\lambda}_{\mathrm{t}}\circ\boldsymbol{\lambda}_{\mathrm{t}} - \boldsymbol{\lambda}_{\mathrm{o}}\circ\boldsymbol{\lambda}_{\mathrm{o}} \geq 0
\end{aligned} \tag{2.71}$$

2.4.3.2 Linear DCP Formulation

Using equations (2.41), (2.42)–(2.44), (2.69), and (2.70), the linear DCP can be written as:

$$\begin{aligned}
\mathbf{M}(\mathbf{q})\dot{\boldsymbol{\nu}} &= \mathbf{W}_{\mathrm{n}}(\mathbf{q})\boldsymbol{\lambda}_{\mathrm{n}} + \mathbf{W}_{\mathrm{f}}(\mathbf{q})\boldsymbol{\lambda}_{\mathrm{f}} + \boldsymbol{\lambda}_{\mathrm{app}}(\mathbf{q},t) + \boldsymbol{\lambda}_{\mathrm{vp}}(\mathbf{q},\boldsymbol{\nu},t) \\
\dot{\mathbf{q}} &= \mathbf{G}(\mathbf{q})\boldsymbol{\nu} \\
0 &= \Phi(\mathbf{q},t) \\
0 \leq \boldsymbol{\lambda}_{\mathrm{n}} &\perp \boldsymbol{\psi}_{\mathrm{n}}(\mathbf{q},t) \geq 0 \\
0 \leq \boldsymbol{\lambda}_{\mathrm{f}} &\perp \mathbf{W}_{\mathrm{f}}^{T}\boldsymbol{\nu} + \mathbf{E}\boldsymbol{\sigma} + \frac{\partial \boldsymbol{\psi}_{\mathrm{f}}}{\partial t} \geq 0 \\
0 \leq \boldsymbol{\sigma} &\perp \mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}} - \mathbf{E}^{T}\boldsymbol{\lambda}_{\mathrm{f}} \geq 0
\end{aligned} \tag{2.72}$$

2.4.4 Visco-Elastic Contacts

While the above time-stepping method provides accurate simulation for a large class of problems, it does not handle well impact dynamics with restitution or phenomena such as micro-slip. In [102] a model appropriate for such situations was derived and a time-stepping method analogous to that above was developed. In this method, the surfaces near each contact are discretized and sets of coupled lumped springs and dampers are inserted between pairs of interior points (in the rigid core of the body) and boundary points (on the compliant surface of the body) on the discretized contact patch.

In this section we describe the 3D linear viscoelastic model of contact [63, 100]. For simplicity of exposition, we consider only one of the objects to be flexible at each contact. The general formulation where both the bodies are flexible will contain the additional constraint that the contact forces acting on both the bodies have to be equal.

Concatenating all the individual force components into vectors allows us to write for each contact (we drop subscript i for legibility),

$$\boldsymbol{\lambda} = \mathbf{K}\boldsymbol{\delta} + \mathbf{C}\dot{\boldsymbol{\delta}} \tag{2.73}$$

where $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_{\mathrm{n}}, \boldsymbol{\lambda}_{\mathrm{t}}, \boldsymbol{\lambda}_{\mathrm{o}}]$ and $\boldsymbol{\delta} = [\delta_{\mathrm{n}}, \delta_{\mathrm{t}}, \delta_{\mathrm{o}}]$ are 3×1 column vectors with $\boldsymbol{\delta}_{\mathrm{n}}, \boldsymbol{\delta}_{\mathrm{t}}, \boldsymbol{\delta}_{\mathrm{o}}$

being the normal and tangential deflection. The matrices $\mathbf{K}$, $\mathbf{C}$ are stiffness and damping matrices given by

$$\mathbf{K} = \begin{bmatrix} K_{\text{nn}} & K_{\text{nt}} & K_{\text{no}} \\ K_{\text{tn}} & K_{\text{tt}} & K_{\text{to}} \\ K_{\text{on}} & K_{\text{ot}} & K_{\text{oo}} \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} C_{\text{nn}} & C_{\text{nt}} & C_{\text{no}} \\ C_{\text{tn}} & C_{\text{tt}} & C_{\text{to}} \\ C_{\text{on}} & C_{\text{ot}} & C_{\text{oo}} \end{bmatrix}$$

For systems with multiple contacts, the contact forces $\boldsymbol{\lambda}$, and body deformations $\boldsymbol{\delta}$ become concatenations of n_c subvectors, where n_c is the number of contacts. The stiffness and damping matrices are block diagonal matrices of size $3n_c \times 3n_c$, where each diagonal block of size 3×3 represent one contact.

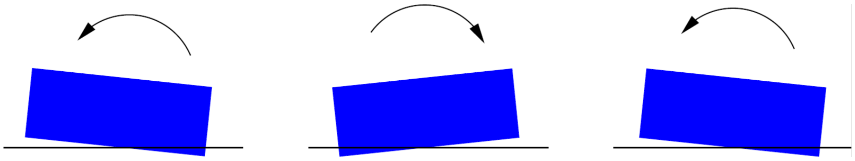


Figure 2.7: Sequence of simulation steps illustrating the oscillation problem arising from penalty based collision response methods.

One problem with penalty based methods is that of oscillation. In figure 2.7, a rectangular body is dropped onto a surface. The vertex of the body that penetrates the most gets the penalty force, and the body will rotate causing the most penetrating point to change. Then the penalty force is applied to the other vertex, causing the body to rotate in the other direction, and the other vertex becomes again the most penetrated. This switching will continue forever (without artificial damping), the end result being that the body never comes to rest on the surface and appears to vibrate. In addition, a compliant layer requires the time when contact first occurs to be determined accurately. To accommodate these needs, an adaptive time step adjustment is required [41].

Another method for handling locally compliant models was developed in [88]. Instead of using a grid of lumped springs and dampers to model the contact patch, the surface around a point of contact is modeled as an elastic half-space. Additionally, they avoid the problem of identifying point correspondences between the bodies, allowing them to separate the problems, and iteratively solve two smaller

LCPs instead of one large LCP.

The introduction of the compliant layer on the body surfaces causes the dynamic equations to become numerically stiff. Consequently, when contacts exist in the system, the size of the time step must be reduced significantly, by many orders of magnitude when the springs are very stiff. The very small time steps required by this method and the large number of variables that are added when contact patches are finely discretized motivate our philosophy to include a range of models and time-stepping methods in our simulation package (described in Chapter 4).

2.5 Solving Constrained Equations of Motion

One of the most popular traditional methods for solving the constrained dynamics equations (equations (2.72) or (2.71)) consists of the integrate-detect-restart methods [53]. Typically in these methods, an active set of unilateral constraints is chosen and the bilaterally constrained system is solved. After a solution to the bilaterally constrained system of equations is found, the solution is analyzed for violation of any unilateral constraint. If a violation is found, the time stepper must determine the time of violation, and go back to fix the violated constraint. These schemes have difficulties handling multiple unilateral constraints since there is no upper bound on the number of subproblems to solve in finite time [113].

More recently [66], unilateral constraints have been incorporated directly into the equations of motion as constraint impulses. This approach is similar to those used in the complementarity methods discussed in the next section, but the authors instead use a Gauss-Seidel iterative method to solve for the contact constraint impulses instead of complementarity techniques. The main benefit of simultaneously solving the contact constraints with discretized equations of motion is that at the end of a time step, the system is in a physically feasible state. No post-processing of the result is required.

In the following discussion, unilateral constraints will be ignored, and various techniques for solving the bilaterally constrained systems (DAE) of equations are presented. At the end of the section, a discussion on incorporating the unilateral constraints is made.

As stated earlier, typically the acceleration level form of the constraints is used, which allows us to represent equations (2.41) in matrix form as the following reduced index-1 DAE.

$$\begin{bmatrix} \mathbf{M}(\mathbf{q},t) & \mathbf{W}_{\mathrm{b}}(\mathbf{q},t) \\ \mathbf{W}_{\mathrm{b}}^{\mathrm{T}}(\mathbf{q},t) & 0 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\nu}} \\ \boldsymbol{\lambda}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda}_{\mathrm{app}}(\mathbf{q},t) + \boldsymbol{\lambda}_{\mathrm{vp}}(\mathbf{q},\boldsymbol{\nu},t) \\ \gamma(\mathbf{q},t) \end{bmatrix} \tag{2.74}$$

where $\gamma = -\left(\frac{\partial \mathbf{W}_{\mathrm{b}}(\mathbf{q},t)^{\mathrm{T}}}{\partial \mathbf{q}}\right)\boldsymbol{\nu} - 2\left[\left(\frac{\partial \mathbf{W}_{\mathrm{b}}^{\mathrm{T}}(\mathbf{q},t)}{\partial t}\right)\boldsymbol{\nu}\right] - \frac{\partial^2 \Phi(\mathbf{q},t)}{\partial t^2}$. The acceleration and Lagrange multiplier can be obtained as the solution of the following matrix form DAE:

$$\begin{bmatrix} \dot{\boldsymbol{\nu}} \\ \boldsymbol{\lambda}_b \end{bmatrix} = \begin{bmatrix} \mathbf{M}(\mathbf{q},t) & \mathbf{W}_{\mathrm{b}}(q,t) \\ \mathbf{W}_{\mathrm{b}}^{\mathrm{T}}(q,t) & 0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\lambda}_{\mathrm{app}}(\mathbf{q},t) + \boldsymbol{\lambda}_{\mathrm{vp}}(\mathbf{q},\boldsymbol{\nu},t) \\ \gamma(\mathbf{q},t) \end{bmatrix} \tag{2.75}$$

To guarantee the inverse exists, these methods must assume [91] that the following properties hold:

$$\begin{gathered} \mathbf{W}_{\mathrm{b}}(q,t) \text{ is full rank} \\ \mathbf{a}^{\mathrm{T}}\mathbf{M}(\mathbf{q},t)\mathbf{a} > 0 \quad \forall \mathbf{a} \in \ker \mathbf{W}_{\mathrm{b}}^{\mathrm{T}}(\mathbf{q},t), \quad \mathbf{a} \neq 0 \end{gathered}$$

As mentioned before, analytical solutions of (2.75) will also satisfy both the velocity level form of the constraints (2.39) and the original position level form (2.37), however, numerical solutions will not. Many methods have been developed for finding a numerical solution to equation (2.75), most of which fit into one of three categories: Constraint stabilization, Projected invariants, or State space.

2.5.1 Stabilization Methods

Early constraint stabilization methods were based on Baumgarte [15]. In his method, he observed the modified equation

$$\ddot{\Phi} + \alpha\dot{\Phi} + \beta\Phi = 0$$

where $\alpha,\ \beta \in \mathbb{R}$ and $\alpha,\ \beta > 0$ is stable. Therefore, the acceleration constraint equation is modified, becoming:

$$\mathbf{W}_{\mathrm{b}}(\mathbf{q},t)^{\mathrm{T}}\dot{\boldsymbol{\nu}} = \boldsymbol{\gamma}(\mathbf{q},t) - \alpha\left(\mathbf{W}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\nu} + \frac{\partial\Phi}{\partial t}\right) - \beta\Phi$$

The stabilization is then to simply use this modified version of the acceleration constraint in the integration. The problem with this method is that there is no reliable way to choose values for the coefficients α and β.

Another stabilization technique transforms the DAE into a system of first order ODEs. Looking back at equations (2.74) (and dropping functional dependencies for brevity),

$$\begin{aligned}\dot{\boldsymbol{\nu}} &= \mathbf{M}^{-1}\big(\boldsymbol{\lambda}_{\mathrm{app}} + \boldsymbol{\lambda}_{\mathrm{vp}}\big) + \mathbf{M}^{-1}\mathbf{W}_{\mathrm{b}}\boldsymbol{\lambda}_{b}\\ \mathbf{W}_{\mathrm{b}}^{\mathrm{T}}\dot{\boldsymbol{\nu}} &= \boldsymbol{\gamma}\end{aligned}$$

we can solve for $\boldsymbol{\lambda}_b$:

$$\boldsymbol{\lambda}_b = (\mathbf{W}_{\mathrm{b}}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{W}_{\mathrm{b}})^{-1}\left(\mathbf{W}_{\mathrm{b}}^{\mathrm{T}}\mathbf{M}^{-1}\big(\boldsymbol{\lambda}_{\mathrm{app}} + \boldsymbol{\lambda}_{\mathrm{vp}}\big) - \boldsymbol{\gamma}\right)$$

and substitute it back into equation (2.41) obtaining an ODE system for $\boldsymbol{\nu}$ and $\mathbf{q}$

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{G}\boldsymbol{\nu}\\ \mathbf{M}\dot{\boldsymbol{\nu}} &= \mathbf{W}_{\mathrm{b}}\Big[\big(\mathbf{W}_{\mathrm{b}}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{W}_{\mathrm{b}}\big)^{-1}\left(\mathbf{W}_{\mathrm{b}}^{\mathrm{T}}\mathbf{M}^{-1}\big(\boldsymbol{\lambda}_{\mathrm{app}} + \boldsymbol{\lambda}_{\mathrm{vp}}\big) - \boldsymbol{\gamma}\right)\Big] + \big(\boldsymbol{\lambda}_{\mathrm{app}} + \boldsymbol{\lambda}_{\mathrm{vp}}\big)\end{aligned} \tag{2.76}$$

However, as before there is a drift on the constraints at the velocity and position level. We could again apply Baumgarte's technique, however Ascher et al. [9] found difficulties with this method and present their own stabilization technique.

The underlying ODE can be written as

$$\dot{\mathbf{z}} = \hat{f}(\mathbf{z})$$

where $\mathbf{z} = (\mathbf{q},\ \boldsymbol{\nu})$ and a stabilization term is added to the right hand side

$$\dot{\mathbf{z}} = \hat{f}(\mathbf{z}) - \alpha \mathbf{F}(\mathbf{z})\mathbf{h}(\mathbf{z})$$

where $\alpha > 0$ is a parameter, $F(z)$ is a matrix with a variety of choices (see [9] for more details), and $h(z) = \begin{bmatrix} \Phi(\mathbf{q}, t) \\ \mathbf{W}_{\mathrm{b}}(\mathbf{q}, t)^{\mathrm{T}}\boldsymbol{\nu} \end{bmatrix}$. Similarly, the difficulties with this method are the selection of $\mathbf{F}(\mathbf{z})$ and α.

2.5.1.1 Projection Methods

Projection methods integrate all generalized coordinates at each time step. Consider the reduced index 1 DAE (2.74), or the system of ODEs (2.76) and the position and velocity constraints constraints (2.37), (2.39). Taken together, the system (2.74), (2.37), and (2.39) form an *overdetermined* DAE. Again, numerical integration will not satisfy the constraints, so additional multipliers are introduced.

There are two main ways to project the solution onto the constraint manifold. The first is to introduce an extra multiplier μ to insure the velocity constraint equation is also satisfied, and we arrive at the following stable index 2 DAE:

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{G}(\mathbf{q})\boldsymbol{\nu} - \mathbf{W}_{\mathrm{b}}(\mathbf{q}, t)\mu \\ \mathbf{M}\dot{\boldsymbol{\nu}} &= \mathbf{W}_{\mathrm{b}}(\mathbf{q}, t)\boldsymbol{\lambda}_b + \boldsymbol{\lambda}_{\mathrm{app}}(\mathbf{q}, t) + \boldsymbol{\lambda}_{\mathrm{vp}}(\mathbf{q}, \boldsymbol{\nu}, t) \\ \Phi(\mathbf{q}, t) &= 0 \\ \mathbf{W}_{\mathrm{b}}(\mathbf{q}, t)^{\mathrm{T}}\boldsymbol{\nu} &= 0 \end{aligned}$$

If desired, the DAE can be reduced to an index 1 and an additional multiplier can be introduced along with the requirement that the acceleration constraint be satisfied.

The other projection technique is known as coordinate projection. Unlike the above method where the derivative is modified by a multiplier to enforce the constraint, the integration of the underlying ODE is performed and the solution found at the end of the step is projected back onto the constraint manifold. Using this method, it is possible to choose which constraint manifold to project the solution onto, or possibly both.

2.5.1.2 State-Space Methods

In state space methods, the problem is reformulated into a local state-space form using a local parameterization of the constraint manifold [81]. If there are n generalized coordinates in q and $m < n$ constraints in $\Phi(\mathbf{q}, t)$, the dimension of the reduced ODE becomes $n - m$. To derive the local ODE, consider at each point t, a matrix $\mathbf{R} \in \mathbb{R}^{(n-m)\times n}$, such that $\begin{bmatrix} \mathbf{R} \\ \mathbf{W}_{\mathrm{b}}(\mathbf{q}, t)^{\mathrm{T}} \end{bmatrix}$ is nonsingular. The local ODE can be derived for the state-space variable

$$\mathbf{u} = \mathbf{R}\mathbf{q}$$

The choice of the constraint parameterization varies among algorithms.

An advantage of this formulation is the reduced size of the problem. However, the constraint equations are usually nonlinear, and a parameterization can be only locally determined. When the constraint is violated, a new parameterization must be formulated, adding to the computational overhead. It also requires a robust detection scheme, to determine violation. The nonlinearity also increases the effort required to obtain dependent generalized coordinates through the parameterization, since a solution of nonlinear equations is required.

2.5.2 Unilateral Constraints

To find a solution that satisfies the unilateral constraints, the points of discontinuity (switch points) must be found. Once these switch points are identified, we have a set of piece-wise smooth solutions between each set of them. At each switch there exists some transition law, and the physically correct solution must be chosen from a large set of possible states. This combinatorial searching becomes intensive very quickly. For example, if the unilateral constraint is a contact constraint (more on contact in the next section) with friction then there are three possible states: separate, roll, or slide. If there are n active constraints, the number of possible states becomes 3^n. This trial-and-approach soon becomes computationally intractable for large numbers of active unilateral constraints.

We have shown in the instantaneous formulation of dynamics with unilateral contact (2.72) that the system becomes a more complicated differential algebraic inequality. Baraff studied the multiple rigid body with contact problem [12], and determined that computing contact forces is NP-hard [10] in the presence of friction. As stated earlier, it is because of these computational issues that many traditional methods choose to identify the switch points, and solve piecewise smooth DAEs.

In constraint-based methods, the collision or impact forces have to be treated differently from the resting contact forces. This is because impact forces are discontinuous impulsive forces occurring for a single instant of time, unlike constraint forces which are present over an interval. These very large forces (theoretically infinite) over a very short period of time (theoretically infinitely small) are difficult to numerically integrate and are often times special cased. During a time step, one must find the time at which the collision occurred, and stop the integration at this point in time. The collision forces are then calculated using a collision response method (e.g. [122]). After these impulses for impact are computed, the new velocities are found and the integrator is restarted with these new initial conditions.

Another technique for handling rigid body impact is the so called impulse based method [76]. In this method, all modes of contact are handled via impulses applied to the body. Under this formulation, a block resting on a table is experiencing many tiny collisions with the table, and each of these collisions is resolved using only local information at the point of contact. One drawback of impulse-based formulations is that they become collision intensive, since collisions are used to determine all interactions between bodies and collision detection can often times become the bottleneck in dynamic simulation. Extra care must be taken to ensure that static or resting contact does not produce an energy loss to the system, since they are static workless forces (see [76] for more details).

2.6 Time-Stepping Formulation of Rigid Body Dynamics

A desirable outcome for any time-stepping scheme is that its solution at the end of each time step of the discrete-time model equals the (continuous) solution of the instantaneous-time model at the same time. Typically however, computational efficiency and/or convergence issues force one to design a scheme that does not exactly meet this outcome. To prepare for the design of a time-stepper that solves a linear problem for each time step, the quadratic friction cone was approximated by a piecewise linear cone. In the following, two time-stepping schemes will be presented. The unknowns for both are the velocity vector, configuration vector, contact forces, and sliding speeds at the end of the time step.

The Stewart-Trinkle formulation [110] builds upon the same complementarity principles developed earlier by Trinkle and Pang [113]. The difference being that the previous work was an instantaneous approach solving for forces and accelerations to update velocities and positions. As stated earlier, problems arise in force based formulations due to the discontinuities in velocity. In contrast, the Stewart-Trinkle method is position based, using complementarity principles for the position, solving for impulses and velocities at the end of a time step. They can incorporate impulses without any difficulties by using the integral of the forces over a time-step, which are finite even if there is an impulsive force.

Let t_ℓ denote the current time and h be the time step (the term h is also often called the step size). Use the superscripts ℓ and $\ell+1$ to denote quantities at beginning and end of the ℓth time step respectively. Using $\dot{\boldsymbol{\nu}} \approx (\boldsymbol{\nu}^{\ell+1} - \boldsymbol{\nu}^{\ell})/h$ and $\dot{\mathbf{q}} \approx (\mathbf{q}^{\ell+1} - \mathbf{q}^{\ell})/h$, where $\mathbf{q}^\ell = \mathbf{q}(t^\ell)$, we get the following nonlinear and linear discrete time systems.

2.6.1 Nonlinear Complementarity Problem Formulation

The nonlinear DCP from equation (2.71) is discretized as:

$$
\begin{aligned}
\mathbf{M}\boldsymbol{\nu}^{\ell+1} &= \mathbf{M}\boldsymbol{\nu}^{\ell} + h(\mathbf{W}_{\mathrm{n}}\boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1} + \mathbf{W}_{\mathrm{t}}\boldsymbol{\lambda}_{\mathrm{t}}^{\ell+1} + \mathbf{W}_{\mathrm{o}}\boldsymbol{\lambda}_{\mathrm{o}}^{\ell+1} + \boldsymbol{\lambda}_{\mathrm{app}}^{\ell} + \boldsymbol{\lambda}_{\mathrm{vp}}^{\ell})\\
\mathbf{q}^{\ell+1} &= \mathbf{q}^{\ell} + h\mathbf{G}\boldsymbol{\nu}^{\ell+1}\\
0 &= (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1})\circ(\mathbf{v}_{\mathrm{t}}^{\ell+1}) + \boldsymbol{\lambda}_{\mathrm{t}}^{\ell+1}\circ\boldsymbol{\sigma}^{\ell+1}\\
0 &= (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1})\circ(\mathbf{v}_{\mathrm{o}}^{\ell+1}) + \boldsymbol{\lambda}_{\mathrm{o}}^{\ell+1}\circ\boldsymbol{\sigma}^{\ell+1}\\
0 \le \boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1} &\perp \psi_{\mathrm{n}}(\mathbf{q}^{\ell+1}) \ge 0\\
0 \le \boldsymbol{\sigma}^{\ell+1} &\perp (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1})\circ(\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1}) - \boldsymbol{\lambda}_{\mathrm{t}}^{\ell+1}\circ\boldsymbol{\lambda}_{\mathrm{t}}^{\ell+1} - \boldsymbol{\lambda}_{\mathrm{o}}^{\ell+1}\circ\boldsymbol{\lambda}_{\mathrm{o}}^{\ell+1} \ge 0
\end{aligned}
\tag{2.77}
$$

where $\psi_{in}(\mathbf{q}^{\ell+1}, t_{\ell+1}) \approx \psi_{in}(\mathbf{q}^{\ell}, t_{\ell}) + h\mathbf{W}_{in}^T\boldsymbol{\nu}^{\ell+1} + \frac{\partial \psi_{\mathrm{n}}^{l}}{\partial t}$.

If we evaluate $\mathbf{W}_{(\cdot)}$ and $\boldsymbol{\lambda}_{\text{vp}}$ at $\ell+1$, we have a fully implicit formulation [24] with an approximation in the gap function. If we evaluate $\mathbf{W}_{(\cdot)}$ and $\boldsymbol{\lambda}_{\text{vp}}$ at ℓ, we recover the Stewart-Trinkle formulation [110] with quadratic friction law.

2.6.2 Linear Complementarity Problem Formulation

The linear DCP from equation (2.72) is discretized as:

$$\begin{bmatrix} 0 \\ 0 \\ \boldsymbol{\rho}_{\text{n}}^{l+1} \\ \boldsymbol{\rho}_{\text{f}}^{l+1} \\ \boldsymbol{s}^{l+1} \end{bmatrix} = \begin{bmatrix} -\mathbf{M} & \mathbf{W}_{\text{b}} & \mathbf{W}_{\text{n}} & \mathbf{W}_{\text{f}} & 0 \\ \mathbf{W}_{\text{b}}^{\text{T}} & 0 & 0 & 0 & 0 \\ \mathbf{W}_{\text{n}}^{\text{T}} & 0 & 0 & 0 & 0 \\ \mathbf{W}_{\text{f}}^{\text{T}} & 0 & 0 & 0 & \mathbf{E} \\ 0 & 0 & \mathbf{U} & -\mathbf{E}^{\text{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}^{\ell+1} \\ \mathbf{p}_{b}^{\ell+1} \\ \mathbf{p}_{\text{n}}^{\ell+1} \\ \mathbf{p}_{\text{f}}^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix} + \begin{bmatrix} \mathbf{M}\boldsymbol{\nu}^{l} + \mathbf{p}_{\text{app}} + \mathbf{p}_{\text{vp}} \\ \frac{\Phi^{\ell}}{h} + \frac{\partial \Phi^{\ell}}{\partial t} \\ \frac{\boldsymbol{\psi}_{\text{n}}^{\ell}}{h} + \frac{\partial \boldsymbol{\psi}_{\text{n}}^{\ell}}{\partial t} \\ \frac{\partial \boldsymbol{\Psi}_{\text{f}}^{\ell}}{\partial t} \\ 0 \end{bmatrix} \tag{2.78}$$

$$0 \leq \begin{bmatrix} \boldsymbol{\rho}_{\text{n}}^{\ell+1} \\ \boldsymbol{\rho}_{\text{f}}^{\ell+1} \\ \mathbf{s}^{\ell+1} \end{bmatrix} \perp \begin{bmatrix} \mathbf{p}_{\text{n}}^{\ell+1} \\ \mathbf{p}_{\text{f}}^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix} \geq 0 \tag{2.79}$$

where $\mathbf{p}_{(\cdot)} = h\boldsymbol{\lambda}_{(\cdot)}$ and $\boldsymbol{\rho}^{l+1} = \psi^{l+1}/h$.

2.7 Examples

In this section we clarify the complementarity based formulation of multi-rigid-body simulation through use of illustrative 2D examples. It begins the examples with a simple planar 2 bar pendulum and continues by adding more and more complexity to the subsequent examples.

For these examples, we must use the 2D analogs of the operators defined previously. There is no cross product for 2D vectors, so instead we use the perpendicular product, denoted by $\otimes$

$$\mathbf{u} \otimes \mathbf{v} = u_x v_y - u_y v_x \tag{2.80}$$

In addition, instead of a skew-symmetric matrix operator, we have a perpendicular operator:

$$\mathbf{a}^{\wedge} = \begin{bmatrix} -a_y \\ a_x \end{bmatrix} \tag{2.81}$$

2.7.1 Planar 2 Bar Pendulum

This system consists of a planar simple double pendulum with all joint axes parallel to the Z-axis. It is constructed of two slender rods A and B with masses m_A and m_B respectively. Link A has length L_A and is connected to ground at point $\mathbf{O}$. Link B has length L_B and is connected to link A at point $\mathbf{P}$ by point $\mathbf{P}'$. Frames are attached to each body and are fixed in that body. The position and orientation of the frames are used as the generalized coordinates. Figure 2.8 illustrates a free body diagram of the system.

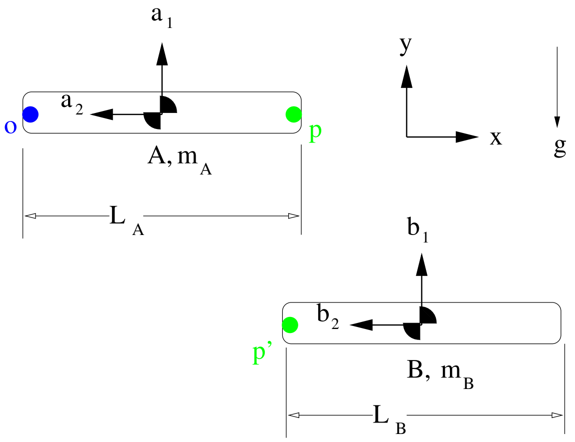


Figure 2.8: Free Body Diagram of Simple planar double pendulum.

The configuration $\mathbf{q}$ of the system is $\mathbf{q} = [\mathbf{q}^A\ \mathbf{q}^B]$ and similarly the velocity twist of the system is $\boldsymbol{\nu} = [\boldsymbol{\nu}^A\ \boldsymbol{\nu}^B]$.

For this system, the generalized mass matrix is

$$\mathbf{M} = \begin{bmatrix} m_A & 0 & 0 & 0 & 0 & 0 \\ 0 & m_A & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{12}m_A L_A^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_B & 0 & 0 \\ 0 & 0 & 0 & 0 & m_B & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{12}m_B L_B^2 \end{bmatrix} \tag{2.82}$$

where $1/12 m_A L_A^2$ is the moment of inertia for a slender rod.

There are 2 bilateral constraints on this system, the position of point O is fixed in the world frame and the position of points P and P' must lie in the same location. Let point $\mathbf{O}$ be $(0,0)$ in the fixed world frame. Next, attach a “joint” frame $\{\alpha\}$ with its origin at point $\mathbf{O}$. For computational simplicity align the joint frame axes with the world frame axes. We can mathematically write the first bilateral constraint ϕ_1 as

$$\phi_1 \triangleq ({}^{\alpha}_{F}\mathbf{R})\,\mathbf{A}^* + ({}^{\alpha}_{F}\mathbf{R})\left({}^{F}_{A}\mathbf{R}\right){}^{A}\mathbf{r}^{A^*O} = 0 \tag{2.83}$$

where $\mathbf{A}^*$ is the location of body A’s center of mass and $\mathbf{r}^{A^*O}$ is a vector from the center of mass to the point $\mathbf{O}$. However, since we aligned the joint frame with the fixed world frame, $({}^{\alpha}_{F}\mathbf{R})$ is the identity matrix and the constraint simplifies to:

$$\phi_1 \triangleq \mathbf{A}^* + \left({}^{F}_{A}\mathbf{R}\right){}^{A}\mathbf{r}^{A^*O} = 0 \tag{2.84}$$

The second bilateral constraint is that points $\mathbf{P}$ and $\mathbf{P}'$ lie at the same location. Again, attach a “joint” frame $\{\beta\}$ at points $\mathbf{P}$ and $\mathbf{P}'$ aligned with the world frame. The constraint that the points cannot separate can be written as:

$$\phi_2 \triangleq \mathbf{P} - \mathbf{P}' = \mathbf{A}^* + \left({}^{F}_{A}\mathbf{R}\right){}^{A}\mathbf{r}^{A^*P} - \left[\mathbf{B}^* + \left({}^{F}_{B}\mathbf{R}\right){}^{B}\mathbf{r}^{B^*P'}\right] = 0 \tag{2.85}$$

However, the unknowns in the time-stepping formulation are not position they are velocity. Therefore, the constraints must be enforced at the velocity level. For 2D revolute joints, the linear velocity is constrained to be zero by two scalar equations, while the angular velocity is unconstrained. We know from equation (2.19) the linear velocity of a point attached to a rigid body. Setting this equation equal to zero will satisfy the 2D revolute joint constraint, however, we will set up the joint constraint more generally in a framework that allows for other joint types, and for this revolute joint we will recover equation (2.19).

The constraint Jacobian is most easily specified in the joint’s frame, however the velocity twist of the body is of the body’s center of gravity in the fixed world frame. Therefore, to constrain the joint, we must first align the body frame with the world frame[7], followed by a coordinate transform of the resulting velocity twist

[7] It is important to realize that ${}^{A}\boldsymbol{\nu}^{A}$ is not 0. Another way of thinking of this velocity is as ${}^{A'}\boldsymbol{\nu}^{A}$, where A' is located at the origin of the fixed frame, but instantaneously aligned with the

to the joint frame. With the velocity twist in the correct frame, we can apply the constraint Jacobian. Mathematically, the constraint can be written as:

$$\begin{aligned} 0 = \left[{}^{\alpha}\mathbf{J}^{A}\right]\left[{}^{\alpha}_{A}\mathbf{Ad}\right]{}^{A}\boldsymbol{\nu}^{A} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \left({}^{\alpha}_{A}\mathbf{R}\right) & \left(\left({}^{\alpha}_{A}\mathbf{R}\right){}^{A}\mathbf{r}^{A^{*}O}\right)^{\wedge} \\ 0_{1\times 2} & 1 \end{bmatrix} \begin{bmatrix} \left({}^{A}_{F}\mathbf{R}\right) & 0 \\ 0_{1\times 2} & 1 \end{bmatrix} \boldsymbol{\nu}^{A} \\ &= \begin{bmatrix} \mathbf{I} & \left(\left({}^{\alpha}_{A}\mathbf{R}\right){}^{A}\mathbf{r}^{A^{*}O}\right)^{\wedge} \end{bmatrix} \boldsymbol{\nu}^{A} \end{aligned} \tag{2.86}$$

where the matrix $\left[{}^{\alpha}\mathbf{J}^{A}\right]$ is the constraint Jacobian for a 2D revolute joint specified at the joint frame. As stated earlier, the constraint equation reduced to the equation for the linear velocity of a point attached to the rigid body.

The second bilateral constraint (the revolute joint between links A and B) can also be written in matrix form as:

$$\left[{}^{\beta}\mathbf{J}^{A}\right]\left[{}^{\beta}_{A}\mathbf{Ad}\right]{}^{A}\boldsymbol{\nu}^{A} + \left[{}^{\beta}\mathbf{J}^{B}\right]\left[{}^{\beta}_{B}\mathbf{Ad}\right]{}^{B}\boldsymbol{\nu}^{B} = 0 \tag{2.87}$$

Expanding equation (2.87):

$$\begin{aligned} 0 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \left({}^{\beta}_{A}\mathbf{R}\right) & \left(\left({}^{\beta}_{A}\mathbf{R}\right){}^{A}\mathbf{r}^{A^{*}P}\right)^{\wedge} \\ 0_{1\times 2} & 1 \end{bmatrix} \begin{bmatrix} \left({}^{A}_{F}\mathbf{R}\right) & 0 \\ 0_{1\times 2} & 1 \end{bmatrix} \boldsymbol{\nu}^{A} \\ &+ \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \left({}^{\beta}_{B}\mathbf{R}\right) & \left(\left({}^{\beta}_{B}\mathbf{R}\right)\mathbf{r}^{B^{*}P'}\right)^{\wedge} \\ 0_{1\times 2} & 1 \end{bmatrix} \begin{bmatrix} \left({}^{B}_{F}\mathbf{R}\right) & 0 \\ 0_{1\times 2} & 1 \end{bmatrix} \boldsymbol{\nu}^{B} \\ &= \begin{bmatrix} \mathbf{I} & \left(\left({}^{\beta}_{A}\mathbf{R}\right){}^{A}\mathbf{r}^{A^{*}P}\right)^{\wedge} \end{bmatrix} \boldsymbol{\nu}^{A} + \begin{bmatrix} -\mathbf{I} & -\left(\left({}^{\beta}_{B}\mathbf{R}\right){}^{B}\mathbf{r}^{B^{*}P'}\right)^{\wedge} \end{bmatrix} \boldsymbol{\nu}^{B} \end{aligned} \tag{2.88}$$

Using our notation,

$$\mathbf{W}_{b1}^{\mathrm{T}} = \begin{bmatrix} \mathbf{I} & \left(\left({}^{\alpha}_{A}\mathbf{R}\right){}^{A}\mathbf{r}^{A^{*}O}\right)^{\wedge} \end{bmatrix} \qquad \mathbf{W}_{b2}^{\mathrm{T}} = \begin{bmatrix} \mathbf{I} & \left(\left({}^{\beta}_{A}\mathbf{R}\right){}^{A}\mathbf{r}^{A^{*}P}\right)^{\wedge} \end{bmatrix} \tag{2.89}$$

$$\mathbf{W}_{b3}^{\mathrm{T}} = \begin{bmatrix} -\mathbf{I} & -\left(\left({}^{\beta}_{B}\mathbf{R}\right){}^{B}\mathbf{r}^{B^{*}P'}\right)^{\wedge} \end{bmatrix} \tag{2.90}$$

moving body frame A. This is why the transformation of ${}^{F}\boldsymbol{\nu}^{A}$ to ${}^{A}\boldsymbol{\nu}^{A}$ has 0 for the position vector in the top right block of the transform matrix (defined in equation (2.26)).

Combining the 3 constraints into a single matrix $\mathbf{W}_{\mathrm{b}}^{\mathrm{T}}$ produces the constraint wrench for the system:

$$\mathbf{W}_{\mathrm{b}}^{\mathrm{T}} = \begin{bmatrix} \mathbf{W}_{b1}^{\mathrm{T}} & 0 \\ \mathbf{W}_{b2}^{\mathrm{T}} & \mathbf{W}_{b3}^{\mathrm{T}} \end{bmatrix} \tag{2.91}$$

We can now formulate the mixed complementarity problem:

$$\begin{bmatrix} \mathbf{0}_{6\times 1} \\ \mathbf{0}_{4\times 1} \end{bmatrix} = \begin{bmatrix} -\mathbf{M} & \mathbf{W}_{\mathrm{b}} \\ \mathbf{W}_{\mathrm{b}}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}^{\ell+1} \\ \mathbf{p}_{\mathrm{b}}^{\ell+1} \end{bmatrix} + \begin{bmatrix} \mathbf{M}\boldsymbol{\nu}^{\ell} + h\boldsymbol{\lambda}_{\mathrm{app}} \\ \frac{1}{h}\boldsymbol{\phi}^{\ell} \end{bmatrix} \tag{2.92}$$

where $\frac{1}{h}\boldsymbol{\phi}^{\ell}$ is the constraint stabilization term for the bilateral constraints and $\boldsymbol{\phi}^{\ell}$ is obtained from equations (2.84) and (2.85):

$$\begin{bmatrix} \mathbf{A}^{*} + \left({}^{F}_{A}\mathbf{R}\right) {}^{A}\mathbf{r}^{A^{*}O} \\ \mathbf{A}^{*} + \left({}^{F}_{A}\mathbf{R}\right) {}^{A}\mathbf{r}^{A^{*}P} - \left[\mathbf{B}^{*} + \left({}^{F}_{B}\mathbf{R}\right) {}^{B}\mathbf{r}^{B^{*}P'}\right] \end{bmatrix} \tag{2.93}$$

2.7.2 Planar 2 Bar Pendulum Attached to Block

This system consists of a planar simple double pendulum attached to a block with all joint axes parallel to the Z-axis. The pendulum is constructed of two slender rods A and B with masses m_A and m_B respectively. Link A has length L_A and is connected to the block at point $\mathbf{O}$ by point $\mathbf{O}'$. Link B has length L_B and is connected to link A at point $\mathbf{P}$ by point $\mathbf{P}'$. Initially, the block is at rest on a horizontal surface. Frames are attached to each body and are fixed in that body. The position and orientation of the frames are used as the generalized coordinates. Figure 2.9 illustrates the system.

The configuration $\mathbf{q}$ of the system is $\mathbf{q} = [\mathbf{q}^{C}\ \mathbf{q}^{A}\ \mathbf{q}^{B}]$ and similarly the velocity twist of the system is $\boldsymbol{\nu} = [\boldsymbol{\nu}^{C}\ \boldsymbol{\nu}^{A}\ \boldsymbol{\nu}^{B}]$.

The bilateral constraint between links A and B of the pendulum is identical to the previous example. The constraint between the pendulum and the block at point $\mathbf{O}$ must now be handled, it can be written in matrix form as:

$$\begin{bmatrix} \mathbf{I} & \left(\left({}^{\alpha}_{C}\mathbf{R}\right) {}^{C}\mathbf{r}^{C^{*}O}\right)^{\wedge} \end{bmatrix} \boldsymbol{\nu}^{C} + \begin{bmatrix} -\mathbf{I} & -\left(\left({}^{\alpha}_{A}\mathbf{R}\right) {}^{A}\mathbf{r}^{A^{*}O'}\right)^{\wedge} \end{bmatrix} \boldsymbol{\nu}^{A} \tag{2.94}$$

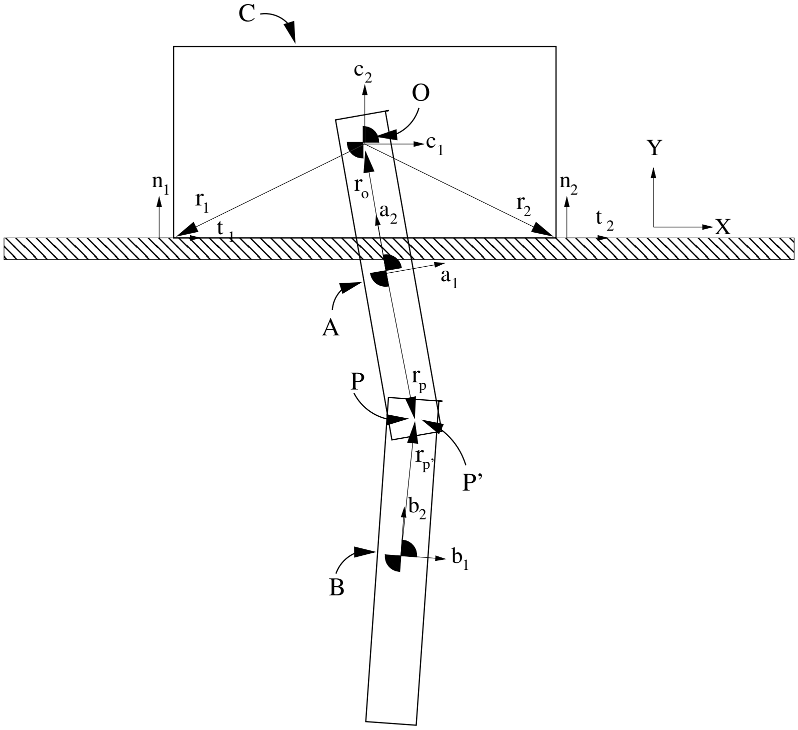


Figure 2.9: A planar simple double pendulum with all joint axes parallel to the Z-axis attached to a block on a surface.

Partitioning into the bilateral constraint wrenches:

$$\mathbf{W}_{b1}^{\mathrm{T}} = \begin{bmatrix} \mathbf{I} & \left(\left({}^{\alpha}_{C}\mathbf{R}\right){}^{C}\mathbf{r}^{C^{*}O}\right)^{\wedge} \end{bmatrix} \qquad \mathbf{W}_{b2}^{\mathrm{T}} = \begin{bmatrix} -\mathbf{I} & -\left(\left({}^{\alpha}_{A}\mathbf{R}\right){}^{A}\mathbf{r}^{A^{*}O'}\right)^{\wedge} \end{bmatrix} \tag{2.95}$$

$$\mathbf{W}_{b3}^{\mathrm{T}} = \begin{bmatrix} \mathbf{I} & \left(\left({}^{\beta}_{A}\mathbf{R}\right){}^{A}\mathbf{r}^{A^{*}P}\right)^{\wedge} \end{bmatrix} \qquad \mathbf{W}_{b4}^{\mathrm{T}} = \begin{bmatrix} -\mathbf{I} & -\left(\left({}^{\beta}_{B}\mathbf{R}\right){}^{B}\mathbf{r}^{B^{*}P'}\right)^{\wedge} \end{bmatrix} \tag{2.96}$$

The nonpenetration constraints between the block and floor are written as:

$$\psi_{1\mathrm{n}} = \left({}^{n}_{c}\mathbf{R}\right){}^{c}\mathbf{r}_{1} \geq 0 \tag{2.97}$$

$$\psi_{2\mathrm{n}} = \left({}^{n}_{c}\mathbf{R}\right){}^{c}\mathbf{r}_{2} \geq 0 \tag{2.98}$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are shown in figure 2.9. Vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ are constant in the body fixed frame $\{C\}$, and must be transformed into the inertial frame $\{F\}$ to determine the gap above the floor. Each gap function has a corresponding multiplier λ_{in} which is the contact force between contact i and the floor.

2.7.2.1 System Dynamics

Assuming the two contacts are included in the active set every time step, the MCP is size 15 and the various quantities appearing in it are:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_c & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \mathbf{M}_a & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \mathbf{M}_b \end{bmatrix} \tag{2.99}$$

where $\mathbf{M}_c = \begin{bmatrix} m_c & 0 & 0 \\ 0 & m_c & 0 \\ 0 & 0 & I_c \end{bmatrix}$, $\mathbf{M}_a = \begin{bmatrix} m_a & 0 & 0 \\ 0 & m_a & 0 \\ 0 & 0 & I_a \end{bmatrix}$, and $\mathbf{M}_b = \begin{bmatrix} m_b & 0 & 0 \\ 0 & m_b & 0 \\ 0 & 0 & I_b \end{bmatrix}$. The scalar $I_c = \frac{1}{12}m_c(l_c^2 + w_c^2)$ is the moment of inertia for a 2D block and the scalars $I_a = \frac{1}{12}m_a l_a^2$ and $I_b = \frac{1}{12}m_b l_b^2$ are the respective moments of inertia for rod's A and B.

The system constraint wrenches are:

$$\mathbf{W}_{\mathrm{b}}^{\mathrm{T}} = \begin{bmatrix} \mathbf{W}_{b1}^{\mathrm{T}} & \vdots & \mathbf{W}_{b2}^{\mathrm{T}} & \vdots & 0_{2\times3} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0_{2\times3} & \vdots & \mathbf{W}_{b3}^{\mathrm{T}} & \vdots & \mathbf{W}_{b4}^{\mathrm{T}} \end{bmatrix} \qquad \mathbf{W}_{\mathrm{n}} = \begin{bmatrix} \hat{\mathbf{n}}_1 & \vdots & \hat{\mathbf{n}}_2 \\ \left({}^F_C\mathbf{R}\right) {}^C\mathbf{r}_1 \otimes \hat{\mathbf{n}}_1 & \vdots & \left({}^F_C\mathbf{R}\right) {}^c\mathbf{r}_2 \otimes \hat{\mathbf{n}}_2 \\ \cdots & \cdots & \cdots \\ 0_{3\times1} & \vdots & 0_{3\times1} \\ \cdots & \cdots & \cdots \\ 0_{3\times1} & \vdots & 0_{3\times1} \end{bmatrix} \tag{2.100}$$

For the terms appearing in the MCP's $\mathbf{b}$ vector:

$$\boldsymbol{\lambda}_{\mathrm{app}} = \begin{bmatrix} 0 & -m_c g & 0 & 0 & -m_a g & 0 & 0 & -m_b g & 0 \end{bmatrix}^{\mathrm{T}} \tag{2.101}$$

where g is the gravitational acceleration constant.

The bilateral constraint stabilization terms are again obtained from the position level constraint functions:

$$\boldsymbol{\phi}^{\ell} = \begin{bmatrix} \mathbf{C}^{*} + \left({}^{F}_{C}\mathbf{R}\right) {}^{C}\mathbf{r}^{C^{*}O} - \left(\mathbf{A}^{*} + \left({}^{F}_{A}\mathbf{R}\right) {}^{A}\mathbf{r}^{A^{*}O'}\right) \\ \mathbf{A}^{*} + \left({}^{F}_{A}\mathbf{R}\right) {}^{A}\mathbf{r}^{A^{*}P} - \left(\mathbf{B}^{*} + \left({}^{F}_{B}\mathbf{R}\right) {}^{B}\mathbf{r}^{B^{*}P'}\right) \end{bmatrix} \tag{2.102}$$

We now also have unilateral constraint terms, and these are obtained from equations (2.97) and (2.98):

$$\boldsymbol{\psi}^{\ell}_{\mathrm{n}} = \begin{bmatrix} \left({}^{n}_{c}\mathbf{R}\right) {}^{c}\mathbf{r}_{1} \\ \left({}^{n}_{c}\mathbf{R}\right) {}^{c}\mathbf{r}_{2} \end{bmatrix} \tag{2.103}$$

Putting it all together, we can now formulate the mixed complementarity problem for this example:

$$\begin{bmatrix} 0_{9\times1} \\ 0_{4\times1} \\ \boldsymbol{\rho}^{\ell+1}_{\mathrm{n}} \end{bmatrix} = \begin{bmatrix} -\mathbf{M} & \mathbf{W}_{\mathrm{b}} & \mathbf{W}_{\mathrm{n}} \\ \mathbf{W}_{\mathrm{b}}^{\mathrm{T}} & 0 & 0 \\ \mathbf{W}_{\mathrm{n}}^{\mathrm{T}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}^{\ell+1} \\ \mathbf{p}^{\ell+1}_{\mathrm{b}} \\ \mathbf{p}^{\ell+1}_{\mathrm{n}} \end{bmatrix} + \begin{bmatrix} \mathbf{M}\boldsymbol{\nu}^{\ell} + h\boldsymbol{\lambda}^{\ell}_{\mathrm{app}} \\ \frac{1}{h}\boldsymbol{\phi}^{\ell} \\ \frac{1}{h}\boldsymbol{\psi}_{\mathrm{n}}{}^{\ell} \end{bmatrix} \tag{2.104}$$

$$0 \leq \boldsymbol{\rho}^{\ell+1}_{\mathrm{n}} \perp \mathbf{p}^{\ell+1}_{\mathrm{n}} \geq 0 \tag{2.105}$$

2.7.3 Planar 2 Bar Pendulum Attached to Block with Friction

This problem is identical to the previous "Planar 2 Bar Pendulum Attached to Block" example, with the addition of a friction force between the block and surface. We can skip right to the system dynamics.

2.7.3.1 System Dynamics

Assuming the two contacts are included in the active set every time step, the MCP is size 21 and the following quantities appearing in it are:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{c} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \mathbf{M}_{a} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \mathbf{M}_{b} \end{bmatrix} \qquad \mathbf{W}_{\mathrm{b}}^{\mathrm{T}} = \begin{bmatrix} \mathbf{W}_{b1}^{\mathrm{T}} & \vdots & \mathbf{W}_{b2}^{\mathrm{T}} & \vdots & 0_{2\times3} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0_{2\times3} & \vdots & \mathbf{W}_{b3}^{\mathrm{T}} & \vdots & \mathbf{W}_{b4}^{\mathrm{T}} \end{bmatrix} \tag{2.106}$$

$$
\mathbf{W}_{\text{n}} = \left[\begin{array}{ccc}
\hat{\mathbf{n}}_1 & \vdots & \hat{\mathbf{n}}_2 \\
\left({}^{F}_{C}\mathbf{R}\right) {}^{C}\mathbf{r}_1 \otimes \hat{\mathbf{n}}_1 & \vdots & \left({}^{F}_{C}\mathbf{R}\right) {}^{C}\mathbf{r}_2 \otimes \hat{\mathbf{n}}_2 \\
\hdashline
0_{3\times 1} & \vdots & 0_{3\times 1} \\
\hdashline
0_{3\times 1} & \vdots & 0_{3\times 1}
\end{array}\right]
\qquad
\boldsymbol{\lambda}_{\text{app}} = \begin{bmatrix} 0 \\ -m_c g \\ 0 \\ 0 \\ -m_a g \\ 0 \\ 0 \\ -m_b g \\ 0 \end{bmatrix}
\tag{2.107}
$$

The new frictional quantities to appear are:

$$
\mathbf{W}_{\text{f}} = \left[\begin{array}{ccccc}
\hat{\mathbf{t}}_1 & -\hat{\mathbf{t}}_1 & \vdots & \hat{\mathbf{t}}_2 & -\hat{\mathbf{t}}_2 \\
\left({}^{F}_{C}\mathbf{R}\right) {}^{C}\mathbf{r}_1 \otimes \hat{\mathbf{t}}_1 & \left({}^{F}_{C}\mathbf{R}\right) {}^{C}\mathbf{r}_1 \otimes -\hat{\mathbf{t}}_1 & \vdots & \left({}^{F}_{C}\mathbf{R}\right) {}^{C}\mathbf{r}_2 \otimes \hat{\mathbf{t}}_2 & \left({}^{F}_{C}\mathbf{R}\right) {}^{C}\mathbf{r}_2 \otimes -\hat{\mathbf{t}}_2 \\
\hdashline
0_{3\times 1} & 0_{3\times 1} & \vdots & 0_{3\times 1} & 0_{3\times 1} \\
\hdashline
0_{3\times 1} & 0_{3\times 1} & \vdots & 0_{3\times 1} & 0_{3\times 1}
\end{array}\right]
\tag{2.108}
$$

$$
\mathbf{E} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}
\qquad
\mathbf{U} = \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix}
\tag{2.109}
$$

where μ_i is the coefficient of friction at contact i.

Putting it all together, we can now formulate the mixed complementarity problem:

$$\begin{bmatrix} 0_{9\times 1} \\ 0_{4\times 1} \\ \boldsymbol{\rho}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\rho}_{\mathrm{f}}^{\ell+1} \\ \mathbf{s}^{\ell+1} \end{bmatrix} = \begin{bmatrix} -\mathbf{M} & \mathbf{W}_{\mathrm{b}} & \mathbf{W}_{\mathrm{n}} & \mathbf{W}_{\mathrm{f}} & 0 \\ \mathbf{W}_{\mathrm{b}}^{\mathrm{T}} & 0 & 0 & 0 & 0 \\ \mathbf{W}_{\mathrm{n}}^{\mathrm{T}} & 0 & 0 & 0 & 0 \\ \mathbf{W}_{\mathrm{f}}^{\mathrm{T}} & 0 & 0 & 0 & \mathbf{E} \\ 0 & 0 & \mathbf{U} & -\mathbf{E}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}^{\ell+1} \\ \mathbf{p}_{\mathrm{b}}^{\ell+1} \\ \mathbf{p}_{\mathrm{n}}^{\ell+1} \\ \mathbf{p}_{\mathrm{f}}^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix} + \begin{bmatrix} \mathbf{M}\boldsymbol{\nu}^{\ell} + h\boldsymbol{\lambda}_{\mathrm{app}} \\ \frac{1}{h}\boldsymbol{\phi}^{\ell} \\ \frac{1}{h}\boldsymbol{\psi}_{\mathrm{n}}^{\ell} \\ 0_{4\times 1} \\ 0_{2\times 1} \end{bmatrix} \tag{2.110}$$

$$0 \leq \begin{bmatrix} \boldsymbol{\rho}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\rho}_{\mathrm{f}}^{\ell+1} \\ \mathbf{s}^{\ell+1} \end{bmatrix} \perp \begin{bmatrix} \mathbf{p}_{\mathrm{n}}^{\ell+1} \\ \mathbf{p}_{\mathrm{f}}^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix} \geq 0 \tag{2.111}$$

2.7.4 Planar 2 Bar Pendulum Attached to Block with Friction and Position Controlled Body

We extend the previous example with the introduction of a position controlled body manipulating the block. For simplification, we assume the pusher is a particle and interacts with the block through a single frictional point contact. The configuration of this system is unchanged as the new body is not force controlled. Figure 2.10 illustrates the problem.

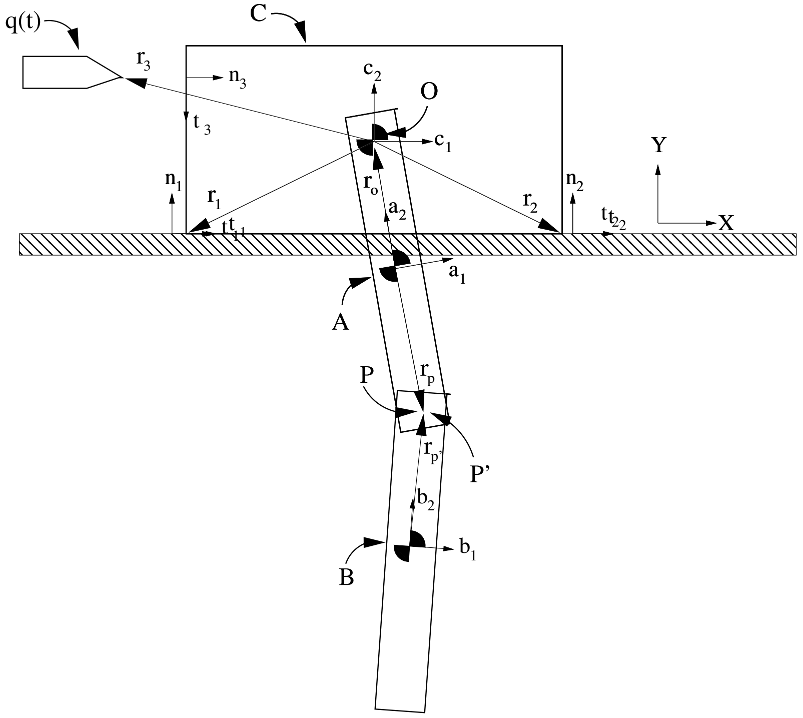


Figure 2.10: A planar simple double pendulum with all joint axes parallel to the Z-axis attached to a block on a surface. In addition, there is a position controlled body manipulating the block.

2.7.4.1 System Dynamics

The following quantities from before still appear:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_c & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \mathbf{M}_a & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \mathbf{M}_b \end{bmatrix} \quad \mathbf{W}_{\mathrm{b}}^{\mathrm{T}} = \begin{bmatrix} \mathbf{W}_{b1}^{\mathrm{T}} & \vdots & \mathbf{W}_{b2}^{\mathrm{T}} & \vdots & 0_{2\times3} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0_{2\times3} & \vdots & \mathbf{W}_{b3}^{\mathrm{T}} & \vdots & \mathbf{W}_{b4}^{\mathrm{T}} \end{bmatrix} \quad \boldsymbol{\lambda}_{\mathrm{app}} = \begin{bmatrix} 0 \\ -m_c g \\ 0 \\ 0 \\ -m_a g \\ 0 \\ 0 \\ -m_b g \\ 0 \end{bmatrix} \tag{2.112}$$

Assuming the 3 contacts are included in the active set every time step, the

MCP is size 25, and the other quantities are:

$$\mathbf{W}_{\mathrm{n}} = \left[\begin{array}{ccccc} \hat{\mathbf{n}}_1 & \vdots & \hat{\mathbf{n}}_2 & \vdots & {}^{C}\hat{\mathbf{n}}_3 \\ \left({}^{F}_{C}\mathbf{R}\right) {}^{C}\mathbf{r}_1 \otimes \hat{\mathbf{n}}_1 & \vdots & \left({}^{F}_{C}\mathbf{R}\right) {}^{C}\mathbf{r}_2 \otimes \hat{\mathbf{n}}_2 & \vdots & \left({}^{C}_{F}\mathbf{R}\right) {}^{F}\mathbf{r}_3 \otimes {}^{c}\hat{\mathbf{n}}_3 \\ \hdashline 0_{3\times1} & \vdots & 0_{3\times1} & \vdots & 0_{3\times1} \\ \hdashline 0_{3\times1} & \vdots & 0_{3\times1} & \vdots & 0_{3\times1} \end{array}\right] \tag{2.113}$$

The friction constraint wrench (for space constraints the rotation matrices performing the frame transforms have been dropped):

$$\mathbf{W}_{\mathrm{f}} = \left[\begin{array}{cccccccc} \hat{\mathbf{t}}_1 & -\hat{\mathbf{t}}_1 & \vdots & \hat{\mathbf{t}}_2 & -\hat{\mathbf{t}}_2 & \vdots & {}^{C}\hat{\mathbf{t}}_3 & -{}^{C}\hat{\mathbf{t}}_3 \\ \mathbf{r}_1 \otimes \hat{\mathbf{t}}_1 & \mathbf{r}_1 \otimes -\hat{\mathbf{t}}_1 & \vdots & \mathbf{r}_2 \otimes \hat{\mathbf{t}}_2 & \mathbf{r}_2 \otimes -\hat{\mathbf{t}}_2 & \vdots & {}^{C}\mathbf{r}_3 \otimes {}^{C}\hat{\mathbf{t}}_3 & {}^{C}\mathbf{r}_3 \otimes -{}^{C}\hat{\mathbf{t}}_3 \\ \hdashline 0_{3\times1} & 0_{3\times1} & \vdots & 0_{3\times1} & 0_{3\times1} & \vdots & 0_{3\times1} & 0_{3\times1} \\ \hdashline 0_{2\times1} & 0_{2\times1} & \vdots & 0_{2\times1} & 0_{2\times1} & \vdots & 0_{2\times1} & 0_{2\times1} \end{array}\right] \tag{2.114}$$

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{U} = \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{bmatrix} \tag{2.115}$$

Since the position of the pusher is a time-dependent function $f(t)$, we must remember to include the partial derivative of that function in the $\mathbf{b}$ vector.

Putting it all together, we can now formulate the mixed complementarity problem:

$$\begin{bmatrix} 0_{9\times1} \\ 0_{4\times1} \\ \boldsymbol{\rho}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\rho}_{\mathrm{f}}^{\ell+1} \\ \mathbf{s}^{\ell+1} \end{bmatrix} = \begin{bmatrix} -\mathbf{M} & \mathbf{W}_{\mathrm{b}} & \mathbf{W}_{\mathrm{n}} & \mathbf{W}_{\mathrm{f}} & 0 \\ \mathbf{W}_{\mathrm{b}}^{\mathrm{T}} & 0 & 0 & 0 & 0 \\ \mathbf{W}_{\mathrm{n}}^{\mathrm{T}} & 0 & 0 & 0 & 0 \\ \mathbf{W}_{\mathrm{f}}^{\mathrm{T}} & 0 & 0 & 0 & \mathbf{E} \\ 0 & 0 & \mathbf{U} & -\mathbf{E}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}^{\ell+1} \\ \mathbf{p}_{\mathrm{b}}^{\ell+1} \\ \mathbf{p}_{\mathrm{n}}^{\ell+1} \\ \mathbf{p}_{\mathrm{f}}^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix} + \begin{bmatrix} \mathbf{M}\boldsymbol{\nu}^{\ell} + h\boldsymbol{\lambda}_{\mathrm{app}} \\ \frac{1}{h}\boldsymbol{\phi}^{\ell} \\ \frac{1}{h}\boldsymbol{\psi}_{\mathrm{n}}^{\ell} + \dfrac{\partial \boldsymbol{\psi}_{\mathrm{n}}^{\ell}}{\partial t} \\ 0_{6\times1} \\ 0_{3\times1} \end{bmatrix} \tag{2.116}$$

$$0 \leq \begin{bmatrix} \boldsymbol{\rho}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\rho}_{\mathrm{f}}^{\ell+1} \\ \mathbf{s}^{\ell+1} \end{bmatrix} \perp \begin{bmatrix} \mathbf{p}_{\mathrm{n}}^{\ell+1} \\ \mathbf{p}_{\mathrm{f}}^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix} \geq 0 \tag{2.117}$$

2.7.5 Planar 2 Bar Pendulum Attached to Block with Friction, Position Controlled Body, and Moving Floor

This example is identical to the previous example with the simple addition of $\dfrac{\partial \boldsymbol{\Psi}_{\mathrm{f}}}{\partial t}$ in the $\mathbf{b}$ vector. This term represents the lateral position change of the frictional surface in one time step, i.e., people movers in airports.

$$\mathbf{b} = \begin{bmatrix} \mathbf{M}\boldsymbol{\nu}^{\ell} + h\boldsymbol{\lambda}_{\mathrm{app}} \\ \frac{1}{h}\boldsymbol{\phi}^{\ell} \\ \frac{1}{h}\boldsymbol{\psi}_{\mathrm{n}}^{\ell} + \dfrac{\partial \boldsymbol{\psi}_{\mathrm{n}}^{\ell}}{\partial t} \\ \dfrac{\partial \boldsymbol{\Psi}_{\mathrm{f}}^{\ell}}{\partial t} \\ 0_{3\times1} \end{bmatrix} \tag{2.118}$$

2.7.6 Full Planar Model

We extend the previous example by replacing the bottom rod of the pendulum with a spring and particle mass. Figure 2.11 illustrates the problem. For this example, vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ are constant in the blocks body frame C and vector $\mathbf{r}_3$ is most naturally represented in the fixed inertial frame. Similarly, vectors $\hat{\mathbf{n}}_1$, $\hat{\mathbf{n}}_2$, $\hat{\mathbf{n}}_1$, and $\hat{\mathbf{n}}_2$ are constant in the fixed inertial frame, and vectors $\hat{\mathbf{n}}_3$ and $\hat{\mathbf{n}}_3$ are constant in the $\{C\}$ frame.

Assuming the 3 contacts are included in the active set every time step, the

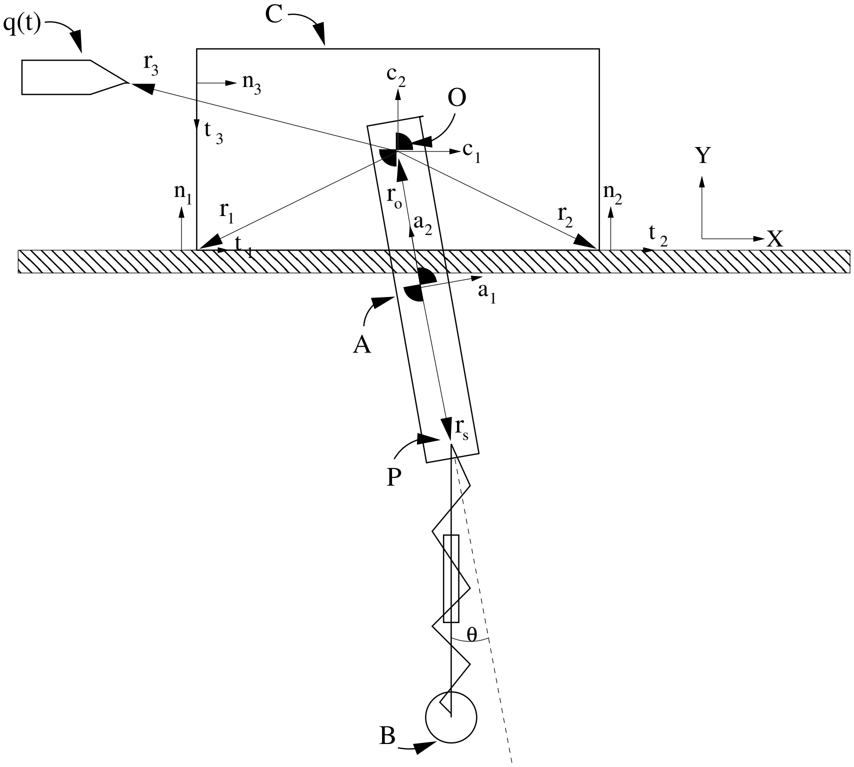


Figure 2.11: A force controlled block is sitting on a conveyor belt with a pendulum attached at its center of mass. At the other end of the pendulum, a spring and particle are attached. The block is interacted along its left edge with a position controlled pusher.

MCP is size 22, and the quantities appearing in the matrix are presented next.

The mass matrices of the three force controlled bodies are:

$$\mathbf{M}_c = \begin{bmatrix} m_c & 0 & 0 \\ 0 & m_c & 0 \\ 0 & 0 & I_c \end{bmatrix} \qquad \mathbf{M}_a = \begin{bmatrix} m_a & 0 & 0 \\ 0 & m_a & 0 \\ 0 & 0 & I_a \end{bmatrix} \qquad \mathbf{M}_b = \begin{bmatrix} m_b & 0 \\ 0 & m_b \end{bmatrix} \tag{2.119}$$

resulting in a system mass matrix of

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_c & 0_{3\times3} & 0_{3\times2} \\ 0_{3\times3} & \mathbf{M}_a & 0_{3\times2} \\ 0_{2\times3} & 0_{2\times3} & \mathbf{M}_b \end{bmatrix} \tag{2.120}$$

Next, we must deal with the single bilateral constraint of the system, the revolute joint at point **O**. From before, we know we must constrain the relative velocity at point **O** between bodies C and A to be zero.

$$\mathbf{W}_{b1}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{2\times 2} & 0 \\ 0_{1\times 2} & 1 \end{bmatrix} \quad \mathbf{W}_{b2}^{\mathrm{T}} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{2\times 2} & \left({}^{F}_{A}\mathbf{R}\right) {}^{A}\mathbf{r}_{o} \\ 0_{1\times 2} & 1 \end{bmatrix} \quad \mathbf{W}_{\mathrm{b}} = \begin{bmatrix} \mathbf{W}_{b1} \\ \mathbf{W}_{b2} \\ 0_{2\times 2} \end{bmatrix} \tag{2.121}$$

Now, the unilateral constraint wrench:

$$\mathbf{W}_{\mathrm{n}} = \left[\begin{array}{ccccc} \hat{\mathbf{n}}_1 & \vdots & \hat{\mathbf{n}}_2 & \vdots & {}^{C}\hat{\mathbf{n}}_3 \\ \left({}^{F}_{C}\mathbf{R}\right) {}^{C}\mathbf{r}_1 \otimes \hat{\mathbf{n}}_1 & \vdots & \left({}^{F}_{C}\mathbf{R}\right) {}^{C}\mathbf{r}_2 \otimes \hat{\mathbf{n}}_2 & \vdots & \left({}^{C}_{F}\mathbf{R}\right) {}^{F}\mathbf{r}_3 \otimes {}^{C}\hat{\mathbf{n}}_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0_{3\times 1} & \vdots & 0_{3\times 1} & \vdots & 0_{3\times 1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0_{2\times 1} & \vdots & 0_{2\times 1} & \vdots & 0_{2\times 1} \end{array} \right] \tag{2.122}$$

The friction constraint wrench (for space constraints the rotation matrices performing the frame transforms have been dropped):

$$\mathbf{W}_{\mathrm{f}} = \left[\begin{array}{cccccccc} \hat{\mathbf{t}}_1 & -\hat{\mathbf{t}}_1 & \vdots & \hat{\mathbf{t}}_2 & -\hat{\mathbf{t}}_2 & \vdots & {}^{C}\hat{\mathbf{t}}_3 & -{}^{C}\hat{\mathbf{t}}_3 \\ \mathbf{r}_1 \otimes \hat{\mathbf{t}}_1 & \mathbf{r}_1 \otimes -\hat{\mathbf{t}}_1 & \vdots & \mathbf{r}_2 \otimes \hat{\mathbf{t}}_2 & \mathbf{r}_2 \otimes -\hat{\mathbf{t}}_2 & \vdots & {}^{C}\mathbf{r}_3 \otimes {}^{C}\hat{\mathbf{t}}_3 & {}^{C}\mathbf{r}_3 \otimes -{}^{C}\hat{\mathbf{t}}_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0_{3\times 1} & 0_{3\times 1} & \vdots & 0_{3\times 1} & 0_{3\times 1} & \vdots & 0_{3\times 1} & 0_{3\times 1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0_{2\times 1} & 0_{2\times 1} & \vdots & 0_{2\times 1} & 0_{2\times 1} & \vdots & 0_{2\times 1} & 0_{2\times 1} \end{array} \right] \tag{2.123}$$

The other matrices required for Coulomb's friction:

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{U} = \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{bmatrix} \tag{2.124}$$

In this example, gravity is not the only external force acting on the system, we also have the spring force and damping force acting on bodies A and B.

For the spring, $\sin(\theta)$ and $\cos(\theta)$ are functions of rod A's endpoint (point $\mathbf{P}$) and the particle B's position.

$$L = \sqrt{(B_x - P_x)^2 + (B_y - P_y)^2} \tag{2.125}$$

$$\sin(\theta) = -(B_x - P_x)/L \tag{2.126}$$

$$\cos(\theta) = -(P_y - B_y)/L \tag{2.127}$$

where L is the length of the spring. This allows us to construct the rotation matrix from the springs frame $\{S\}$ into the fixed world frame:

$$\left({}^F_S\mathbf{R}\right) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{2.128}$$

The force produced by the spring acts along the spring's y-axis and is equal to $-K(L - L_0)$, where L_0 is the rest length of the spring and K is the spring constant. This allows us to write down the spring force in the springs frame as: ${}^S\mathbf{f}_s$ $= [0 \ \ -K(L - L_0)]^{\mathrm{T}}$. To compute the force of the spring on the particle, a simple change of frame is all that is required: $\mathbf{f}_s = \left({}^F_S\mathbf{R}\right) {}^S\mathbf{f}_s$.

For the pendulum, slightly more work is needed. We first compute the spring force acting at point $\mathbf{P}$, which we know must be opposite and equal the force acting on the particle B, $-\mathbf{f}_s$. Now that we know the force acting at point $\mathbf{P}$, we need to compute the wrench acting at $\mathbf{A}^*$. Similar to contact forces, the wrench associated with the spring $\mathbf{W}_s$ for body A is:

$$\mathbf{W}_s = \begin{bmatrix} -\mathbf{f}_s \\ \left({}^F_A\mathbf{R}\right) {}^A\mathbf{r}_s \otimes -\mathbf{f}_s \end{bmatrix} \tag{2.129}$$

For the damping force, we need to first compute the relative velocity between the particle B and the point $\mathbf{P}$. The velocity of point $\mathbf{P}$ can be obtained from the velocity twist of the pendulum:

$$\mathbf{v}^p = \mathbf{v}^A + \omega^A \left(\mathbf{r}_p\right)^\wedge \tag{2.130}$$

The damping force acting on the particle B therefore becomes:

$$\mathbf{f}_d = -C(\mathbf{v}^b - \mathbf{v}^p) \tag{2.131}$$

and the force acting at point $\mathbf{P}$ is $-\mathbf{f}_d$ (opposite and equal). For the body A, we again have to transform the damping force acting at point P into the corresponding wrench acting at A^*.

$$\mathbf{W}_d = \begin{bmatrix} -\mathbf{f}_d \\ \mathbf{r}_s \otimes -\mathbf{f}_d \end{bmatrix} \tag{2.132}$$

We group all the external forces/wrenches into the applied wrench, $\boldsymbol{\lambda}_{\text{app}}$:

$$\boldsymbol{\lambda}_{\text{app}} = \begin{bmatrix} \mathbf{g}_c \\ \mathbf{g}_a + \mathbf{W}_s + \mathbf{W}_d \\ \mathbf{g}_b + \mathbf{f}_s + \mathbf{f}_d \end{bmatrix} \tag{2.133}$$

The other elements of $\mathbf{b}$ are the constraint stabilization terms and the partial derivative appearing from the position controlled body. Putting it all together, we can formulate the mixed linear complementarity problem:

$$\begin{bmatrix} 0_{8\times 1} \\ 0_{2\times 1} \\ \boldsymbol{\rho}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\rho}_{\mathrm{f}}^{\ell+1} \\ \mathbf{s}^{\ell+1} \end{bmatrix} = \begin{bmatrix} -\mathbf{M} & \mathbf{W}_{\mathrm{b}} & \mathbf{W}_{\mathrm{n}} & \mathbf{W}_{\mathrm{f}} & 0 \\ \mathbf{W}_{\mathrm{b}}^{\mathrm{T}} & 0 & 0 & 0 & 0 \\ \mathbf{W}_{\mathrm{n}}^{\mathrm{T}} & 0 & 0 & 0 & 0 \\ \mathbf{W}_{\mathrm{f}}^{\mathrm{T}} & 0 & 0 & 0 & \mathbf{E} \\ 0 & 0 & \mathbf{U} & -\mathbf{E}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}^{\ell+1} \\ \mathbf{p}_{\mathrm{b}}^{\ell+1} \\ \mathbf{p}_{\mathrm{n}}^{\ell+1} \\ \mathbf{p}_{\mathrm{f}}^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix} + \begin{bmatrix} \mathbf{M}\boldsymbol{\nu}^{\ell} + h(\boldsymbol{\lambda}_{\mathrm{app}}) \\ \frac{1}{h}\boldsymbol{\phi}^{\ell} \\ \frac{1}{h}\boldsymbol{\psi}_{\mathrm{n}} + \dfrac{\partial \boldsymbol{\psi}_{\mathrm{n}}}{\partial t} \\ \dfrac{\partial \boldsymbol{\Psi}_{\mathrm{f}}}{\partial t} \\ 0_{3x1} \end{bmatrix} \tag{2.134}$$

$$0 \leq \begin{bmatrix} \boldsymbol{\rho}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\rho}_{\mathrm{f}}^{\ell+1} \\ \mathbf{s}^{\ell+1} \end{bmatrix} \perp \begin{bmatrix} \mathbf{p}_{\mathrm{n}}^{\ell+1} \\ \mathbf{p}_{\mathrm{f}}^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix} \geq 0 \tag{2.135}$$

3. Alternative Time-Stepping Formulations

The time-stepping formulation presented in the previous chapter was based upon the work in [110]. In this chapter, we also build upon this work developing new alternative time-stepping formulations. These subsequent formulations were constructed for a variety of reasons, including accuracy, performance, and design. In section 3.2, a new method is presented that correctly handles non-penetration constraints for non-convex corners. In section 3.3, a 2.5D method is presented that adds surface friction to a 2D body. This allows for an approximation of surface friction effects, without the associated costs of modeling the entire 3D body. Section 3.4 presents a time-stepping formulation for multi-body quasi-static systems, which despite being narrower than the dynamic model, includes a large number of important tasks, such as low-speed assembly, static grasping, and walking using tripods of support. Lastly, section 3.5 presents a fully implicit time-stepping scheme, which we currently believe is the most accurate time-stepping formulation available.

These alternative time-stepping methods are not mutually exclusive. For example, in chapter 4 we present a simulation using a quasi-static 2.5D formulation. It would also be possible for one to incorporate the non-convex formulation results presented in section 3.2. However, not all combinations are possible. For example, the fully implicit time-stepper in section 3.5 requires the body geometry to be modeled as convex implicit functions, so the non-convex corner handling method in section 3.2 is not applicable.

3.1 Existing Alternative Complementarity Formulations of Dynamics

The time-stepping formulation presented in the previous chapter was based upon the work in [110], but this was not the first complementarity based formulation of dynamics. One of the first formulations of multi-rigid-body dynamics subject to unilateral constraints incorporating complementarity theory was done by Per Lötstedt [67, 68]. In these papers, Lötstedt used the instantaneous formulation of the dynamics and modeled the non-penetration constraint as a complementarity problem. He proved that in the frictionless case one will get a unique solution from the LCP. When he extended the formulation to incorporate Coulomb friction [69], he modified the Coulomb model to be a relationship between the normal force of the previous step and the friction forces of the current step. This was done to avoid the solution existence and uniqueness issues associated with frictional contact problems first noted by Painlevé [83]. In particular, Painlevé showed that there are systems where no solution to the frictional forces exist, and systems where if a frictional force is found, $\dot{\boldsymbol{\nu}}$ may not be unique. There has been a great deal of research on acceleration-level modeling of rigid bodies subject to Coulomb friction addressing the Painlevé problem, including [73, 13, 47, 10, 14, 37, 48, 89, 113, 86, 46]. Baraff [10] used the lack of solution guarantee to show and popularize that the acceleration based LCP is NP hard. Trinkle *et al.* [113] provided the first complementarity problem formulation for the three dimensional multi-rigid-body contact problem with Coulomb friction, and Pang and Trinkle [86] provided existence and uniqueness results under certain (limiting) assumptions.

A common resolution (though not without its own controversy, e.g. [25]) to solution inconsistency of acceleration based rigid-body models involves the use of time-stepping formulations based on impulses (integrals of the forces) with complementarity theory (*e.g.* [110, 106, 6]) or optimization conditions (*e.g.* [56, 74, 66, 2]) to resolve the contact state. In [109], Stewart provided the first convergence results and rigorously proved solutions exist with velocity level formulations, overcoming Painlevé-type problems. The seminal work of Stewart and Trinkle [110] has led to an active area of research in complementarity based rigid-body time-stepping, including: [5, 107, 97, 116, 115, 28, 4, 1, 64, 51, 39, 40, 92, 24, 60]. Anitescu and Potra [6] were able to prove solution existence with a minor modification to the formulation. Tzitzouris [116] was able to make a first order implicit time-stepping formulation, but required a distance function between bodies and two levels of derivatives be available in closed form. Anitescu [1] incorporated Newton's restitution law into a fixed step-size time-stepping framework. Potra *et al.* [92] replaced the first order Euler step with a higher order trapezoidal method, but required an adaptive step size and needed to determine the time of impacts.

Anitescu and Hart [3] showed that even in simple situations, the complementarity problems may have a nonconvex solution set, and conclude that in general velocity-based LCPs may be hard to solve. Time-stepping methods have also been extended to other models, including quasi-rigid [101, 88, 102], quasi-static [87], and elastic [11, 108, 54]. The remaining sections of this chapter present some of the new time-stepping models we have developed, nonconvex nonpenetration constraints [36], 2.5D [18], quasi-static [112], and geometrically implicit [24, 23].

3.2 Modeling Nonconvex NonPenetration Constraints

As mentioned in the introduction, a common source of error in simulation of multi-rigid-body systems caused by polyhedral models is the assumption that non-penetration constraints can be accurately represented as a conjunction of linear inequalities. This implicit assumption of local convexity introduces artifacts in the results of the simulation, as shown in figure 3.1.

The complementarity time-stepping formulation derived in section 2.6 suffers this problem. We are now going to look more carefully at the non-penetration constraint in the original time-stepper, and how we can modify it to allow for a non-convex constraint.

3.2.1 Non-Penetration Constraint

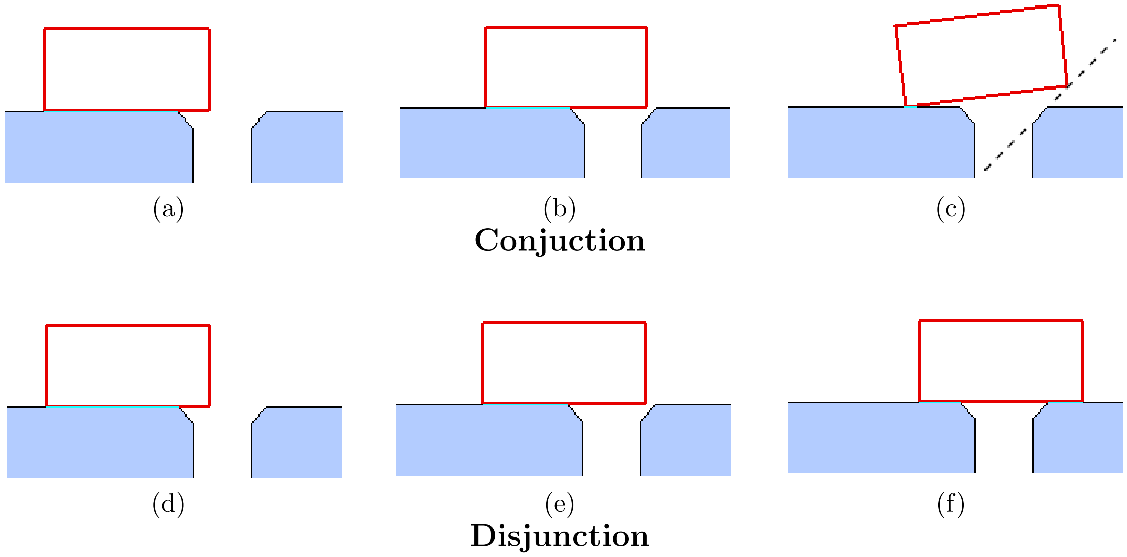
Given a point mass approaching a vertex with incident edges e_1 and e_2, the non-penetration constraint (see section 2.4.2 for more details) can be written as the following two complementarity conditions:

$$0 \leq \psi_{1\mathrm{n}} \perp p_{1\mathrm{n}} \geq 0 \tag{3.1}$$

$$0 \leq \psi_{2\mathrm{n}} \perp p_{2\mathrm{n}} \geq 0 \tag{3.2}$$

where $\psi_{i\mathrm{n}}$ is the distance to the potential edge e_i, $p_{i\mathrm{n}}$ is the non-penetration impulse required to prevent inter-penetration of edge i.

This formulation accurately models convex corners, such as in figure 3.2a,



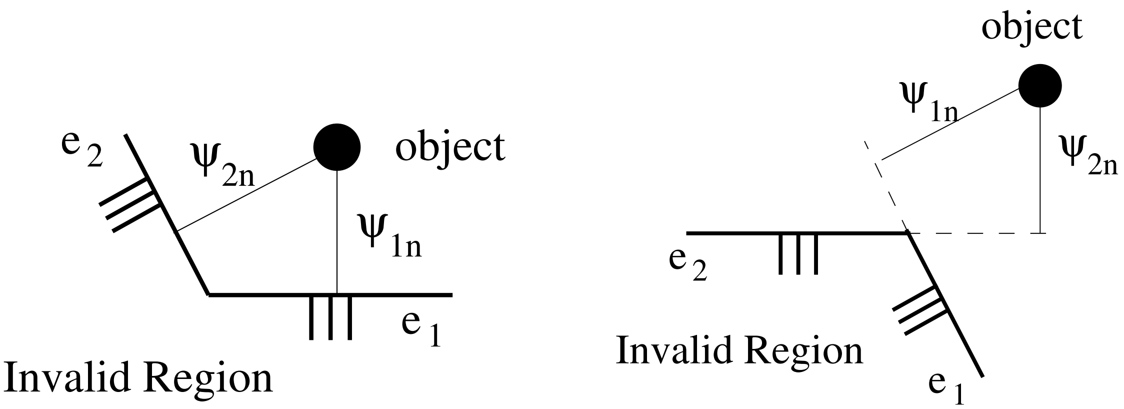
(a) (b) (c)

Conjuction

(d) (e) (f)

Disjunction

Figure 3.1: Screenshots of simulating a block sliding over a hole. In both examples there is a potential contact with the lower right corner of the block and the diagonal edge of the right side of the hole (3.1b, 3.1e). If the non-penetration constraint is represented as a conjunction of linear inequalities, then the diagonal edge on the right side of the hole is incorrectly modeled as extending forever (3.1c). Using a disjunction, the block correctly slides past the corner (3.1f).



(a) Convex

(b) Nonconvex

Figure 3.2: The convex feature on the left is correctly models as $\psi_{1n} \geq 0$ *and* $\psi_{2n} \geq 0$. The non-convex feature on the right is incorrectly modeled as a conjunction. The feature should be modeled as a disjunction, $\psi_{1n} \geq 0$ *or* $\psi_{2n} \geq 0$.

however fails to properly model the corner when it is non-convex, as shown in figure 3.2b. Given two edges e_1 and e_2 of a non-convex corner, the gap constraint should be modeled as,

$$0 \leq \psi_{1\text{n}} \text{ or } \psi_{2\text{n}} \geq 0 \tag{3.3}$$

and the non-penetration constraint is more properly modeled as,

$$p_{1\text{n}} \geq 0 \iff (\max\{\psi_{1\text{n}}, \psi_{2\text{n}}\} = 0) \wedge (\psi_{1\text{n}} = 0) \tag{3.4}$$

$$p_{2\text{n}} \geq 0 \iff (\max\{\psi_{1\text{n}}, \psi_{2\text{n}}\} = 0) \wedge (\psi_{2\text{n}} = 0) \tag{3.5}$$

Enforcing this constraint by use of complementarity is not trivial, and is still an active area of research [82]. Presented now are the first results for modeling this constraint, taken from an early tech report [36]. More details and alternative formulations can be found in the tech report and more recent formulations.

Ignoring friction at the contact, artificial variables c_1, c_2, d, g_1, g_2, h_1, h_2 and p are created. Additionally, a large positive constant γ is needed. The following equations are now used to model equations (3.4) and (3.5):

$$0 \leq c_1 + \psi_{1\text{n}} - 1 \perp c_1 \geq 0 \tag{3.6}$$

$$0 \leq c_2 + \psi_{2\text{n}} - 1 \perp c_2 \geq 0 \tag{3.7}$$

$$0 \leq g_1 + g_2 - 1 \perp d \geq 0 \tag{3.8}$$

$$0 \leq c_1 + d - \gamma \perp g_1 \geq 0 \tag{3.9}$$

$$0 \leq c_2 + d - \gamma \perp g_2 \geq 0 \tag{3.10}$$

$$0 \leq h_1 + \psi_{1\text{n}} \perp h_1 \geq 0 \tag{3.11}$$

$$0 \leq h_2 + \psi_{2\text{n}} \perp h_2 \geq 0 \tag{3.12}$$

$$0 \leq d - \gamma + 1 \perp p \geq 0 \tag{3.13}$$

$$0 \leq h_1 + d - \gamma + 1 \perp p_{1\text{n}} \geq 0 \tag{3.14}$$

$$0 \leq h_2 + d - \gamma + 1 \perp p_{2\text{n}} \geq 0 \tag{3.15}$$

Equations (3.6) and (3.7) constrain

$$c_1 = |\min\{\psi_{1\mathrm{n}}, 1\} - 1| \qquad\qquad c_2 = |\min\{\psi_{2\mathrm{n}}, 1\} - 1| \tag{3.16}$$

Equations (3.11) and (3.12) constrain

$$h_1 = |\min\{\psi_{1\mathrm{n}}, 0\}| \qquad\qquad h_2 = |\min\{\psi_{2\mathrm{n}}, 0\}| \tag{3.17}$$

Equations (3.8), (3.9) and (3.10) are designed to find the maximum value of $\psi_{1\mathrm{n}}$ and $\psi_{2\mathrm{n}}$.

As long as $\gamma > \max\{c_1, c_2\}$, we will have $d > 0$ from Equations (3.6), (3.7), (3.8) and (3.8). In general $d \geq \max\{\gamma - c_1, \gamma - c_2\}$. Because $d > 0$ the third equation forces $g_1 + g_2 = 1$, and therefore $g_1 > 0$ or $g_2 > 0$. To make g_1 or g_2 greater than 0 either $d + c1 - \gamma = 0$ or $d + c2 - \gamma = 0$. This fixes d to either $\gamma - c_1$ or $\gamma - c_2$, which in turn implies a strict equality $d = \max\{\gamma - c_1, \gamma - c_2\}$. We have manipulated d to select the smaller of the c_1 and c_2 values, and therefore select the maximum (least negative) of $\psi_{1\mathrm{n}}$ and $\psi_{1\mathrm{n}}$. Relating d to $\psi_{1\mathrm{n}}$ and $\psi_{1\mathrm{n}}$, we can say $d = \min\{\max\{a_1, a_2\}, 1\} + \gamma - 1$. Finally we have $d - \gamma + 1 = \min\{\max\{a_1, a_2\}, 1\}$. In the degenerate case where $c_1 = c_2$ both g_1 and g_2 may be greater than 0, but this does not change the relationship between d, $\psi_{1\mathrm{n}}$ and $\psi_{2\mathrm{n}}$.

If $d - \gamma + 1 > 0$ we have no contact, and if $d - \gamma + 1 = 0$ we do have contact. In equation (3.13) we have $d - \gamma + 1 \geq 0$, which enforces the non-penetration constraint. Note that p in equation (3.13) is not used in any other equation, and has no effect on the formulation. In equations (3.14) and (3.15), the normal force magnitude for individual planes of the nonconvex constraint are calculated. For $p_{1\mathrm{n}}$ to be greater than 0, both $d - \gamma + 1$ and h_1 must be 0. This enforces the rule that a contact force can only come from a plane that is in contact.

To make the formulation work, the constant value γ must be greater than $|\min\{a_1, a_2, 0\}|$ to force d to be positive in equations (3.9) and (3.10). To be conservative γ can be set to the diameter of the scene, or the twice the distance the fastest point can travel in the next time step.

3.2.2 Future Work

The LCP formulation as shown here for a non-convex corner does not include a frictional response. The incorporation of friction can be found in the tech report [36]. There are many problems with the presented formulation, it requires a parameter that must be tuned, requires a large amount of complementarity constraints, and was never implemented for 3D bodies. Ongoing work is being considered addressing all of these issues [82].

3.3 2.5D Model

There are many planar manipulation and assembly tasks in which we need to model surface friction between planar parts and a planar support surface. While this surface friction results in planar forces, they are dependent on the normal forces or the pressure distribution which contribute an out-of-plane component. The 2.5D model models the out-of-plane normal forces which do not directly play a role in determining the motion in the plane but do influence the frictional forces in the plane. We use the well-known model due to Mason [71], which states the surface pressure distribution can be modeled by an equivalent force distribution in a three-point support.

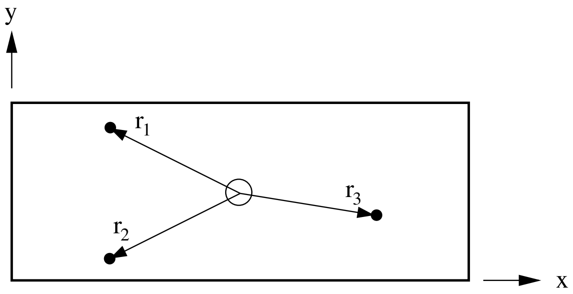


Figure 3.3: Surface friction distribution modeled with a three-point support tripod. The normal force at each contact point is solved for, and Coulomb's friction law is applied at each contact point using the known normal force.

The surface contact for the part is modeled by three non-collinear points whose triangle contains the center of mass of the part (figure 3.3). Coulomb's friction law is applied at each contact point, but a choice is available. This law may be used in its standard quadratic form, or linearized in order to obtain time step subproblems that are LCPs instead of NCPs.

3.3.1 Surface Friction Normal Force

In order to apply Coulomb's law at each of the three support points, we need to first solve for the surface normal force at the contact point. The vectors from the center of mass of the body to the three support points can be written as, $\mathbf{r}_1 = [x_1, y_1, 0]$, $\mathbf{r}_2 = [x_2, y_2, 0]$, and $\mathbf{r}_3 = [x_3, y_3, 0]$. The z-components of the vectors are all zero since these are 2D bodies that we are adding surface friction too.

Without loss of generality, the normal vector at each contact point, $\hat{\mathbf{n}}$, and the two basis vectors in the tangent plane, $\hat{\mathbf{t}}$ and $\hat{\mathbf{o}}$ can be defined as:

$$\hat{\mathbf{n}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \hat{\mathbf{t}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \hat{\mathbf{0}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{3.18}$$

This allows us to formulate the surface friction contact wrenches,

$$\mathbf{W}_{\mathrm{n}s} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ y_1 & y_2 & y_3 \\ -x_1 & -x_2 & -x_3 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{W}_{\mathrm{t}s} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -y_1 & -y_2 & -y_3 \end{bmatrix} \quad \mathbf{W}_{\mathrm{o}s} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ x_1 & x_2 & x_3 \end{bmatrix} \tag{3.19}$$

The next step is to solve for the surface friction normal force, $\boldsymbol{\lambda}_{\mathrm{n}s}$ in the Newton-Euler equations,

$$\mathbf{M}\dot{\boldsymbol{\nu}} = \mathbf{W}_{\mathrm{n}s}\boldsymbol{\lambda}_{\mathrm{n}s} + \mathbf{W}_{\mathrm{t}s}\boldsymbol{\lambda}_{\mathrm{t}s} + \mathbf{W}_{\mathrm{o}s}\boldsymbol{\lambda}_{\mathrm{o}s} + \boldsymbol{\lambda}_{\mathrm{app}} \tag{3.20}$$

The three non-zero rows in $\mathbf{W}_{\mathrm{n}s}$, correspond to rows of zeroes in $\mathbf{W}_{\mathrm{t}s}$ and

$\mathbf{W}_{os}$, so the equation simplifies to:

$$\widetilde{\mathbf{M}} \begin{bmatrix} \dot{\mathbf{v}}_z \\ \dot{\boldsymbol{\omega}}_x \\ \dot{\boldsymbol{\omega}}_y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ y_1 & y_2 & y_3 \\ -x_1 & -x_2 & -x_3 \end{bmatrix} \boldsymbol{\lambda}_{\mathrm{n}s} + \begin{bmatrix} -mg \\ 0 \\ 0 \end{bmatrix} \tag{3.21}$$

where $\widetilde{\mathbf{M}}$ is the mass matrix with the corresponding rows and columns removed and $-mg$ is the gravitation force. We can now solve for $\boldsymbol{\lambda}_{\mathrm{n}s}$,

$$\begin{bmatrix} 1 & 1 & 1 \\ y_1 & y_2 & y_3 \\ -x_1 & -x_2 & -x_3 \end{bmatrix} \boldsymbol{\lambda}_{\mathrm{n}s} = \widetilde{\mathbf{M}} \begin{bmatrix} \dot{\mathbf{v}}_z \\ \dot{\boldsymbol{\omega}}_x \\ \dot{\boldsymbol{\omega}}_y \end{bmatrix} - \begin{bmatrix} -mg \\ 0 \\ 0 \end{bmatrix} \tag{3.22}$$

$$\boldsymbol{\lambda}_{\mathrm{n}s} = \begin{bmatrix} 1 & 1 & 1 \\ y_1 & y_2 & y_3 \\ -x_1 & -x_2 & -x_3 \end{bmatrix}^{-1} \left(\widetilde{\mathbf{M}} \begin{bmatrix} \dot{\mathbf{v}}_z \\ \dot{\boldsymbol{\omega}}_x \\ \dot{\boldsymbol{\omega}}_y \end{bmatrix} - \begin{bmatrix} -mg \\ 0 \\ 0 \end{bmatrix} \right) \tag{3.23}$$

Since we are adding the surface friction constraint to a 2D body, $[\dot{\mathbf{v}}_z \; \dot{\boldsymbol{\omega}}_x \; \dot{\boldsymbol{\omega}}_y]^T = [0\;0\;0]^T$, and we can uniquely solve for $\boldsymbol{\lambda}_{\mathrm{n}s}$ as a function of the applied force,

$$\boldsymbol{\lambda}_{\mathrm{n}s} = \begin{bmatrix} 1 & 1 & 1 \\ y_1 & y_2 & y_3 \\ -x_1 & -x_2 & -x_3 \end{bmatrix}^{-1} \begin{bmatrix} mg \\ 0 \\ 0 \end{bmatrix} \tag{3.24}$$

3.3.2 Nonlinear DCP Formulation

We can now extend the 2D analog of equation (2.71) to include surface friction arriving at the nonlinear DCP 2.5D formulation:

$$
\begin{aligned}
\mathbf{M}(\mathbf{q})\dot{\boldsymbol{\nu}} &= \mathbf{W}_{\mathrm{n}}(\mathbf{q})\boldsymbol{\lambda}_{\mathrm{n}} + \mathbf{W}_{\mathrm{t}}(\mathbf{q})\boldsymbol{\lambda}_{\mathrm{t}} + \mathbf{W}_{\mathrm{o}}(\mathbf{q})\boldsymbol{\lambda}_{\mathrm{o}} + \boldsymbol{\lambda}_{\mathrm{app}}(\mathbf{q}, t) \\
\dot{\mathbf{q}} &= \boldsymbol{\nu} \\
0 &= \Phi(\mathbf{q}, t) \\
0 &= (\mathbf{U}_s\boldsymbol{\lambda}_{\mathrm{n}s})\circ(\mathbf{v}_{\mathrm{t}}) + \boldsymbol{\lambda}_{\mathrm{t}s}\circ\boldsymbol{\sigma}_s \\
0 &= (\mathbf{U}_s\boldsymbol{\lambda}_{\mathrm{n}s})\circ(\mathbf{v}_{\mathrm{o}}) + \boldsymbol{\lambda}_{\mathrm{o}s}\circ\boldsymbol{\sigma}_s \\
0 &= (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}})\circ(\mathbf{v}_{\mathrm{t}}) + \boldsymbol{\lambda}_{\mathrm{t}}\circ\boldsymbol{\sigma} \\
0 &= (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}})\circ(\mathbf{v}_{\mathrm{o}}) + \boldsymbol{\lambda}_{\mathrm{o}}\circ\boldsymbol{\sigma} \\
0 \leq \boldsymbol{\lambda}_{\mathrm{n}} &\perp \boldsymbol{\psi}_{\mathrm{n}}(\mathbf{q}, t) \geq 0 \\
0 \leq \boldsymbol{\sigma}_s &\perp (\mathbf{U}_s\boldsymbol{\lambda}_{\mathrm{n}s})\circ(\mathbf{U}_s\boldsymbol{\lambda}_{\mathrm{n}s}) - \boldsymbol{\lambda}_{\mathrm{t}s}\circ\boldsymbol{\lambda}_{\mathrm{t}s} - \boldsymbol{\lambda}_{\mathrm{o}s}\circ\boldsymbol{\lambda}_{\mathrm{o}s} \geq 0 \\
0 \leq \boldsymbol{\sigma} &\perp (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}})\circ(\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}) - \boldsymbol{\lambda}_{\mathrm{t}}\circ\boldsymbol{\lambda}_{\mathrm{t}} - \boldsymbol{\lambda}_{\mathrm{o}}\circ\boldsymbol{\lambda}_{\mathrm{o}} \geq 0
\end{aligned}
\tag{3.25}
$$

where $\mathbf{q} = [x\ y\ \theta]^T$, $\boldsymbol{\nu} = [\dot{x}\ \dot{y}\ \dot{\theta}]^T$, $\mathbf{U}_s \in \mathbb{R}^{3\times 3}$ is the matrix of support point friction coefficients, $\boldsymbol{\lambda}_{\mathrm{n}s} \in \mathbb{R}^3$ is a *known* vector containing the support points' normal forces, and $\boldsymbol{\sigma}_s$ is the surface friction slip speed.

3.3.3 Linear DCP Formulation

The linearized model can be written in a form similar to that given in equation (2.72), after incorporating the additional elements corresponding to the tangential surface frictions and normal friction force vector:

$$
\begin{bmatrix} 0 \\ \boldsymbol{\rho}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\rho}_{\mathrm{fs}}^{\ell+1} \\ \boldsymbol{\rho}_{\mathrm{f}}^{\ell+1} \\ \mathbf{s}_s^{\ell+1} \\ \mathbf{s}^{\ell+1} \end{bmatrix}
=
\begin{bmatrix}
-\mathbf{M} & \mathbf{W}_{\mathrm{n}} & \mathbf{W}_{\mathrm{f}s} & \mathbf{W}_{\mathrm{f}} & 0 & 0 \\
\mathbf{W}_{\mathrm{n}}^T & 0 & 0 & 0 & 0 & 0 \\
\mathbf{W}_{\mathrm{f}s}^T & 0 & 0 & 0 & \mathbf{E}_s & 0 \\
\mathbf{W}_{\mathrm{f}}^T & 0 & 0 & 0 & 0 & \mathbf{E} \\
0 & 0 & -\mathbf{E}_s^T & 0 & 0 & 0 \\
0 & \mathbf{U} & 0 & -\mathbf{E}^T & 0 & 0
\end{bmatrix}
\begin{bmatrix} \boldsymbol{\nu}^{\ell+1} \\ \mathbf{p}_{\mathrm{n}}^{\ell+1} \\ \mathbf{p}_{\mathrm{f}s}^{\ell+1} \\ \mathbf{p}_{\mathrm{f}}^{\ell+1} \\ \boldsymbol{\sigma}_s^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix}
+
\begin{bmatrix} \mathbf{M}\boldsymbol{\nu}^{\ell} + h\boldsymbol{\lambda}_{\mathrm{app}} \\ \frac{1}{h}\boldsymbol{\psi}_{\mathrm{n}}^{\ell} + \frac{\partial \boldsymbol{\psi}_{\mathrm{n}}^{\ell}}{\partial t} \\ 0 \\ \frac{\partial \boldsymbol{\Psi}_{\mathrm{f}}^{\ell}}{\partial t} \\ h\mathbf{U}_s\boldsymbol{\lambda}_{\mathrm{n}s} \\ 0 \end{bmatrix}
\tag{3.26}
$$

$$0 \le \begin{bmatrix} \boldsymbol{\rho}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\rho}_{\mathrm{fs}}^{\ell+1} \\ \boldsymbol{\rho}_{\mathrm{f}}^{\ell+1} \\ \mathbf{s}_s^{\ell+1} \\ \mathbf{s}^{\ell+1} \end{bmatrix} \perp \begin{bmatrix} \mathbf{p}_{\mathrm{n}}^{\ell+1} \\ \mathbf{p}_{\mathrm{f}s}^{\ell+1} \\ \mathbf{p}_{\mathrm{f}}^{\ell+1} \\ \boldsymbol{\sigma}_s^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix} \ge 0 \tag{3.27}$$

where $\mathbf{q} = [x\ y\ \theta]^T$, $\boldsymbol{\nu} = [\dot{x}\ \dot{y}\ \dot{\theta}]^T$, $\mathbf{W}_{\mathrm{f}s} \in \mathbb{R}^{3\times 3n_d}$ has n_d directions for each surface contact, $\mathbf{U}_s \in \mathbb{R}^{3\times 3}$ is the matrix of support point friction coefficients, $\boldsymbol{\lambda}_{\mathrm{n}s} \in \mathbb{R}^3$ contains the support points' normal impulses. and $\boldsymbol{\sigma}_s$, $\mathbf{p}_{\mathrm{f}s}$, and $\mathbf{E}_s$ are the surface friction support tripod contact analogs to $\boldsymbol{\sigma}$, $\mathbf{p}_{\mathrm{f}}$ and $\mathbf{E}$ respectively.

3.4 Quasistatic Formulation

In quasistatic formulations, the sum of the external forces must equal zero. This model change shows up in the left hand side of equation (2.35), where the term $\mathbf{M}(\mathbf{q}, t)\dot{\boldsymbol{\nu}}$ is replaced with 0:

$$0 = \boldsymbol{\lambda}_{\mathrm{vp}}(\mathbf{q}, \boldsymbol{\nu}, t) + \boldsymbol{\lambda}_{\mathrm{app}}(\mathbf{q}, t) \tag{3.28}$$

Pang, Trinkle, and Lo [87] studied the quasistatic planar motion, with a focus on better understanding the quasistatic contact problem. Unlike the time stepper formulation shown previously, they formulated the problem as an uncoupled complementarity problem and used linear programming techniques to solve it.

Trinkle, Berard, and Pang [112] presented two quasistatic multi-rigid-body models: one linear and the other nonlinear. Using these models they formulated two semi-implicit complementarity problems, a mixed LCP and a mixed NCP respectively. Using these models they were able to develop a theorem for solution uniqueness for quasistatic systems. These results are now presented.

3.4.1 Instantaneous-time models

A quasistatic system must satisfy equilibrium at all times, therefore equation (3.28) is substituted into equation (2.71), arriving at the Instantaneous time model:

$$0 = \mathbf{W}_{\mathrm{n}}(\mathbf{q})\boldsymbol{\lambda}_{\mathrm{n}} + \mathbf{W}_{\mathrm{t}}(\mathbf{q})\boldsymbol{\lambda}_{\mathrm{t}} + \mathbf{W}_{\mathrm{o}}(\mathbf{q})\boldsymbol{\lambda}_{\mathrm{o}} + \boldsymbol{\lambda}_{\mathrm{vp}}(\mathbf{q}, \boldsymbol{\nu}, t) + \boldsymbol{\lambda}_{\mathrm{app}}(\mathbf{q}, t) \tag{3.29}$$

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\boldsymbol{\nu} \tag{3.30}$$

$$0 = \Phi(\mathbf{q}, t) \tag{3.31}$$

$$0 = (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}})\circ(\mathbf{v}_{\mathrm{t}}) + \boldsymbol{\lambda}_{\mathrm{t}}\circ\boldsymbol{\sigma} \tag{3.32}$$

$$0 = (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}})\circ(\mathbf{v}_{\mathrm{o}}) + \boldsymbol{\lambda}_{\mathrm{o}}\circ\boldsymbol{\sigma} \tag{3.33}$$

$$0 \leq \boldsymbol{\lambda}_{\mathrm{n}} \perp \boldsymbol{\psi}_{\mathrm{n}}(\mathbf{q}, t) \geq 0 \tag{3.34}$$

$$0 \leq \boldsymbol{\sigma} \perp (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}})\circ(\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}) - \boldsymbol{\lambda}_{\mathrm{t}}\circ\boldsymbol{\lambda}_{\mathrm{t}} - \boldsymbol{\lambda}_{\mathrm{o}}\circ\boldsymbol{\lambda}_{\mathrm{o}} \geq 0 \tag{3.35}$$

3.4.2 Discrete-time models

In the following, two time-stepping schemes will be presented. The unknowns for both are the configuration vector, contact forces, and sliding speeds at the end of the time step.

Again, let t^{ℓ} and denote the time at which one has a solution and let $t^{\ell+1} = t^{\ell} + h$ denote the time at which one would like an estimate of the solution. To eliminate $\boldsymbol{\nu}$, $\dot{\mathbf{q}}$ can be approximated using a backward Euler formula as follows:

$$\Delta\mathbf{q} = \mathbf{q}^{\ell+1} - \mathbf{q}^{\ell} = \mathbf{G}(\mathbf{q})\boldsymbol{\nu}^{\ell+1}h \tag{3.36}$$

Note that since $\Delta\mathbf{q}$ is in the range of $\mathbf{G}$ (see equation (2.36)), the following useful identity holds: $\Delta\mathbf{q} = \mathbf{G}\mathbf{G}^T\Delta\mathbf{q}$.

3.4.3 A mildly nonlinear model: Model-DQC

After substituting equation (3.36) into **Model-IQC**, and replacing all occurrences of the variables ($\mathbf{q}$, $\boldsymbol{\lambda}_{\mathrm{n}}$, $\boldsymbol{\lambda}_{\mathrm{t}}$, $\boldsymbol{\lambda}_{\mathrm{o}}$, $\boldsymbol{\sigma}$) with their values at the end of the time step, ($\mathbf{q}^{\ell+1}$, $\boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1}$, $\boldsymbol{\lambda}_{\mathrm{t}}^{\ell+1}$, $\boldsymbol{\lambda}_{\mathrm{o}}^{\ell+1}$, $\boldsymbol{\sigma}^{\ell+1}$), all model equations are nonlinear in the unknowns.

To remove some of the nonlinearities from the time-stepper, let $\mathbf{W}_{\mathrm{n}}$, $\mathbf{W}_{\mathrm{t}}$, $\mathbf{W}_{\mathrm{o}}$, $\mathbf{G}$, $\boldsymbol{\lambda}_{\mathrm{app}}$, and $\boldsymbol{\lambda}_{\mathrm{vp}}$ be evaluated at $\mathbf{q}^{\ell}$. In addition, let the distance function vector be approximated by the linear terms in its Taylor series expansion:

$$0 \leq \boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1} \;\perp\; \mathbf{W}_{\mathrm{n}}^{T}\mathbf{G}^{T}\mathbf{q}^{\ell+1} + \mathbf{b}_{\mathrm{n}} \geq 0 \tag{3.37}$$

where $\mathbf{b}_{\mathrm{n}} = \boldsymbol{\psi}_{\mathrm{n}}^{\ell} + \frac{\partial \boldsymbol{\psi}_{\mathrm{n}}^{\ell}}{\partial t} h - \mathbf{W}_{\mathrm{n}}^{T}\mathbf{G}^{T}\mathbf{q}^{\ell}$. Now the only remaining nonlinearities are the quadratic terms in Coulomb's law. The result is a mildly nonlinear discrete-time model, **Model-DQC**. For each time step, the NCP composed of equations (3.29, 3.37) and the following must be solved:

$$0 = (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}) \circ (\mathbf{W}_{\mathrm{t}}^{T}\mathbf{G}^{T}\mathbf{q} + \mathbf{b}_{\mathrm{t}}) + \boldsymbol{\lambda}_{\mathrm{t}} \circ \boldsymbol{\sigma} h \tag{3.38}$$

$$0 = (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}) \circ (\mathbf{W}_{\mathrm{o}}^{T}\mathbf{G}^{T}\mathbf{q} + \mathbf{b}_{\mathrm{o}}) + \boldsymbol{\lambda}_{\mathrm{o}} \circ \boldsymbol{\sigma} h \tag{3.39}$$

$$0 \leq \boldsymbol{\sigma} \perp (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}) \circ (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}) - \boldsymbol{\lambda}_{\mathrm{t}} \circ \boldsymbol{\lambda}_{\mathrm{t}} - \boldsymbol{\lambda}_{\mathrm{o}} \circ \boldsymbol{\lambda}_{\mathrm{o}} \geq 0 \tag{3.40}$$

where the variables $\mathbf{q}$, $\boldsymbol{\lambda}_{\mathrm{n}}$, $\boldsymbol{\lambda}_{\mathrm{t}}$, $\boldsymbol{\lambda}_{\mathrm{o}}$, and $\boldsymbol{\sigma}$ appearing in equations (3.38-3.40) are to be evaluated at time $t^{\ell+1}$, $\mathbf{b}_{\mathrm{t}} = \frac{\partial \boldsymbol{\psi}_{\mathrm{t}}^{\ell}}{\partial t} h - \mathbf{W}_{\mathrm{t}}^{T}\mathbf{G}^{T}\mathbf{q}^{\ell}$ and $\mathbf{b}_{\mathrm{o}} = \frac{\partial \boldsymbol{\psi}_{\mathrm{o}}^{\ell}}{\partial t} h - \mathbf{W}_{\mathrm{o}}^{T}\mathbf{G}^{T}\mathbf{q}^{\ell}$.

Summary of **Model-DQC**:
For each time step, solve mixed NCP of size $n_{\mathrm{q}} + 4n_{\mathrm{c}}$ defined by equations (3.29,3.37-3.40).

3.4.4 A linear model: Model-DLC

The other discrete-time model of interest, **Model-DLC** can be derived from **Model-ILC** by the same procedure. The result is a mixed LCP defined as follows:

$$\begin{bmatrix} 0 \\ \boldsymbol{\rho}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\rho}_{\mathrm{f}}^{\ell+1} \\ \mathbf{s} \end{bmatrix} = \mathbf{B} \begin{bmatrix} \mathbf{q}^{\ell+1} \\ \boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\lambda}_{\mathrm{f}}^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix} + \; \mathbf{b} \tag{3.41}$$

$$0 \leq \begin{bmatrix} \boldsymbol{\rho}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\rho}_{\mathrm{f}}^{\ell+1} \\ \mathbf{s} \end{bmatrix} \perp \begin{bmatrix} \boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\lambda}_{\mathrm{f}}^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix} \geq 0 \tag{3.42}$$

where

$$\mathbf{B} = \begin{bmatrix} 0 & \mathbf{W}_\mathrm{n} & \mathbf{W}_\mathrm{f} & 0 \\ \mathbf{W}_\mathrm{n}^T\mathbf{G}^T & 0 & 0 & 0 \\ \mathbf{W}_\mathrm{f}^T\mathbf{G}^T & 0 & 0 & \mathbf{E} \\ 0 & \mathbf{U} & -\mathbf{E}^T & 0 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} \boldsymbol{\lambda}_\mathrm{app} + \boldsymbol{\lambda}_\mathrm{vp} \\ \mathbf{b}_\mathrm{n} \\ \mathbf{b}_\mathrm{f} \\ 0 \end{bmatrix}, \tag{3.43}$$

$\mathbf{b}_\mathrm{n}$ is defined as above, and $\mathbf{b}_\mathrm{f} = \frac{\partial \boldsymbol{\Psi}_\mathrm{f}^\ell}{\partial t} h - \mathbf{W}_\mathrm{f}^T\mathbf{G}^T\mathbf{q}^\ell$.

Summary of **Model-DLC**:

For each time step, solve mixed LCP of size $n_\mathrm{q} + (2 + n_\mathrm{d})n_\mathrm{c}$ defined by equations (3.41,3.42).

3.4.5 Uniqueness

The theorem presented here is the first known solution uniqueness result for general quasistatic multibody systems with dry friction. It applies only to the discrete-time models, **Model-DQC** and **Model-DLC**.

Before stating the result, the friction force components can be written as the following functions of the normal force component and the relative tangential displacement components $\Delta_{i\mathrm{t}} = W_{i\mathrm{t}}^T\mathbf{G}^T\mathbf{q}^{\ell+1} + b_{i\mathrm{t}}$ and $\Delta_{i\mathrm{o}} = W_{i\mathrm{o}}^T\mathbf{G}^T\mathbf{q}^{\ell+1} + b_{i\mathrm{o}}$:

$$\begin{aligned} \lambda_{i\mathrm{t}} &= -\mu_i\, \lambda_{in}\, \frac{\Delta_{i\mathrm{t}}}{\sqrt{\Delta_{i\mathrm{t}}^2 + \Delta_{i\mathrm{o}}^2}} \\ \lambda_{i\mathrm{o}} &= -\mu_i\, \lambda_{in}\, \frac{\Delta_{i\mathrm{o}}}{\sqrt{\Delta_{i\mathrm{t}}^2 + \Delta_{i\mathrm{o}}^2}} \end{aligned} \tag{3.44}$$

where when $\Delta_{i\mathrm{t}} = \Delta_{i\mathrm{o}} = 0$, the fractions appearing in equation (3.44) are both equal to $0/0$, and are taken to be a suitable pair of scalars (α, β) such that $\alpha^2 + \beta^2 \leq 1$.

For given $\{\mu_i\lambda_{in}\}_{i=1}^{n_c}$, consider the following convex, nondifferentiable optimization problem in the variable $\mathbf{q}^{\ell+1}$:

$$\begin{aligned} &\min -\boldsymbol{\lambda}_\mathrm{app}^T\mathbf{q}^{\ell+1} + \sum_{i=1}^{n_c} \mu_i\, \lambda_{in}\, \sqrt{\Delta_{i\mathrm{t}}^2 + \Delta_{i\mathrm{o}}^2} \\ &\text{s.t.:} \quad \mathbf{W}_\mathrm{n}^T\mathbf{G}^T\mathbf{q}^{\ell+1} + \mathbf{b}_n \,\geq\, 0 \end{aligned} \tag{3.45}$$

where recall that Δ_{it} and Δ_{io} are functions of $\mathbf{q}^{\ell+1}$. The physical interpretation of this problem is that the displacement of the system is one that avoids penetration while minimizing the work done against external and frictional forces. In other words, the system is "lazy" and so moves no more than it absolutely must.

The following result describes the precise connection between the above optimization problem (3.45) and the discrete-time model **Model-DQC**.

Theorem 1. *If* $(\mathbf{q}^{\ell+1}, \boldsymbol{\lambda}_n, \boldsymbol{\lambda}_t, \boldsymbol{\lambda})$ *solves* **Model-DQC** *then* $\mathbf{q}^{\ell+1}$ *is a globally optimal solution to (3.45) corresponding to* $\boldsymbol{\lambda}_n$. *Conversely, if* $\mathbf{q}^{\ell+1}$ *is a globally optimal solution to (3.45) for a given* $\boldsymbol{\lambda}_n$ *and if* $\boldsymbol{\lambda}_n$ *is equal to an optimal Karush-Kuhn-Tucker (KKT) multiplier of the constraint in (3.45), then defining* $(\boldsymbol{\lambda}_t, \boldsymbol{\lambda}_o)$ *by (3.44), the tuple* $(\mathbf{q}^{\ell+1}, \boldsymbol{\lambda}_n, \boldsymbol{\lambda}_t, \boldsymbol{\lambda}_o)$ *solves* **Model-DQC**.

A question relevant to the design of fixed-point time stepping schemes is whether or not the convex optimization problem (3.45) has a unique solution, for fixed $\{\mu_i \lambda_{in}\}_{i=1}^{n_c}$. Let $(\mathbf{q}^{\ell+1}, \boldsymbol{\lambda}_{\mathrm{n}}, \boldsymbol{\lambda}_{\mathrm{t}}, \boldsymbol{\lambda}_{\mathrm{o}})$, solve **Model-DQC**. Denote by $\mathbf{dq}$ a small change in $\mathbf{q}^{\ell+1}$, and define the index sets:

$$\mathcal{I} \;\equiv\; \{\, i \,:\, \psi_{\mathrm{in}} \,=\, 0 \,<\, \lambda_{in} \,\} \tag{3.46}$$

$$\mathcal{J} \;\equiv\; \{\, i \,:\, \psi_{\mathrm{in}} \,=\, 0 \,=\, \lambda_{in} \,\}. \tag{3.47}$$

Proposition 1. *Corresponding to the solution* $(\mathbf{q}^{\ell+1}, \boldsymbol{\lambda}_n, \boldsymbol{\lambda}_t, \boldsymbol{\lambda}_o)$ *of* **Model-DQC**, $\mathbf{q}^{\ell+1}$ *is the unique solution of (3.45) if and only if the following implication holds:*

$$\left.\begin{array}{ll} W_{in}^T \mathbf{G}^T \mathbf{dq} \,\geq\, 0, & i \,\in\, \mathcal{I} \cup \mathcal{J} \\ W_{it}^T \mathbf{G}^T \mathbf{dq} \,=\, 0, & i \,\in\, \mathcal{I} \\ W_{io}^T \mathbf{G}^T \mathbf{dq} \,=\, 0, & i \,\in\, \mathcal{I} \\ (\boldsymbol{\lambda}_{app})^T \mathbf{dq} \,\geq\, 0 & \end{array}\right\} \;\Rightarrow\; \mathbf{dq} \,=\, 0. \tag{3.48}$$

Finally, consider an alternative model where the quadratic friction cone at each contact i is replaced by a four-sided linearized cone:

$$\{\, (\, \lambda_{\mathrm{it}}, \lambda_{io} \,) \,:\, \max(|\lambda_{\mathrm{it}}|, |\lambda_{io}|) \,\leq\, \mu_i \, \lambda_{in} \,\}. \tag{3.49}$$

In this case, instead of (3.44), we have

$$\begin{aligned}\lambda_{i\mathrm{t}} &= -\mu_i \lambda_{i\mathrm{n}} \frac{\Delta_{i\mathrm{t}}}{|\Delta_{i\mathrm{t}}|} \\ \lambda_{i\mathrm{o}} &= -\mu_i \lambda_{i\mathrm{n}} \frac{\Delta_{i\mathrm{o}}}{|\Delta_{i\mathrm{o}}|}.\end{aligned} \tag{3.50}$$

Moreover, a result similar to Theorem 1 holds with the optimization problem (3.45) replaced by the following linear program:

$$\begin{aligned}&\min -\boldsymbol{\lambda}_{\mathrm{app}}^T \mathbf{q}^{\ell+1} + \sum_{i=1}^{n_c} \mu_i \, \lambda_{i\mathrm{n}} \; (\,|\,\Delta_{i\mathrm{t}}\,| + |\,\Delta_{i\mathrm{o}}\,|) \\ &\text{s.t.:} \quad \mathbf{W}_{\mathrm{n}}^T \mathbf{G}^T \mathbf{q}^{\ell+1} + \mathbf{b}_n \;\geq\; 0\end{aligned} \tag{3.51}$$

where again recall that $\Delta_{i\mathrm{t}}$ and $\Delta_{i\mathrm{o}}$ are functions of $\mathbf{q}^{\ell+1}$.

3.4.6 Example: Fence-Particle Problem

Consider the problem of manipulating a particle (shown as a finite disc) of mass m initially at rest on a horizontal plane (the (x, y)-plane in Figure 3.4). The configuration of this system is $\mathbf{q} = [x_{\mathrm{p}} \; y_{\mathrm{p}} \; z_{\mathrm{p}}]^T$, where z_{p} is the height of the particle above the plane (of the page). The wall on the right is parallel to the (y, z)-plane (perpendicular to the plane of the page) and of infinite extent. The fence is parallel to the wall, of infinite extent, and can translate in the x- and y-directions, but cannot translate in the z-direction or rotate.[8] The vector of noncontact and noninertial forces $\boldsymbol{\lambda}_{\mathrm{app}} = [0 \;\; 0 \;\; -mg]^T$ is the gravitational force which acts in the negative z-direction.

The three nonpenetration constraints, $\boldsymbol{\psi}_{\mathrm{n}}(\mathbf{q}, t) = [\psi_{1\mathrm{n}}(\mathbf{q}) \;\; \psi_{2\mathrm{n}}(\mathbf{q}, t) \;\; \psi_{3\mathrm{n}}(\mathbf{q})]^T$ are written as:

$$\psi_{1n} = 1 - x_{\mathrm{p}} \;\;\geq\;\; 0 \tag{3.52}$$

$$\psi_{2n} = x_{\mathrm{p}} - x_{\mathrm{fence}}(t) \;\;\geq\;\; 0 \tag{3.53}$$

$$\psi_{3n} = z_{\mathrm{p}} \;\;\geq\;\; 0. \tag{3.54}$$

[8] The latter constraint is to simplify the problem making the particle remain within the (x, y)-plane.

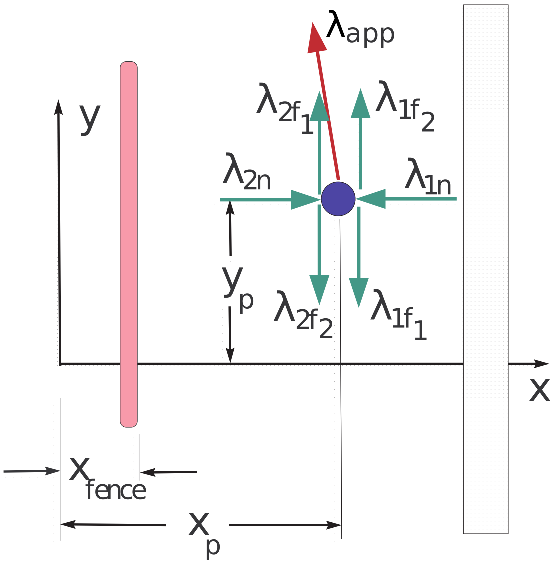


Figure 3.4: Schematic of fence-particle system.

The corresponding Lagrange multipliers are the normal components of the contact forces, $\boldsymbol{\lambda}_{\rm n}(\mathbf{q}, t) = [\lambda_{\rm 1n} \quad \lambda_{\rm 2n} \quad \lambda_{\rm 3n}]^T$. Even though as shown, the particle is not in contact with the fence or wall on the right, the components of the corresponding contact forces are shown.[9] The possible contact force components between the particle and the plane are not shown.

In this example, solution uniqueness will be explored for two different friction laws for the contact between the particle and the (y, z)-plane: no friction and quadratic friction. An interesting point, is that for dynamic systems, the absence of friction guarantees solution existence and uniqueness of the predicted motion (not necessarily uniqueness of the contact forces) and the inclusion of friction leads to motion nonuniqueness. In the quasistatic system studied here, the reverse is true. For the case of linearized friction, the quadratic cone will be approximated by a four-sided friction pyramid (see Figure 3.5). The various friction direction vectors at the three potential contacts imply the following definitions of the local tangential

[9] Since translation in the z-direction is not possible in this problem, friction forces can act only in the plane of motion of the particle. This is why there are only two friction force directions for contacts 1 and 2.

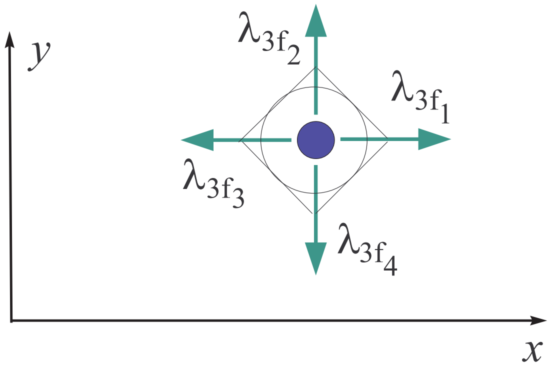


Figure 3.5: Friction direction vectors between the particle and the (x, y)-plane.

displacement functions:

$$(\psi_{1\mathrm{f}})_1 = -y_\mathrm{p} \tag{3.55}$$

$$(\psi_{1\mathrm{f}})_2 = y_\mathrm{p} \tag{3.56}$$

$$(\psi_{2\mathrm{f}})_1 = y_\mathrm{p} - y_\mathrm{fence}(t) \tag{3.57}$$

$$(\psi_{2\mathrm{f}})_2 = -y_\mathrm{p} + y_\mathrm{fence}(t) \tag{3.58}$$

$$\psi_{3\mathrm{t}} = (\psi_{3\mathrm{f}})_1 = x_\mathrm{p} \tag{3.59}$$

$$\psi_{3\mathrm{o}} = (\psi_{3\mathrm{f}})_2 = y_\mathrm{p} \tag{3.60}$$

$$(\psi_{3\mathrm{f}})_3 = -x_\mathrm{p} \tag{3.61}$$

$$(\psi_{3\mathrm{f}})_4 = -y_\mathrm{p}. \tag{3.62}$$

where $y_\mathrm{fence}(t)$ is the vertical position of the fence.

The various submatrices appearing in the matrix $\mathbf{B}$ are:

$$\mathbf{W}_{\text{n}} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{W}_{\text{f}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ -1 & 1 & 1 & -1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.63}$$

$$\mathbf{U} = \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{bmatrix} \qquad \mathbf{E} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.64}$$

Also, since the particle is a point mass, the matrix $\mathbf{G}$ is simply the identity matrix of size 3.

Other matrices for the nonlinear problem are

$$\mathbf{W}_{\text{t}} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{W}_{\text{o}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \tag{3.65}$$

The time-dependent functions needed to define the vectors $\mathbf{b}_{\text{n}}$, $\mathbf{b}_{\text{t}}$, $\mathbf{b}_{\text{o}}$, and $\mathbf{b}_{\text{f}}$ were chosen as:

$$x_{\text{fence}}(t) = 0.5 + 0.4\sin(t) \tag{3.66}$$

$$y_{\text{fence}}(t) = t \tag{3.67}$$

With these choices, the fence translates in the y direction while oscillating in the x-direction without ever hitting the wall.

3.4.6.1 Results

Various values of the problem data were chosen to illustrate the theorems given in section 3.4.5. One common aspect of these problems is that the only forces that can act in the z-direction are the gravitational force and the normal component of the contact force between the particle and the (x, y)-plane. This implies that $\lambda_{3\text{n}} = mg > 0$ and $\psi_{3\text{n}} = 0$.

Results: Model-DLC, no friction As stated earlier, the frictionless example of **Model-DLC** has many solutions. Looking back at Proposition 1, the 2nd and 3rd rows of implication (3.48) are vacuous in the absence of friction. It is the removal of these equalities from the implication that allow the construction of a $\mathbf{dq} \neq 0$ satisfying the two remaining inequalities, breaking the implication. To do this, assume a solution of the mixed LCP with contact between the particle and the (x, y)-plane, but not with the wall or fence. In this case, we have $W_{in}^T = [0 \;\; 0 \;\; 1]$ and $\boldsymbol{\lambda}_{\text{app}}^T = [0 \;\; 0 \;\; -mg]$. Let $\mathbf{dq} = [dx \;\; dy \;\; dz]^T$. The inequalities in this stripped down version of implication (3.48) yield $dz = 0$, but dx and dy are unconstrained. Since there exists a $\mathbf{dq} \neq 0$ satisfying the left hand side of the implication, the implication does not hold. Therefore, by applying Proposition 1, the solution of $\mathbf{q}^{\ell+1}$ **Model-DLC** is not unique. In this particular case, the possible $\mathbf{q}^{\ell+1}$ solving **Model-DLC** are all those for which the particle remains in contact with the (x, y)-plane, and between the wall and fence. This conclusion was observed in practice. Specifically, the solution obtained was dependent on the initial guess.

Results: Model-DQC From the frictionless case, we saw how the stripped down version of implication (3.48) was only capable of constraining the z-component of $\mathbf{dq}$ to 0. Now with friction present, we do not lose any rows of the implication, and we will see how the implication holds true for all $\mathbf{dq}$.

Again, consider a solution for the system when the particle is not touching the fence or wall and the quadratic friction law is in effect at the contact with the (x, y)-plane. In this case, the matrices $\mathbf{W}_{\text{t}}$ and $\mathbf{W}_{\text{o}}$ are given as follows:

$$\mathbf{W}_{\text{t}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad\qquad \mathbf{W}_{\text{o}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \tag{3.68}$$

and $\mathbf{W}_{\text{n}}$ and $\boldsymbol{\lambda}_{\text{app}}$ are as in the frictionless case.

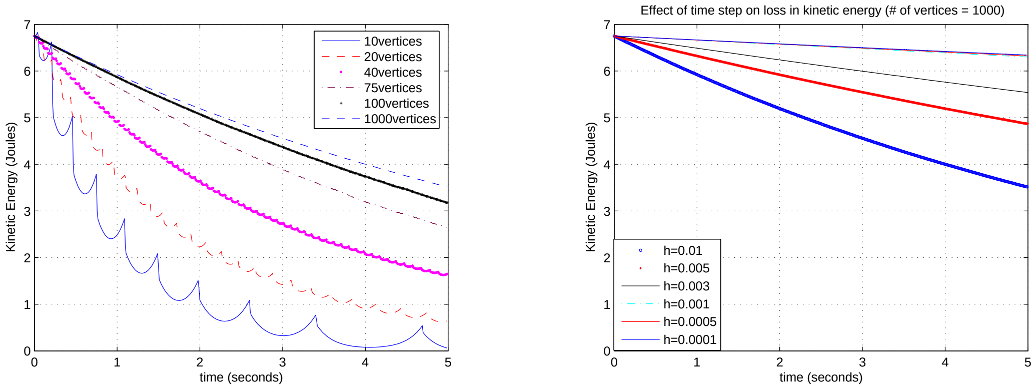
Again, let $\mathbf{dq} = [dx \;\; dy \;\; dz]^T$. It is easily seen how the left hand side of the implication now forces dx, dy, and dz to all be 0. Since the implication holds, by Theorem 1 all $\mathbf{q}^{\ell+1}$ are unique. In this case, if over the course of a time step the fence will not reach the particle, the particle will not move.

Now, consider a solution in which the particle is in contact with the fence. In this case, the matrices $\mathbf{W}_\mathrm{t}$ and $\mathbf{W}_\mathrm{o}$ gain rows, but do not change the conclusion - the motion of the particle is unique.

3.5 Geometrically Implicit Formulation

This section presents our work [24, 23] on developing a *geometrically implicit* optimization-based time-stepper for dynamic simulation. State-of-the-art time steppers [110, 102, 66] use geometric information obtained from a collision detection algorithm at the current time, and the state of the system at the end of the time step is computed (by solving a dynamics time step subproblem) without modifying this information. Thus, state-of-the-art time-steppers can be viewed as *explicit* methods with respect to geometric information. We developed the first time-stepping method that is implicit in the geometric information (when the distance function is not available in closed form) by incorporating body geometry in the dynamic time-stepping subproblem. In other words, our formulation solves the collision detection and dynamic stepping problem in the same time-step, which allows us to satisfy contact constraints at the end of the time step. This allows us to side-step the unsolved problem of identifying a relevant set of potential contacts [7, 60].

To illustrate the effects of geometric approximation, consider the simple planar problem of a uniform disc rolling on a horizontal support surface. For this problem, the exact solution is known, *i.e.,* the disc will roll at constant speed *ad infinitum.* However, when the disc is approximated by a uniform regular polygon, energy is lost a) due to collisions between the vertices and the support surface, b) due to contact sliding that is resisted by friction and c) due to artificial impulses generated by the approximate distance function that is to be satisfied at the end of the time-step. We simulated this example in dVC [18] using the Stewart-Trinkle time-stepping algorithm [110]. The parametric plots in Figure 3.6 show the reduction of kinetic energy over time caused by the accumulation of these effects. The top plot shows that increasing the number of edges, with the step-size fixed, decreases the energy loss; the energy loss approaches a limit determined by the size of the time-step. The bottom plot shows reducing energy loss with decreasing step size, with the number of vertices fixed at 1000. However, even with the decrease in time-step an energy loss limit is reached. These plots make it clear that the discretization of geometry and linearization of the distance function lead to the artifact of loss in energy in the simulations.



(a) As the number of edges of the “rolling” polygon increases, the energy loss decreases. The computed value obtained by our time-stepper using an implicit surface description of the disc is the horizontal line at the top. The time step used is 0.01 seconds.

(b) For a given number of edges, the energy loss decreases with decreasing step size, up to a limit. In this case, the limit is approximately 0.001 seconds (the plots for 0.001, 0.0005, and 0.0001 are indistinguishable).

Figure 3.6: For a disc rolling on a surface, plots of the reduction of kinetic energy over time caused by approximating the disc as a uniform regular polygon.

This section presents a highly accurate geometrically implicit time-stepping method for convex objects described as an intersection of implicit surfaces. The resulting subproblem at each time-step will be a mixed nonlinear complementarity problem and we call our time-stepping scheme a *geometrically implicit* time-stepping scheme. We assume the objects to be convex objects described as an intersection of implicit surfaces. We first present the method for rigid bodies and then extend it to locally compliant or quasi-rigid bodies (where each body consists of a rigid core surrounded by a thin compliant shell [100, 101, 102, 88]). This method also takes into consideration other important nonlinear elements such as quadratic Coulomb friction. This method will provide a baseline for understanding and quantifying the errors incurred when using a geometrically explicit method and when making various linearizing approximations. Our ultimate goal is to develop techniques for automatically selecting the appropriate method for a given application, and to guide method switching, step size adjustment, and model approximations on the fly.

3.5.1 Contact Constraint for Rigid Bodies

In this section we rewrite the contact condition (Equation (2.45)) as a complementarity condition in the work space, combine it with an optimization problem to find the closest points, and prove that the resultant system of equations ensures that the contact constraints are satisfied. Let us consider the ith contact. For ease of exposition, we first present the case where each object is a convex object described by a single implicit surface. A more general formulation where each object is described by an intersection of implicit surfaces is given in section 3.5.1.3.

3.5.1.1 Objects described by a single convex function

Let the two objects be defined by convex functions $f(\boldsymbol{\xi}_1) \leq 0$ and $g(\boldsymbol{\xi}_2) \leq 0$ respectively, where $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ are the coordinates of points in the two objects. Let $\mathbf{a}_1$ and $\mathbf{a}_2$ be the closest points on the two objects. The equation of an implicit surface has the property that for any point $\mathbf{x}$, the point lies inside the object for $f(\mathbf{x}) < 0$, on the object surface for $f(\mathbf{x}) = 0$, and outside the object for $f(\mathbf{x}) > 0$. Thus, we can define the gap function in the work space as either $f(\mathbf{a}_2)$ or $g(\mathbf{a}_1)$ and write the complementarity conditions as either one of the following two conditions:

$$\begin{aligned} 0 \leq \lambda_{in} \perp f(\mathbf{a}_2) \geq 0 \\ 0 \leq \lambda_{in} \perp g(\mathbf{a}_1) \geq 0 \end{aligned} \tag{3.69}$$

where $\mathbf{a}_1$ and $\mathbf{a}_2$ are given by a solution to the following minimization problem:

$$(\mathbf{a}_1,\ \mathbf{a}_2) = \arg\min_{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2} \left\{ \frac{1}{2} \|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|^2 : f(\boldsymbol{\xi}_1) \leq 0,\quad g(\boldsymbol{\xi}_2) \leq 0 \right\}. \tag{3.70}$$

The Karush-Kuhn-Tucker (KKT) optimality conditions of Equation (3.70) that the solutions $\mathbf{a}_1$ and $\mathbf{a}_2$ must satisfy are given by the following system of algebraic equations:

$$\begin{aligned}
\mathbf{a}_1 - \mathbf{a}_2 &= -\hat{l}_1 \nabla f(\mathbf{a}_1) \\
\mathbf{a}_1 - \mathbf{a}_2 &= \hat{l}_2 \nabla g(\mathbf{a}_2) \\
f(\mathbf{a}_1) + s_1 &= 0 \\
g(\mathbf{a}_2) + s_2 &= 0 \\
0 \leq \hat{l}_1 &\perp s_1 \geq 0 \\
0 \leq \hat{l}_2 &\perp s_2 \geq 0
\end{aligned} \tag{3.71}$$

where $\hat{l}_1$ and $\hat{l}_2$ are the Lagrange multipliers and s_1 and s_2 are the slack variables. After eliminating the slack variables, the system of equations (3.71) can be rewritten as the following equivalent system:

$$\mathbf{a}_1 - \mathbf{a}_2 = -\hat{l}_1 \nabla f(\mathbf{a}_1) \tag{3.72}$$

$$\hat{l}_1 \nabla f(\mathbf{a}_1) = -\hat{l}_2 \nabla g(\mathbf{a}_2) \tag{3.73}$$

$$0 \leq \hat{l}_1 \perp -f(\mathbf{a}_1) \geq 0 \tag{3.74}$$

$$0 \leq \hat{l}_2 \perp -g(\mathbf{a}_2) \geq 0. \tag{3.75}$$

The geometric meaning of equations (3.72) and (3.73) is that the normals to the two surfaces at their closest points are parallel to the line joining the closest points. The solution to the system of equations (3.72) to (3.75) gives the closest point when the two objects are separate. However, when $\mathbf{a}_1 = \mathbf{a}_2$, both Lagrange multipliers are forced to become 0 (since both $|\nabla f(\mathbf{a}_1)| > 0$ and $|\nabla g(\mathbf{a}_2)| > 0$) and the normals to the two surfaces at the contact point are no longer constrained to be parallel. Because of this degeneracy, intersection points of two interpenetrating objects satisfy equations (3.72)–(3.75) and the complementarity conditions in equation (3.69) (see figure 3.7). Thus, as written, equations (3.72)–(3.75) and (3.69) do not guarantee non-penetration. We want to form a system of equations that is equivalent to the KKT conditions (equations (3.72)–(3.75)) when the distance between the objects is non-zero but only gives the touching solution when the distance is zero (middle case in figure 3.7 with parallel surface normals).

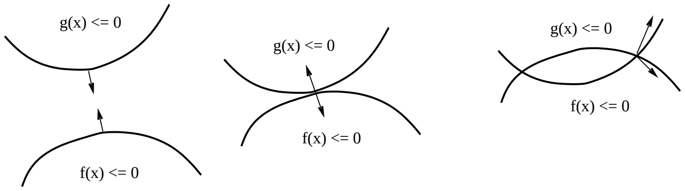


Figure 3.7: Three Contact cases: (left) objects are separate, (middle) objects are touching, (right) objects are intersecting.

Proposition 1. *Equations* (3.76)–(3.79) *are equivalent to the KKT conditions when the distance between the objects is non-zero. Moreover, when combined with either complementarity condition in equation* (3.80)*, it gives only the touching solution when the distance between the objects is zero.*

$$\mathbf{a}_1 - \mathbf{a}_2 = -l_1 \nabla f(\mathbf{a}_1) \tag{3.76}$$

$$\nabla f(\mathbf{a}_1) = -l_2 \nabla g(\mathbf{a}_2) \tag{3.77}$$

$$0 \leq l_1 \perp -f(\mathbf{a}_1) \geq 0 \tag{3.78}$$

$$0 \leq l_2 \perp -g(\mathbf{a}_2) \geq 0 \tag{3.79}$$

$$0 \leq \lambda_{in} \perp f(\mathbf{a}_2) \geq 0 \tag{3.80}$$

Proof of Proposition 1. Let us first consider the case when the distance between the objects is greater than zero. In this case $\mathbf{a}_1 \neq \mathbf{a}_2$. Since the gradient vectors cannot be zero and $\hat{l}_1, \hat{l}_2$ in equation (3.74), (3.75) are constrained to be non-negative, $\hat{l}_1, \hat{l}_2$ are strictly positive (or non-zero) in this case (from equations (3.72) and (3.73)). Therefore equation (3.73) can be written as $\nabla f(\mathbf{a}_1) = -\frac{\hat{l}_2}{\hat{l}_1} \nabla g(\mathbf{a}_2)$. Using $l_1 = \hat{l}_1$ and $l_2 = \frac{\hat{l}_2}{\hat{l}_1}$, the equations (3.72) to (3.74) can be written as equations (3.76) to (3.78). Since $\hat{l}_1 > 0$ we can rewrite the complementarity condition in equation (3.75) as

$$\begin{aligned} &\hat{l}_2(-g(\mathbf{a}_2)) = 0, \quad \hat{l}_2 \geq 0, \quad -g(\mathbf{a}_2) \geq 0 \\ \Leftrightarrow\ &\frac{\hat{l}_2}{\hat{l}_1}(-g(\mathbf{a}_2)) = 0 \quad \frac{\hat{l}_2}{\hat{l}_1} \geq 0, \quad -g(\mathbf{a}_2) \geq 0 \\ \Leftrightarrow\ &l_2(-g(\mathbf{a}_2)) = 0 \quad l_2 \geq 0, \quad -g(\mathbf{a}_2) \geq 0 \end{aligned} \tag{3.81}$$

Thus equations (3.76) to (3.79) are equivalent to equations (3.72) to (3.75) when the distance between the objects is greater than zero.

When the distance between the objects is equal to zero, $\mathbf{a}_1 = \mathbf{a}_2$. Thus $l_1 = 0$ from equation (3.76) which implies $f(\mathbf{a}_1) \leq 0$ from equation (3.78). Equation (3.77) implies that $l_2 > 0$ and that gives $g(\mathbf{a}_2) = 0$ from equation (3.79). Thus the system (3.76) to (3.79) gives solution points that lie on the surface of the object defined by function g and that lie inside or on the surface of the object defined by function f with the gradients of the functions defining the surfaces in opposite directions at the common point. However, equation (3.80) implies that the point $\mathbf{a}_2$ cannot lie within the defined by function f. Hence the only possible solution is that $\mathbf{a}_1 = \mathbf{a}_2$ with the normals to the two surfaces at the solution point parallel. This disallows the interpenetrating case, and we are left with only the touching solution. □

Proposition 2. *Equations* (3.76) *to* (3.80) *together represent the contact constraints, i.e., the two objects will satisfy the contact constraints at the end of each time step if and only if equations* (3.76) *to* (3.80) *hold together.*

Proof. As discussed above. □

3.5.1.2 New Discrete Time Model

We can now rewrite equation (2.77) as a mixed NCP for the geometrically-implicit time-stepper. The vector of unknowns, $\mathbf{z}$, can be partitioned into $\mathbf{z} = [\mathbf{u},\ \mathbf{v}]$ where $\mathbf{u} = [\boldsymbol{\nu},\ \mathbf{a}_1,\ \mathbf{a}_2,\ \mathbf{p}_{\mathrm{t}},\ \mathbf{p}_{\mathrm{o}},\ \mathbf{p}_{\mathrm{r}}]$ and $\mathbf{v} = [\mathbf{l},\ \mathbf{p}_{\mathrm{n}},\ \boldsymbol{\sigma}]$. The equality constraints in the mixed NCP are:

$$\begin{aligned}
0 &= -\mathbf{M}\boldsymbol{\nu}^{\ell+1} + \mathbf{M}\boldsymbol{\nu}^{\ell} + \mathbf{W}_{\mathrm{n}}^{\ell+1}\mathbf{p}_{\mathrm{n}}^{\ell+1} + \mathbf{W}_{\mathrm{t}}^{\ell+1}\mathbf{p}_{\mathrm{t}}^{\ell+1} + \mathbf{W}_{\mathrm{o}}^{\ell+1}\mathbf{p}_{\mathrm{o}}^{\ell+1} + \mathbf{W}_{\mathrm{r}}^{\ell+1}\mathbf{p}_{\mathrm{r}}^{\ell+1} \\
&\qquad + \mathbf{p}_{\mathrm{app}}^{\ell} + \mathbf{p}_{\mathrm{vp}}^{\ell} \\
0 &= (\mathbf{a}_1^{\ell+1} - \mathbf{a}_2^{\ell+1}) + l_1 \nabla f(\mathbf{a}_1^{\ell+1}) \\
0 &= \nabla f(\mathbf{a}_1^{\ell+1}) + l_2 \nabla g(\mathbf{a}_2^{\ell+1}) \qquad (3.82) \\
0 &= \mathbf{E}_{\mathrm{t}}^2 \mathbf{U}\mathbf{p}_n^{\ell+1} \circ (\mathbf{W}_{\mathrm{t}}^T)^{\ell+1}\boldsymbol{\nu}^{\ell+1} + \mathbf{p}_{\mathrm{t}}^{\ell+1} \circ \boldsymbol{\sigma}^{\ell+1} \\
0 &= \mathbf{E}_{\mathrm{o}}^2 \mathbf{U}\mathbf{p}_n^{\ell+1} \circ (\mathbf{W}_{\mathrm{o}}^T)^{\ell+1}\boldsymbol{\nu}^{\ell+1} + \mathbf{p}_{\mathrm{o}}^{\ell+1} \circ \boldsymbol{\sigma}^{\ell+1} \\
0 &= \mathbf{E}_{\mathrm{r}}^2 \mathbf{U}\mathbf{p}_n^{\ell+1} \circ (\mathbf{W}_{\mathrm{r}}^T)^{\ell+1}\boldsymbol{\nu}^{\ell+1} + \mathbf{p}_{\mathrm{r}}^{\ell+1} \circ \boldsymbol{\sigma}^{\ell+1}.
\end{aligned}$$

The complementarity constraints on $\mathbf{v}$ are:

$$0 \leq \begin{bmatrix} l_1 \\ l_2 \\ \mathbf{p}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix} \perp \begin{bmatrix} f(\mathbf{a}_1^{\ell+1}) \\ g(\mathbf{a}_2^{\ell+1}) \\ f(\mathbf{a}_2^{\ell+1}) \\ \boldsymbol{\zeta} \end{bmatrix} \geq 0 \tag{3.83}$$

where

$$\begin{aligned}\boldsymbol{\zeta} = \mathbf{U}\mathbf{p}_{\mathrm{n}}^{\ell+1} \circ \mathbf{U}\mathbf{p}_{\mathrm{n}}^{\ell+1} - \left(\mathbf{E}_{\mathrm{t}}^2\right)^{-1}\left(\mathbf{p}_{\mathrm{t}}^{\ell+1} \circ \mathbf{p}_{\mathrm{t}}^{\ell+1}\right) - \left(\mathbf{E}_{\mathrm{o}}^2\right)^{-1}\left(\mathbf{p}_{\mathrm{o}}^{\ell+1} \circ \mathbf{p}_{\mathrm{o}}^{\ell+1}\right) \\ - \left(\mathbf{E}_{\mathrm{r}}^2\right)^{-1}\left(\mathbf{p}_{\mathrm{r}}^{\ell+1} \circ \mathbf{p}_{\mathrm{r}}^{\ell+1}\right).\end{aligned}$$

In the above formulation, we see $\mathbf{u} \in \mathbb{R}^{6n_b+9n_c}$, $\mathbf{v} \in \mathbb{R}^{2n_c}$, the vector function of equality constraints maps $[\mathbf{u}, \mathbf{v}]$ to $\mathbb{R}^{6n_b+9n_c}$ and the vector function of complementarity constraints maps $[\mathbf{u}, \mathbf{v}]$ to $\mathbb{R}^{2n_c}$ where n_b and n_c are the number of bodies and number of contacts respectively. If using convex bodies only, the upper bound on the number of contacts can be determined directly from the number of bodies, $n_c = \sum_{i=1}^{n_b-1} i$.

3.5.1.3 Objects described by intersections of convex functions

We present here the contact conditions for the general case where each convex object is defined as an intersection of convex inequalities. Let $f_j(\boldsymbol{\xi}_1) \leq 0, j = 1, \ldots, m$, $g_j(\boldsymbol{\xi}_2) \leq 0, j = m+1, \ldots, n$, be convex functions representing the two convex objects. Since the closest point is outside the object if it is outside at least one of the intersecting surfaces, the complementarity conditions for nonpenetration can be written as either one of the following two sets of conditions:

$$\begin{aligned} 0 \leq \lambda_{in} \perp \max\{f_j(\mathbf{a}_2)\} \geq 0 \quad j = 1, \ldots m \\ 0 \leq \lambda_{in} \perp \max\{g_j(\mathbf{a}_1)\} \geq 0 \quad j = m+1, \ldots n \end{aligned} \tag{3.84}$$

where $\mathbf{a}_1$ and $\mathbf{a}_2$ are the closest points on the two bodies and are given by the KKT conditions

$$\begin{aligned}
\mathbf{a}_1 - \mathbf{a}_2 &= -\sum_{i=1}^{m} \hat{l}_i \nabla f_i(\mathbf{a}_1) \\
\mathbf{a}_1 - \mathbf{a}_2 &= \sum_{j=m+1}^{n} \hat{l}_j \nabla g_j(\mathbf{a}_2) \\
0 \leq \hat{l}_i &\perp -f_i(\mathbf{a}_1) \geq 0 \\
0 \leq \hat{l}_j &\perp -g_j(\mathbf{a}_2) \geq 0
\end{aligned} \tag{3.85}$$

At the optimal solution only some of the constraints are active. Let $\mathbb{I}$ be the index set of active constraints. Thus $\hat{l}_k = 0$, if $k \notin \mathbb{I}$ and the KKT optimality conditions can be written as the following set of nonlinear equations:

$$\begin{aligned}
\mathbf{a}_1 - \mathbf{a}_2 &= -\sum_{i \in \mathbb{I} \cap \{i\}} \hat{l}_i \nabla f_i(\mathbf{a}_1) \\
\sum_{i \in \mathbb{I} \cap \{i\}} \hat{l}_i \nabla f_i(\mathbf{a}_1) &= -\sum_{j \in \mathbb{I} \cap \{j\}} \hat{l}_j \nabla g_j(\mathbf{a}_2) \\
0 \leq \hat{l}_i &\perp -f_i(\mathbf{a}_1 \geq 0 \\
0 \leq \hat{l}_j &\perp -g_j(\mathbf{a}_2) \geq 0
\end{aligned} \tag{3.86}$$

Equations (3.84) and (3.86) together represent the contact constraints as long as $\mathbf{a}_1 \neq \mathbf{a}_2$. Using arguments similar to the single surface case in Section IV we can see that it is not possible to distinguish between touching points and intersecting points using the above formulation. In this case also, we can rewrite Equation (3.86) suitably by dividing throughout by one of the non-zero Lagrange multipliers to eliminate the intersection point solutions.

$$\begin{aligned}
\mathbf{a}_1 - \mathbf{a}_2 &= -(\nabla f_{k_1}(\mathbf{a}_1) + \sum_{k \in \{\mathbb{I} \setminus k_1\} \cap \{i\}} l_k \nabla f_k(\mathbf{a}_1)) \\
\nabla f_{k_1}(\mathbf{a}_1) + \sum_{k \in \{\mathbb{I} \setminus k_1\} \cap \{i\}} l_k \nabla f_k(\mathbf{a}_1) &= -\sum_{k \in \mathbb{I} \cap \{j\}} l_k \nabla g_k(\mathbf{a}_2) \\
0 \leq \hat{l}_i \perp -f_i(\mathbf{a}_1 \geq 0 & \\
0 \leq \hat{l}_j \perp -g_j(\mathbf{a}_2) \geq 0 &
\end{aligned} \tag{3.87}$$

Proposition 3. *Equation* (3.84) *and* (3.87) *together represent the nonpenetration constraints, i.e., the two objects will satisfy the contact constraints at the end of each time step if and only if Equation* (3.84) *and* (3.87) *hold together.*

3.5.2 Contact Constraints for Compliant Bodies

In this section we describe the 3D linear viscoelastic model of contact [63] and modify our contact constraints to include the deflections at the contact. We incorporate this model in our time-stepping scheme and present the mixed NCP problem that we are solving at each time step. We extend the Kelvin-Voigt model with the physically motivated observation that the deformations in the normal direction are bounded by some maximum value. For example, a human finger has a thin compliant layer of muscle and tissue surrounding the rigid core (bone). The application of a force on the finger results in a deformation of the thin compliant layer until the rigid core is reached, at which point the non-penetration response is rigid. Therefore, our model allows for a maximum possible deflection, beyond which the contact behaves as a rigid body contact. The linear model can be replaced by a nonlinear model but this comes at the cost of more unknown modeling parameters to be determined experimentally. For simplicity of exposition, we consider only one of the objects to be flexible at each contact. The general formulation where both the bodies are flexible will contain the additional constraint that the contact forces acting on both the bodies have to be equal. For each contact i, the normal impact force $\lambda_{i\mathrm{n}}$ is the sum of two components:

$$\lambda_{i\mathrm{n}} = \lambda_{i\mathrm{nr}} + \lambda_{i\mathrm{ns}} \tag{3.88}$$

where $\lambda_{i\mathrm{ns}}$ is the component of the force that is obtained from the deformation of the spring and $\lambda_{i\mathrm{nr}}$ is the component from impact with the rigid core. The tangential force at each contact, $\lambda_{i\mathrm{f}} = [\lambda_{i\mathrm{t}}, \lambda_{i\mathrm{o}}]$ is also given by a linear spring-damper model. However, we do not have a bound on the maximum displacement in the tangential direction. Concatenating all the individual force components into vectors allows us to write for each contact (we drop subscript i for legibility), $\boldsymbol{\lambda} = \mathbf{K}\boldsymbol{\delta} + \mathbf{C}\dot{\boldsymbol{\delta}}$, where $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_{\mathrm{ns}}, \boldsymbol{\lambda}_{\mathrm{t}}, \boldsymbol{\lambda}_{\mathrm{o}}]$ and $\boldsymbol{\delta} = [\delta_{\mathrm{n}}, \delta_{\mathrm{t}}, \delta_{\mathrm{o}}]$ are 3×1 column vectors with $\boldsymbol{\delta}_{\mathrm{n}}, \boldsymbol{\delta}_{\mathrm{t}}, \boldsymbol{\delta}_{\mathrm{o}}$ being the normal and tangential deflections respectively. The matrices $\mathbf{K}$, $\mathbf{C}$ are stiffness and damping matrices given by

$$\mathbf{K} = \begin{bmatrix} K_{\mathrm{nn}} & K_{\mathrm{nt}} & K_{\mathrm{no}} \\ K_{\mathrm{tn}} & K_{\mathrm{tt}} & K_{\mathrm{to}} \\ K_{\mathrm{on}} & K_{\mathrm{ot}} & K_{\mathrm{oo}} \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} C_{\mathrm{nn}} & C_{\mathrm{nt}} & C_{\mathrm{no}} \\ C_{\mathrm{tn}} & C_{\mathrm{tt}} & C_{\mathrm{to}} \\ C_{\mathrm{on}} & C_{\mathrm{ot}} & C_{\mathrm{oo}} \end{bmatrix}$$

For systems with multiple contact, the contact forces $\boldsymbol{\lambda}$, and body deformations $\boldsymbol{\delta}$ become concatenations of n_c subvectors, where n_c is the number of contacts. The stiffness and damping matrices are block diagonal matrices of size $3n_c \times 3n_c$, where each diagonal block of size 3×3 represent one contact.

3.5.2.1 Objects described by a single convex function

When we consider contact compliance the contact constraints in Section 3 need to be modified to take into account the deflection $\boldsymbol{\delta}$. We denote the maximum normal deflection by $\delta_{\mathrm{n}}^{\mathrm{o}} > 0$ and assume that it will be determined experimentally. Figure (3.8) shows two objects in contact with each other. The bold line shows the deformed shapes of the two objects. The point of contact is the point where the virtual objects shown by dotted lines touch. The deflections of the two objects along the normal at the contact point are $\delta_{\mathrm{n}1}$ and $\delta_{\mathrm{n}2}$ respectively. In the subsequent discussion, we will assume $\delta_{\mathrm{n}1} = 0$ for simplicity and drop the subscript 2 from $\delta_{\mathrm{n}2}$. Therefore, the constraints for the closest points are given by

$$\begin{aligned} \mathbf{a}_1 - \mathbf{a}_2 &= -l_1 \nabla f(\mathbf{a}_1) \\ \nabla f(\mathbf{a}_1) &= -l_2 \nabla g(\mathbf{a}_2) \\ 0 \leq l_1 &\perp -f(\mathbf{a}_1) \geq 0 \\ 0 \leq l_2 &\perp -(g(\mathbf{a}_2) + \bar{\delta}_{\mathrm{n}}) \geq 0 \end{aligned} \tag{3.89}$$

where $\bar{\boldsymbol{\delta}}_{\mathrm{n}}$ is the *algebraic distance*. However, the normal contact force is given in terms of the Euclidean deflection. To obtain the Euclidean deflection from this algebraic deflection we note that the Euclidean deflection is the distance between the point $\mathbf{a}_2$ and the point where the normal to $g(\boldsymbol{\xi}_2) + \delta_{\mathrm{n}2} = 0$ at $\mathbf{a}_2$ intersects

$g(\boldsymbol{\xi}_2) = 0$. From the above argument it can be seen that

$$g\left(\mathbf{a}_2 + \delta_n \frac{\nabla g(\mathbf{a}_2)}{\|\nabla g(\mathbf{a}_2)\|}\right) = 0 \tag{3.90}$$

The complementarity conditions in Equation (3.69) thus becomes:

$$\begin{aligned} &0 \leq \boldsymbol{\lambda}_{\mathrm{ns}} \perp \psi(\mathbf{a}_1, \mathbf{a}_2) + \bar{\delta}_{\mathrm{n}} \geq 0 \\ &0 \leq \boldsymbol{\lambda}_{\mathrm{nr}} \perp \boldsymbol{\delta}_{\mathrm{n}}^{\mathrm{o}} - \delta_{\mathrm{n}} \geq 0 \end{aligned} \tag{3.91}$$

where $\psi(\mathbf{a}_1, \mathbf{a}_2) = f(\mathbf{a_2})$ or $g(\mathbf{a_1})$ for implicit surfaces. When the two bodies are not in contact the right hand side of both the complementarity constraints are positive and hence we do not have any contact force. The above system of equations are to be written for each of the contacting bodies.

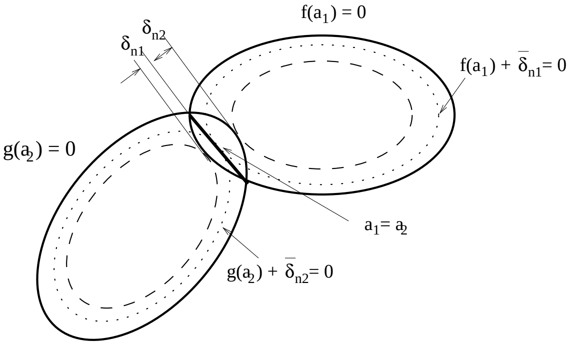

$f(a_1) = 0$
δ_{n2}
δ_{n1}
$f(a_1) + \bar{\delta}_{n1} = 0$
$g(a_2) = 0$
$a_1 = a_2$
$g(a_2) + \bar{\delta}_{n2} = 0$

Figure 3.8: Schematic representation of the deflection at contact. The contact is where the dotted curves touch.

This formulation ensures that we satisfy the contact constraints at the end of the time step taking into consideration the possibility of the deflection of the body. It does not require the computation of penetration depth for obtaining the deflection as required in [102]. It ensures that we get a collision response in a fixed time-step scheme.

We can now formulate the mixed NCP for the geometrically-implicit lumped compliant contact time-stepper. The vector of unknowns $\mathbf{z}$ can be partitioned into

$\mathbf{z} = [\mathbf{u},\ \mathbf{v}]$ where

$$\begin{aligned}
\mathbf{u} &= [\boldsymbol{\nu},\ \mathbf{a}_1,\ \mathbf{a}_2,\ \bar{\boldsymbol{\delta}}_{\mathrm{n}},\ \boldsymbol{\delta}_{\mathrm{n}},\ \boldsymbol{\delta}_{\mathrm{t}},\ \boldsymbol{\delta}_{\mathrm{o}},\ \mathbf{p}_{\mathrm{t}},\ \mathbf{p}_{\mathrm{o}},\ \mathbf{p}_{\mathrm{r}}] \\
\mathbf{v} &= [\mathbf{l},\ \mathbf{p}_{\mathrm{ns}},\ \mathbf{p}_{\mathrm{nr}},\ \boldsymbol{\sigma}].
\end{aligned}$$

The equality constraints in the mixed NCP are:

$$\begin{aligned}
0 &= -\mathbf{M}\boldsymbol{\nu}^{\ell+1} + \mathbf{M}\boldsymbol{\nu}^{\ell} + \mathbf{W}_{\mathrm{n}}^{\ell+1}\mathbf{p}_{\mathrm{n}}^{\ell+1} + \mathbf{W}_{\mathrm{t}}^{\ell+1}\mathbf{p}_{\mathrm{t}}^{\ell+1} + \mathbf{W}_{\mathrm{o}}^{\ell+1}\mathbf{p}_{\mathrm{o}}^{\ell+1} + \mathbf{W}_{\mathrm{r}}^{\ell+1}\mathbf{p}_{\mathrm{r}}^{\ell+1} \\
&\qquad + \mathbf{p}_{\mathrm{app}}^{\ell} + \mathbf{p}_{\mathrm{vp}}^{\ell} \\
0 &= \mathbf{p}^{\ell+1} - \big(h\mathbf{K}\boldsymbol{\delta}^{\ell+1} + \mathbf{C}(\boldsymbol{\delta}^{\ell+1} - \boldsymbol{\delta}^{\ell})\big) \\
0 &= (\mathbf{a}_1^{\ell+1} - \mathbf{a}_2^{\ell+1}) + l_1\nabla f(\mathbf{a}_1^{\ell+1}) \\
0 &= \nabla f(\mathbf{a}_1^{\ell+1}) + l_2\nabla g(\mathbf{a}_2^{\ell+1}) \\
0 &= g(\mathbf{a}_2^{\ell+1} + \delta_n^{\ell+1}\frac{\nabla g(\mathbf{a}_2^{\ell+1})}{\|\nabla g(\mathbf{a}_2^{\ell+1})\|}) \\
0 &= \mathbf{E}_{\mathrm{t}}^2\mathbf{U}\mathbf{p}_n^{\ell+1} \circ (\mathbf{W}_{\mathrm{t}}^T)^{\ell+1}\boldsymbol{\nu}^{\ell+1} + \mathbf{p}_{\mathrm{t}}^{\ell+1} \circ \boldsymbol{\sigma}^{\ell+1} \\
0 &= \mathbf{E}_{\mathrm{o}}^2\mathbf{U}\mathbf{p}_n^{\ell+1} \circ (\mathbf{W}_{\mathrm{o}}^T)^{\ell+1}\boldsymbol{\nu}^{\ell+1} + \mathbf{p}_{\mathrm{o}}^{\ell+1} \circ \boldsymbol{\sigma}^{\ell+1} \\
0 &= \mathbf{E}_{\mathrm{r}}^2\mathbf{U}\mathbf{p}_n^{\ell+1} \circ (\mathbf{W}_{\mathrm{r}}^T)^{\ell+1}\boldsymbol{\nu}^{\ell+1} + \mathbf{p}_{\mathrm{r}}^{\ell+1} \circ \boldsymbol{\sigma}^{\ell+1}
\end{aligned} \tag{3.92}$$

where $\mathbf{p}^{\ell+1} = [\mathbf{p}_{\mathrm{n}}^{\ell+1}\ \mathbf{p}_{\mathrm{t}}^{\ell+1}\ \mathbf{p}_{\mathrm{o}}^{\ell+1}]^T$, $\mathbf{p}_{\mathrm{n}}^{\ell+1} = \mathbf{p}_{\mathrm{ns}}^{\ell+1} + \mathbf{p}_{\mathrm{nr}}^{\ell+1}$ and $\boldsymbol{\delta}^{\ell+1} = [\boldsymbol{\delta}_{\mathrm{n}}^{\ell+1}\ \boldsymbol{\delta}_{\mathrm{t}}^{\ell+1}\ \boldsymbol{\delta}_{\mathrm{o}}^{\ell+1}]^T$.

The complementarity constraints on $\mathbf{v}$ are:

$$0 \le \begin{bmatrix} l_1 \\ l_2 \\ \mathbf{p}_{\mathrm{ns}}^{\ell+1} \\ \mathbf{p}_{\mathrm{nr}}^{\ell+1} \\ \boldsymbol{\sigma}^{\ell+1} \end{bmatrix} \perp \begin{bmatrix} -f(\mathbf{a}_1^{\ell+1}) \\ -(g(\mathbf{a}_2^{\ell+1}) + \bar{\boldsymbol{\delta}}_{\mathrm{n}}^{\ell+1}) \\ \psi(\mathbf{a}_1^{\ell+1},\ \mathbf{a}_2^{\ell+1}) + \bar{\boldsymbol{\delta}}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\delta}_{\mathrm{n}}^{\mathrm{o}} - \boldsymbol{\delta}_{\mathrm{n}}^{\ell+1} \\ \boldsymbol{\zeta} \end{bmatrix} \ge 0 \tag{3.93}$$

where

$$\begin{aligned}
\boldsymbol{\zeta} = \mathbf{U}\mathbf{p}_{\mathrm{n}}^{\ell+1} \circ \mathbf{U}\mathbf{p}_{\mathrm{n}}^{\ell+1} &- \left(\mathbf{E}_{\mathrm{t}}^2\right)^{-1}\left(\mathbf{p}_{\mathrm{t}}^{\ell+1} \circ \mathbf{p}_{\mathrm{t}}^{\ell+1}\right) - \left(\mathbf{E}_{\mathrm{o}}^2\right)^{-1}\left(\mathbf{p}_{\mathrm{o}}^{\ell+1} \circ \mathbf{p}_{\mathrm{o}}^{\ell+1}\right) \\
&- \left(\mathbf{E}_{\mathrm{r}}^2\right)^{-1}\left(\mathbf{p}_{\mathrm{r}}^{\ell+1} \circ \mathbf{p}_{\mathrm{r}}^{\ell+1}\right)
\end{aligned}$$

In the above formulation, we see $u \in \mathbb{R}^{6n_b+13n_c}$, $v \in \mathbb{R}^{3n_c}$, the vector function of equality constraints maps $[u, v]$ to $\mathbb{R}^{6n_b+13n_c}$ and the vector function of complementarity constraints maps $[u, v]$ to $\mathbb{R}^{3n_c}$ where n_b and n_c are the number of bodies and number of contacts respectively. If using convex bodies only, the number of contacts can be determined directly from the number of bodies, $n_c = \sum_{i=1}^{n_b} i$.

3.5.2.2 Objects described by intersections of convex functions

Using the results obtained in Section 3.5.1.3 and the previous section, we can now formulate the contact constraints for quasi-rigid or locally compliant bodies where each body is represented as an intersection of convex implicit surfaces. In this case the complementarity conditions for nonpenetration can be written as either one of the following two sets of conditions:

$$\begin{aligned} &0 \leq \lambda_{in} \perp \max\{f_j(\mathbf{a}_2) + \bar{\delta}_{\mathrm{jn}}\} \geq 0 \quad j = 1, \dots m \\ &0 \leq \lambda_{in} \perp \max\{g_j(\mathbf{a}_1) + \bar{\delta}_{\mathrm{jn}}\} \geq 0 \quad j = m+1, \dots n \end{aligned} \tag{3.94}$$

Moreover, the closest points on the two objects are given by

$$\begin{aligned} \mathbf{a}_1 - \mathbf{a}_2 &= -(\nabla f_{k_1}(\mathbf{a}_1) + \sum_{k \in \{\mathbb{I} \setminus k_1\} \cap \{i\}} l_k \nabla f_k(\mathbf{a}_1)) \\ \nabla f_{k_1}(\mathbf{a}_1) + \sum_{k \in \{\mathbb{I} \setminus k_1\} \cap \{i\}} l_k \nabla f_k(\mathbf{a}_1) &= - \sum_{k \in \{\mathbb{I} \setminus k_2\} \cap \{j\}} l_k \nabla g_k(\mathbf{a}_2) \\ 0 \leq l_i \perp -f_i(\mathbf{a}_1) &\geq 0 \\ 0 \leq l_j - (g_j(\mathbf{a}_2) + \bar{\delta}_{j\mathrm{n}}) &= 0 \end{aligned} \tag{3.95}$$

which is a modification of Equation (3.87) and the notation is identical to the discussion in Section 3.5.1.3. To determine the actual deflection from the algebraic distance, we need to assume that the normal at the point of contact is well defined. Assuming this to be true, we can then obtain the deflection using Equation (3.90) and the forces on the two bodies can be obtained using the appropriate constitutive laws.

3.5.3 Illustrative Examples

In this section we present examples to validate our technique against known analytical results and previous approaches. The first example is for rigid bodies and is the same example of a disc rolling without slip on a plane that we studied in Section 3.5. The seconds example consists of a small sphere rolling in contact with two larger spheres, and validates our results against previous simulations of this system. The third example is for quasi-rigid bodies and consists of a unit disc falling on a half-plane with frictionless contact. This example gives a simple validation of our method. The final example is to repeat, in simulation, 100,800 grasping experiments done by Brost and Christensen in a laboratory at Sandia National Laboratories over a period of weeks, with considerable technician support and dedicated robotics equipment [20].

3.5.3.1 Example 1: Disc on a Plane

In this example we revisit the unit disc example from Section 3.5. For illustrative purposes, we explain the formulation of the full dynamic model in detail. The normal axis of the contact frame $\hat{\mathbf{n}} = [0, 1]^T$ always points in the inertial y-axis direction and tangential axis $\hat{\mathbf{t}} = [1, 0]^T$ always coincides with the x-direction. The mass matrix, $\mathbf{M}$ is constant and the only force acting on the body is due to gravity. The equation of the disc is given by $f_1(x, y) = (x - q_x)^2 + (y - q_y)^2 - 1$, where $\mathbf{q} = (q_x, q_y)$ is the location of the center of the disc in the inertial frame. Let $\boldsymbol{\nu} = [v_x, v_y, \omega_z]$ be the vector of linear and angular velocities and $\mathbf{a}_1$ be the closest point on body 1 (the disc) to the plane (defined by $y = 0$). Similarly, let $\mathbf{a}_2$ be the closest point on the plane to body 1 ($a_{2y} = 0$ and can be removed from the system of unknowns). Thus we have $\mathbf{M} = \text{diag}(m, m, 0.5m)$, where m is the mass of the disc and $0.5m$ the moment of inertia, $\mathbf{p}_{\text{app}} = [0, -mgh, 0]^T$, with $g = 9.81$, the acceleration due to gravity, and

$$\mathbf{r} = \begin{bmatrix} (a_{1x}^{\ell+1} - q_x) \\ (a_{1y}^{\ell+1} - q_y) \end{bmatrix} \qquad \mathbf{W}_{\mathrm{n}} = \begin{bmatrix} \hat{\mathbf{n}} \\ \mathbf{r} \otimes \hat{\mathbf{n}} \end{bmatrix}$$

$$\mathbf{W}_{\mathrm{t}} = \begin{bmatrix} \hat{\mathbf{t}} \\ \mathbf{r} \otimes \hat{\mathbf{t}} \end{bmatrix} \qquad \nabla_{\mathbf{a}_1} f_1(\mathbf{a}_1^{\ell+1}) = \begin{bmatrix} 2(a_{1x}^{\ell+1} - q_x) \\ 2(a_{1y}^{\ell+1} - q_y) \end{bmatrix}$$

where $\mathbf{r}$ is the vector from the center of gravity of the disc to $\mathbf{a}_1$ and $\otimes$ connotes the 2D analog of the cross product (for two vectors $\mathbf{x} = [x_1, x_2]^T, \mathbf{y} = [y_1, y_2]^T, \mathbf{x} \otimes \mathbf{y} = x_1 y_2 - x_2 y_1$).

Assuming that $y = 0$ is the equation of the ground plane, there are 11 unknowns for this system: $\mathbf{z} = [\boldsymbol{\nu},\ \mathbf{a}_1,\ a_{2_x},\ l_1,\ l_2,\ p_{\mathrm{n}},\ p_{\mathrm{t}},\ \sigma]$. The system of equations for the unit disc is:

$$0 = -\mathbf{M}\boldsymbol{\nu}^{\ell+1} + \mathbf{M}\boldsymbol{\nu}^{\ell} + \mathbf{W}_{\mathrm{n}}^{\ell+1} p_{\mathrm{n}}^{\ell+1} + \mathbf{W}_{\mathrm{t}}^{\ell+1} p_{\mathrm{t}}^{\ell+1} + \mathbf{p}_{\mathrm{app}} \tag{3.96}$$

$$0 = \mathbf{a}_2^{\ell+1} - \mathbf{a}_1^{\ell+1} + l_1 \nabla_{\mathbf{a}_1} f_1(\mathbf{a}_1^{\ell+1}) \tag{3.97}$$

$$0 = \nabla_{\mathbf{a}_1} f_1(\mathbf{a}_1^{\ell+1}) + l_2 \hat{n} \tag{3.98}$$

$$0 \leq l_1 \perp f_1(\mathbf{a}_1^{\ell+1}) \geq 0 \tag{3.99}$$

$$0 \leq p_n \perp f_1(\mathbf{a}_2^{\ell+1}) \geq 0 \tag{3.100}$$

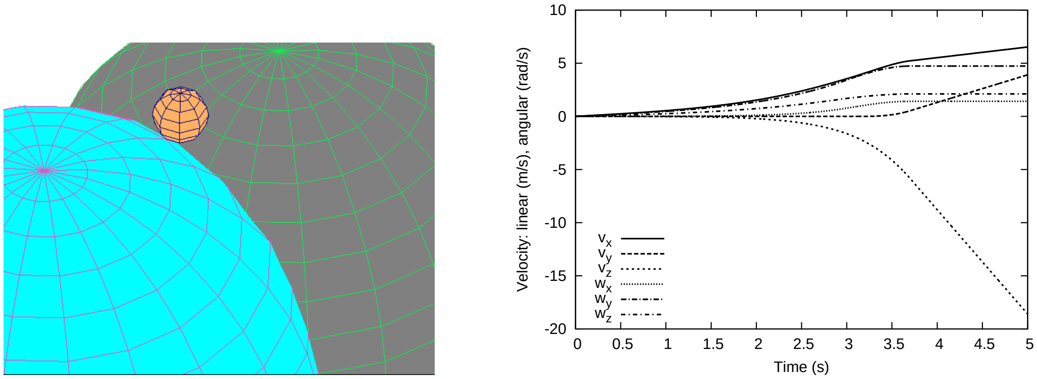
$$0 = \mu p_{\mathrm{n}}^{\ell+1} (\mathbf{W}_{\mathrm{t}}^{\mathbf{T}\ell+1} \boldsymbol{\nu}^{\ell+1}) + \sigma^{\ell+1} p_{\mathrm{t}}^{\ell+1} \tag{3.101}$$

$$0 \leq \sigma \perp \mu^2 p_{\mathrm{n}}^{\ell+1} p_{\mathrm{n}}^{\ell+1} - p_{\mathrm{t}}^{\ell+1} p_{\mathrm{t}}^{\ell+1} \geq 0 \tag{3.102}$$

The initial configuration of the disc is $\mathbf{q} = [0, 1, 0]$, initial velocity is $\boldsymbol{\nu} = [-3, 0, 3]$, mass is $m = 1$, and $\mu = 0.4$. Figure 3.6a shows the kinetic energy of the disc for our implicit representation along with the Stewart-Trinkle LCP implementation using various levels of discretization as it rolls along the horizontal surface. When using an implicit curve representation to model the disc and our new formulation presented above, we get no energy loss (within the numerical tolerance of 10^{-6} used for our simulations) as seen by the horizontal line. When using the LCP formulation we have energy loss as discussed earlier.

3.5.3.2 Example 2: Sphere on Two Spheres

This example consists of a small sphere moving in contact with two larger fixed spheres. This example is chosen to compare the results of our geometrically-implicit method to those presented in [115] and [66] ([115] was able to solve this problem with implicit geometric information because a closed form distance function is available between two spheres). Figure 3.9a shows a small unit sphere in simultaneous contact with two larger fixed spheres. The sphere of radius 10 units is located at $(0,\ 0,\ 0)$ in the inertial frame and the sphere of radius 9 units is located at $(0,\ 11.4,\ 0)$. There is also a constant force of $\lambda_{app} = [1.0,\ 2.6,\ -9.81,\ 0,\ 0,\ 0]^T$ applied to the small sphere. With this force, the sphere initially has one of its contacts rolling while the other contact is simultaneously sliding, the rolling contact transitions to sliding, and both contacts eventually separate. It is important to emphasize that all these transitions are captured using a fixed time step implementation.



(a) A small sphere in contact with two large spheres.

(b) Velocities of small moving sphere

The initial configuration and velocity of the small moving sphere is

$$\mathbf{q} = [0,\ 6.62105263157895,\ 8.78417110772903,\ 1,\ 0,\ 0,\ 0]^T$$
$$\boldsymbol{\nu} = [0,\ 0,\ 0,\ 0,\ 0,\ 0]^T.$$

The friction parameters are: $e_t = 1$, $e_o = 1$, $e_r = 0.3$, and $\mu = 0.2$. There are 32 variables in our NCP formulation (6 velocity variables and 13 variables for each contact: 6 contact point coordinates for the two bodies, 2 Lagrange multipliers, 4 contact impulses and σ for the contact). We used a step size $h = 0.01$ (Tzitzouris-Pang use $h = 0.1$).

The generalized velocity of the sphere is shown in Figure 3.9b. The smooth velocity profile agrees well with the nonlinear Tzitzouris-Pang formulation [115]. The Liu-Wang formulation [66] experienced non-smooth velocity jumps when the small sphere separated from the larger fixed spheres, which they attributed to an explicit time-stepping scheme. In the LCP Stewart-Trinkle implementation, the velocity profiles were very non-smooth. These results further confirm our belief that both linearization and explicit time-stepping lead to inaccuracies.

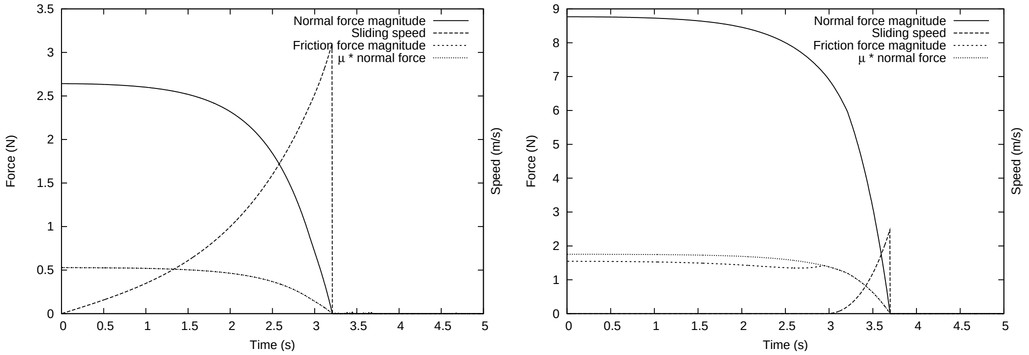
The fidelity of our method is further emphasized by Figures 3.9a and 3.9b that show the forces and sliding speed magnitudes at the two contacts. Contact 1 starts as a sliding contact and we see the sliding speed increases as the normal force decreases. Also, the magnitude of the friction force is equal to $\mu\lambda_{1n}$, consistent with our friction law for a sliding contact. At approximately 3.2 seconds, the small sphere separates from the large sphere at this contact, and all forces acting at contact 1 and the sliding speed drop to zero. Contact 2 on the other hand starts out as a rolling contact until approximately $t = 3$ seconds when it transitions to sliding. During the rolling phase the frictional magnitude is bounded by $\mu\lambda_{2n}$ as required by the friction law, and the sliding speed is 0. At the transition to sliding, the magnitude of the friction force becomes equal to $\mu\lambda_{2n}$ and the sliding speed begins to increase. Finally, at approximately $t = 3.6$ seconds, the contact breaks and all forces at this contact and the sliding speed drop to zero.

3.5.3.3 Example 3: Disc falling on a compliant half-plane

In this example, we simulate a rigid unit disc falling onto a compliant horizontal half-plane. The contact is modeled as a single frictionless contact with no damping. Depending on the value of maximum deflection, the disc may or may not make contact with the rigid core of the half-plane. Figure 3.10 illustrates the problem.

There are 12 unknowns in this system, with 4 complementarity constraints:

$$\mathbf{z} = [\mathbf{u}, \mathbf{v}] = [v_x, v_y, \omega, a_{1x}, a_{1y}, a_{2x}, a_{2y}, l_1, l_2, \delta_n, p_{\mathrm{ns}}, p_{\mathrm{nr}}].$$



(a) Force and sliding speed at contact 1 (b) Force and sliding speed at contact 2

Figure 3.9: Contact 1 is always sliding until separation, hence the μ normal force curve and friction magnitude curve overlap for the duration. The value of $\mu = 0.2$

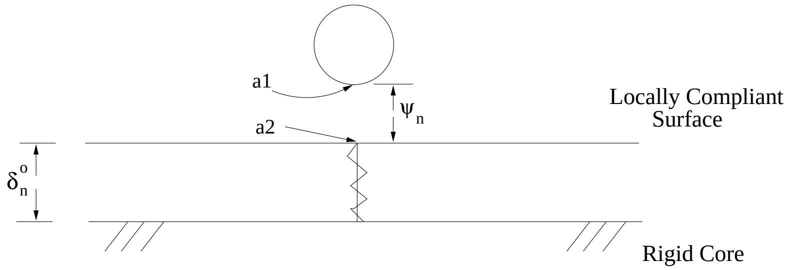


Figure 3.10: Unit disc falling onto a frictionless compliant surface

The equations of motion for this system are (omitted superscripts indicate time ℓ, except for the Lagrange multipliers l_1 and l_2 that are always evaluated at $\ell + 1$):

$$0 = -\mathbf{M}\boldsymbol{\nu}^{\ell+1} + \mathbf{M}\boldsymbol{\nu} + \mathbf{W}_{\mathrm{n}}^{\ell+1} p_{\mathrm{ns}}^{\ell+1} + \mathbf{W}_{\mathrm{n}}^{\ell+1} p_{\mathrm{nr}}^{\ell+1} + \mathbf{p}_{\mathrm{app}} \tag{3.103}$$

$$0 = p_{\mathrm{ns}}^{\ell+1} - hk\delta_n^{\ell+1} \tag{3.104}$$

$$0 = \mathbf{a}_2^{\ell+1} - \mathbf{a}_1^{\ell+1} + l_1 \hat{\mathbf{n}} \tag{3.105}$$

$$0 = l_2 \nabla_{\mathbf{a}_1} f_1(\mathbf{a}_1^{\ell+1}) + \hat{\mathbf{n}} \tag{3.106}$$

$$0 \leq l_2 \perp f_1(\mathbf{a}_1^{\ell+1}) \geq 0 \tag{3.107}$$

$$0 \leq l_1 \perp a_{2y}^{\ell+1} + \delta_{\mathrm{n}}^{\ell+1} \geq 0 \tag{3.108}$$

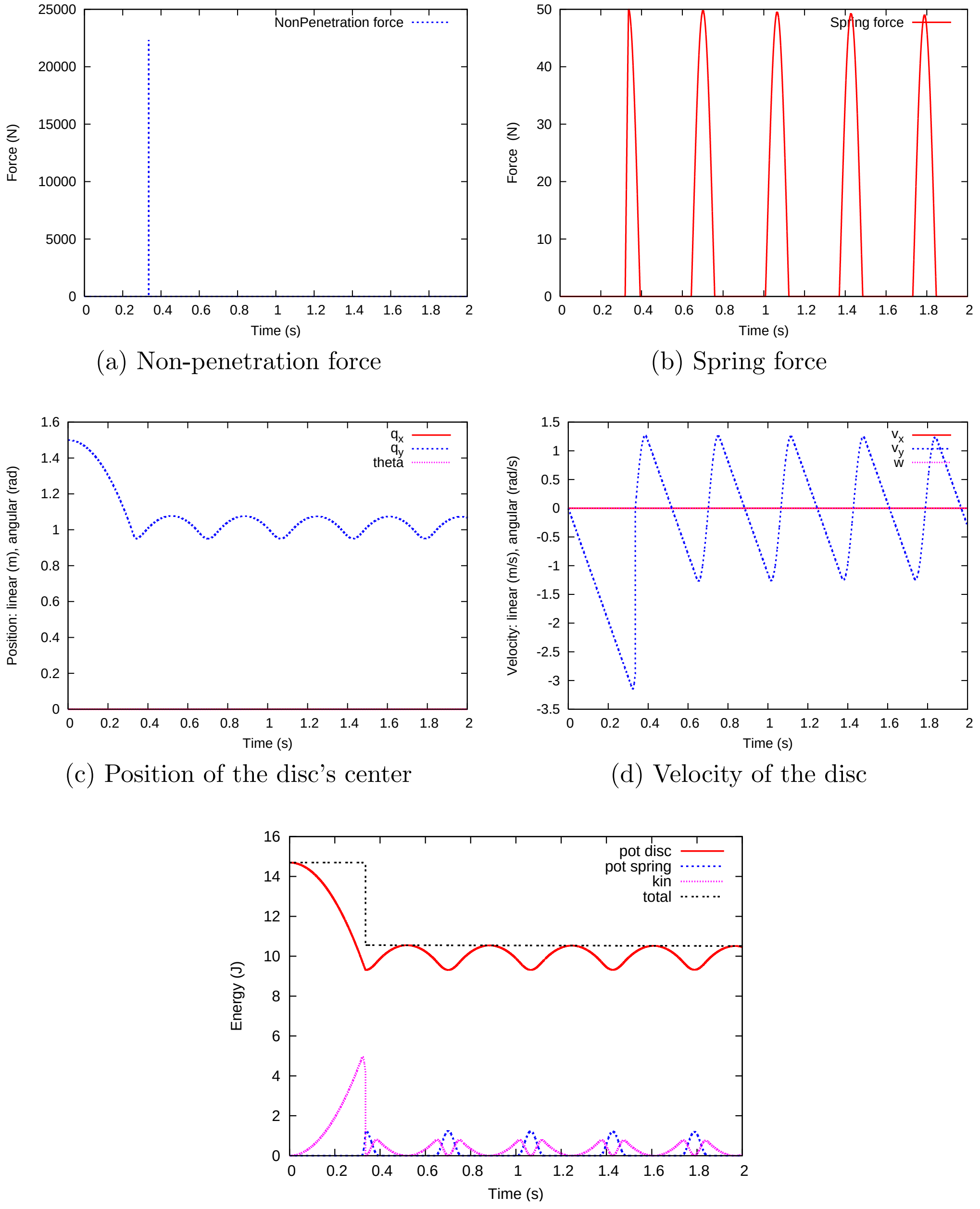
$$0 \leq p_{\mathrm{ns}}^{\ell+1} \perp a_{1y}^{\ell+1} + \delta_{\mathrm{n}}^{\ell+1} \geq 0 \tag{3.109}$$

$$0 \leq p_{\mathrm{nr}}^{\ell+1} \perp \delta_{\mathrm{n}}^{o} - \delta_{\mathrm{n}}^{\ell+1} \geq 0 \tag{3.110}$$

The unit disc's initial position was $\mathbf{q} = [0,\ 1.5, 0]$ with zero initial velocity $\boldsymbol{\nu} = [0,\ 0,\ 0]$. The only force acting on the disc was gravity. The mass of the disc was 1 kg and the moment of inertia about the center of mass was 0.5 kg$\cdot$ m^2. We used a step size $h = 10^{-4}$s. The spring stiffness we used was $k = 1000$kg/s^2. The maximum penetration depth was altered for two experiments such that for the first experiment impact with the rigid core occurs, and for the second experiment impact with the rigid core does not occur. For experiment one, $\delta_n^0 = 0.05$m and for experiment two $\delta_n^0 = 1$m.

Figure 3.11 illustrates the results of the first experiment in which the maximum spring deflection was not large enough to prevent impact with the rigid core. There is a large non-penetration impulse (Fig. 3.11a) generated at approximately 0.34 seconds corresponding to when the spring reached maximum deflection and impact with the rigid core occurs. As expected with a rigid impact, we also see an instantaneous change in velocity (Fig. 3.11d) to zero and loss of energy (Fig. 3.11e). Subsequent to the impact, the motion of the disc (Fig. 3.11c) become oscillatory as it bounces on the undamped spring (Fig. 3.11b) and the velocity is smooth. As seen in Figure 3.11e the total energy is preserved after impact within a tolerance of 10^{-5}J, which is acceptable using a time step of 10^{-4}s and an Euler approximation in the time-stepping formulation.

For the second round of experiments, the maximum spring deflection was set large enough that impact with the rigid core never occurs. We see the oscillatory behavior of the position over the lifetime of the simulation (Fig. 3.12c) as expected with an undamped spring. As guaranteed by our model, no component of the normal force comes from impact with the rigid core; the spring contributes solely to the normal force (Fig. 3.12b). Additionally, without any impacts the plot of velocity is smooth with changes occurring only from the force of gravity and the spring force (Fig. 3.12d). Since there is no impact nor damping of the spring, we expect there to be no loss of energy in the system. Figure 3.12a confirms this, where again the energy is conserved within a numerical tolerance of 10^{-5}J.



(a) Non-penetration force

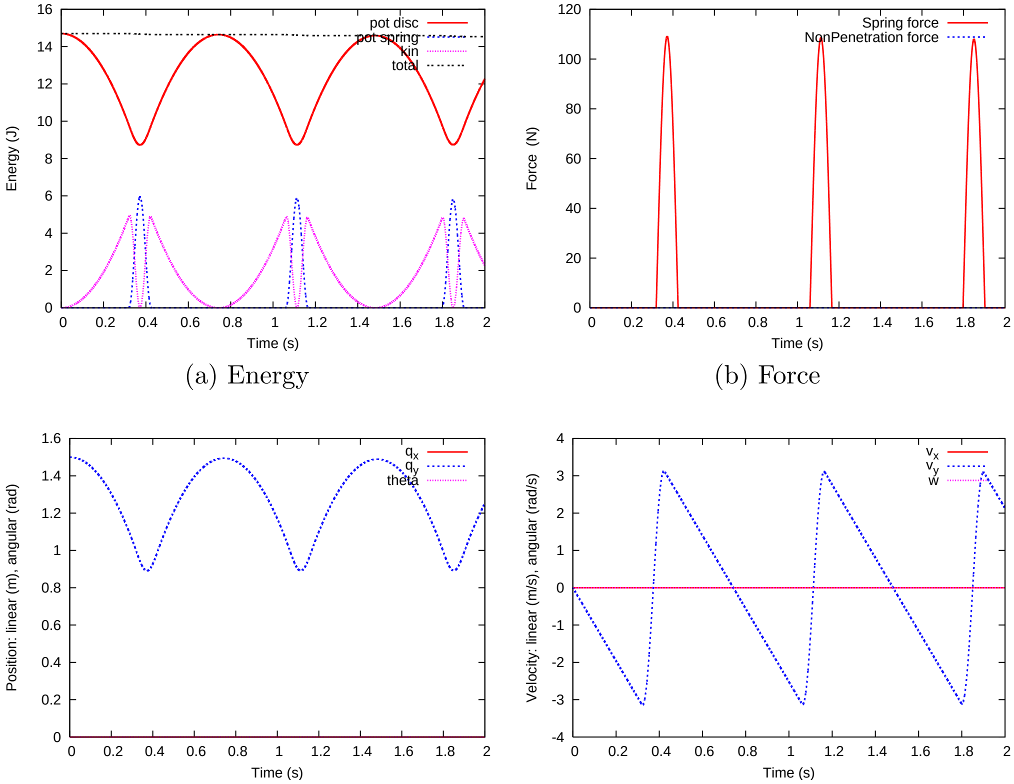
(b) Spring force

(c) Position of the disc's center

(d) Velocity of the disc

(e) Energy

Figure 3.11: Simulation results for a unit disc falling on a half-plane making contact with the rigid core.



(a) Energy

(b) Force

(c) Position

(d) Velocity

Figure 3.12: Simulation results for a unit disc falling on a half-plane without making contact with the rigid core.

3.5.3.4 Example 4: Probabilistic Grasp Planning

In this example, we repeated, in simulation, an earlier grasping experiment done by Brost and Christensen [20]. The goal of the experiment was to study the probability of success of a given grasping strategy under uncertainty in the friction forces between the gripper and part and between the part and the support surface. Performing these experiments in simulation has the advantage of being faster and cheaper, but is useless if the simulation is not accurate enough. If one uses a polyhedral approximation for representing the geometry of the lock piece, they have implicitly assumed that the set of nonpenetration constraints is “locally” convex. That is, for the duration of the current time step, the set of nonpenetration constraints can be accurately represented as a conjunction of linear inequalities. This assumption is violated whenever a vertex of the gripper is “close” to a vertex of the part [36].

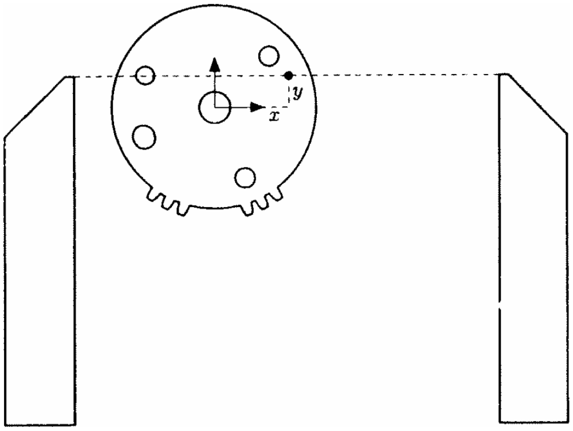


Figure 3.13: The parameterization of the system as shown in Figure 11 from [20]

Figure 3.13 is taken from [20] and illustrates the system, a parallel jaw gripper and a gear piece. We model the gear as a union of super-ellipses and model the grippers as intersections. Let $\mathbf{q} = [x_g \; y_g \; \theta_g \; x_r \; x_l]^T$ be the generalized configuration of the system, where $[x_g \; y_g \; \theta_g]$ is the position and orientation of the gear, x_r is the x-position of the right gripper's fingertip, and x_l is the x-position of the left gripper's fingertip.

The gear's equations are given by the union of:

$$f(\xi_x, \xi_y) \equiv (\xi_x - q_1)^2 + (\xi_y - q_2)^2 - 20.75 = 0 \tag{3.111}$$

$$f_i(\xi_x, \xi_y) \equiv \left(\frac{\xi_x - (q_1 + r_{ix})}{2.3}\right)^4 + \left(\frac{\xi_y - (q_2 + r_{iy})}{1.1}\right)^4 - 1 = 0 \tag{3.112}$$

where $\mathbf{r}_i$ is the vector from the gear's center of mass to the center of the ith tooth[10]. The values used for this simulation were $\mathbf{r}_i = 20.75[\cos(\theta_i) \;\; \sin(\theta_i)]^T$, where $\theta_1 = 4.103727$, $\theta_2 = 4.253125$, $\theta_3 = 4.402522$, $\theta_4 = 5.022255$, $\theta_5 = 5.171653$, and $\theta_6 = 5.321051$.

The grippers only make contact with the gear with their respective fingertip, inside edge, or top edge. Therefore we simplify the modeling of the grippers to be

[10] For brevity frame transformations have not be shown, in practice one must be careful to maintain frame consistency.

the intersection of the two planes adjacent to the fingertip vertex. For the right gripper, the equations are given by the intersection of:

$$f_{r1}(\xi_x, \xi_y) \equiv \xi_y = 0 \tag{3.113}$$

$$f_{r2}(\xi_x, \xi_y) \equiv -(\xi_x - q_4) = 0 \tag{3.114}$$

and for the left gripper, the equations are given by the intersection of:

$$f_{l1}(\xi_x, \xi_y) \equiv \xi_y = 0 \tag{3.115}$$

$$f_{l2}(\xi_x, \xi_y) \equiv \xi_x - q_5 = 0 \tag{3.116}$$

The non-penetration constraint between the right gripper and the disc part of the gear is,

$$0 = \mathbf{a}_2 - \mathbf{a}_1 + l_1 \nabla_{\mathbf{a}_2} f(\mathbf{a}_2) \tag{3.117}$$

$$0 = \nabla_{\mathbf{a}_2} f(\mathbf{a}_2) + l_2 \nabla_{\mathbf{a}_1} f_{r1}(\mathbf{a}_1) + l_3 \nabla_{\mathbf{a}_1} f_{r2}(\mathbf{a}_1) \tag{3.118}$$

$$0 \leq l_1 \perp -f(\mathbf{a}_2) \geq 0 \tag{3.119}$$

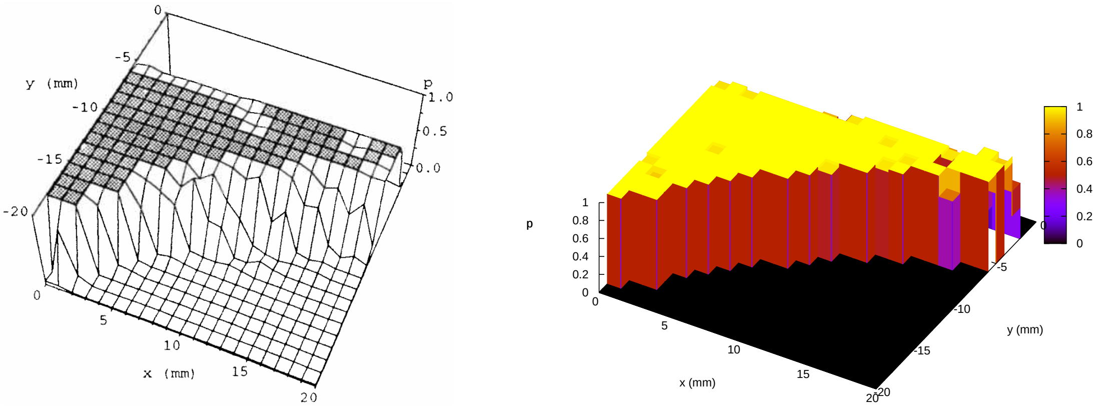
$$0 \leq l_2 \perp -f_{r1}(\mathbf{a}_1) \geq 0 \tag{3.120}$$

$$0 \leq l_3 \perp -f_{r2}(\mathbf{a}_1) \geq 0 \tag{3.121}$$

$$0 \leq \boldsymbol{\lambda}_{in} \perp f(\mathbf{a}_1) \geq 0 \tag{3.122}$$

where $\mathbf{a}_1$ is the closest point on the right gripper to the disc, $\mathbf{a}_2$ is the closest point on the disc to the right gripper, and $\boldsymbol{\lambda}_{in}$ is the component of the non-penetration force associated with this contact. The remaining contact constraints can be formulated analogously.

Friction in the plane of motion is important, so for this problem we used a 2.5D dynamic model [18], with force-controlled parallel jaw grippers. In order to virtual replicate the original simulation, we assumed that the coefficients of surface friction, and coefficient of gripper friction were unknown parameters. Then for each (x_i, y_i) in a 20mm by 20mm box of initial positions (sampled at 1mm resolution), assign random values to the unknown parameters and simulate the grasp. The success probability for each (x_i, y_i) pair was estimated as the total number of successful grasps divided by the number of trials. The goal configuration (as defined in [20]) is when the tips of the fingers lie on the outer edges of the outer-most teeth.



(a) Actual

(b) Simulation

Figure 3.14: Comparison between the results found by Brost and Christiansen in [20] to our simulation results.

Figure 3.14 illustrates a comparison between our results and the results reported by Brost and Christiansen; qualitatively, our results agree well.

3.6 Summary

Four new time-stepping formulations were presented. These formulations were constructed for a variety of reasons, including accuracy, performance, problem class, and design. We see several directions for future work. We would like to address the existence and uniqueness of solutions for the fully implicit mixed NCP we developed. For our work with non-convex polygonal constraints, we would like to extend the method to 3D bodies. In addition, when interpenetration has occurred, there is nondeterminism in the formulation that we would like to remove. We have already begun implementing these extensions, details can be found in the following tech report [82]. The next chapter discusses a new software tool, which implements these time-stepping formulations.

4. daVinci Code

This chapter discusses a new software tool we created dubbed dVC. It is designed to facilitate simulation, analysis, and virtual design of multibody systems with intermittent unilateral contacts with dry friction. As stated earlier, these systems are notoriously difficult to simulate accurately due to the nonsmooth nature of the underlying mathematical model. In fact, commercial software for simulating multibody systems with unilateral contacts and dry friction (DADs, Adams, Working Model) deal with the nonsmoothness by *ad hoc* regularization methods (*e.g.,* penalty methods to remove unwanted interpenetration of bodies by local nonlinear spring and damper effects). These methods require tuning of simulation parameters and the algorithms often have no guarantees on stability and convergence. As a result, the user must select several tuning parameters (including the size of the time step and material properties) by trial-and-error just to achieve stable, believable simulation. In these approaches, it is difficult to prove convergence, and there is no way to benchmark the accuracy of such methods.

In contrast, dVC uses state-of-the-art time-stepping methods, discussed in earlier chapters, to capture the nonsmooth phenomena (stick-slip transitions and contact loss and formation) without regularization. These methods are numerically stable and provably convergent. In particular, it has been proven that when the Stewart-Trinkle (ST) and Song-Pang-Kumar (SPK) methods are applied to dynamic systems, as the step size goes to zero, the trajectories produced by the time-steppers converge to the exact solution of the original instantaneous-time model [45, 109]. This convergence property has been available for solvers for ordinary differential equations and differential algebraic equations for some time, but only recently for some constraint-based time-stepping methods.

As alluded to above, dVC allows the user to choose between different motion models. It allows comparison and validation of multibody modeling choices (*e.g.*, one chooses to use a quasistatic model, but should have chosen a dynamic model). Second, it enables a hierarchical and iterative approach to design processes. As an example, consider the design of a part feeder (also discussed later). Geometric and dynamic parameters are critical to the functioning of the device and there are many parameters (e.g., coefficients of friction) that are characterized by uncertainty. We would like to be able to consider only the geometry initially to prune the design space eliminating large sets of geometric parameters that render the design infeasible. This can be done by simple geometric models. We then want to be able to refine the design space with quasi-static models, and then using more expensive dynamic models. Thus we can use low-resolution models to quickly eliminate many design alternatives using more complicated models characterized by many unknown parameters to search over a smaller set of feasible designs.

4.1 Simulation Overview

dVC consists of several modular components (plugins) that have been written in such a way that they can be extended and replaced at runtime based on the desired configuration. Figure 4.1 gives a high level view of the components of dVC. The main function of the library is to provide the *API* (Application Programming Interface), which is our defined set of calling conventions. The library acts as the central controller organizing calls to the plugins, which in turn do the actual physics simulation work.

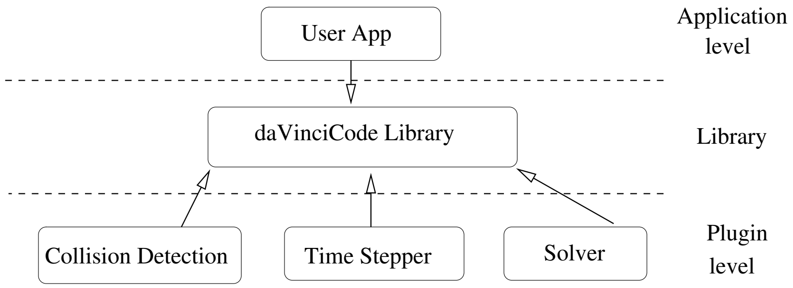


Figure 4.1: daVinci Code Architecture.

During simulation, an example time step consists of the following actions:

1. call collision detection plugin with the current scene's state

2. call the time-stepper plugin with the results of the collision detection plugin

3. call the solver plugin with the complementarity problem formulated in the time-stepper plugin

4. Update the bodies' state with the result of the solver plugin

4.1.1 Bodies

The framework for specifying bodies in a simulation was designed to be both flexible and object oriented. All bodies in dVC will from a base *Body* type, which consists of the minimal amount of data any body in dVC will need. For example, the base class contains configuration information, a pointer to geometric data, a list of contacts, etc. A scene cannot actually contain a body from this base class, a user must implement their own body or use one of the three pre-defined classes available: *Obstacle*, *Kinematical*, or *Dynamical.*

The reasons for these different body types are efficiency and clarity. The simplest body type is Obstacle, which is a body with a configuration fixed for the duration of the simulation. It does not contribute any unknowns to the time-stepping model, however, it can become involved in collisions (or near misses), so its geometric information is made available during collision detection.

The next slightly more complicated body type is the Kinematical body. This body type implies the body is not subject to dynamical forces, but is instead kinematically controlled by a user specified function of time. Since the configuration and velocity of this type of body is known for all time during simulation, it does not contribute unknowns to the model of the system. However, it too can become involved in collisions, so its geometric information is made available during collision detection. From the user's perspective, including a Kinematical body in a simulation is quite simple. One merely needs to set the position control law for the body, which is handled by our abstract PositionController class that is easily extendable. The user simply implements a single function callback specifying the position and orientation of the body as a function of time.

The last body type, and also the most complicated, is the Dynamical body. This body type contains velocity, constraint forces, mass and inertia values, etc. It also can have several force controllers attached (in similar fashion to position controller) for the application of non-constraint forces (e.g. control input.)

4.2 Plugin Overview

From figure 4.1, it may appear that the plugins are independent, but this is not the case. For an example, the time-stepper plugin implicitly defines if the problem formulation is an NCP or an LCP, which effects the solver plugin. In addition, the collision detection plugin implicitly defines a geometric model. These inter-level dependencies are managed at the library level, assuring that the selected plugins are compatible.

4.2.1 Collision Detection

dVC uses collision detection and distance computation in two different ways. Under the rigid body assumption, the formulation of a time-step problem requires knowledge of points in contact, contact normals, and the same information for points not quite in contact and for points that have penetrated (due to numerical or linearization errors). In the rigid body case, specific times and locations of collisions are not required; nonetheless, the solution of the time-step problem is consistent with the model at the end of the time step. On the other hand, if the bodies are assumed to have compliant surfaces, then distance queries must be used to find precise times of impact between bodies. This is required to properly calculate local deformations and for adapting the time step based on the effective local stiffness. In both cases, dVC maintains an *active constraint set*, which contains the distance and normal information for all geometric feature pairs that could come into contact during the ensuing time-step.

4.2.2 Time-steppers

The TimeStepper class approximates force and state trajectories for the bodies in the scene over the course of the simulation. By designing the TimeStepper class to be extensible, we have been able to implement several time-stepping models that the user can use interchangeably.

4.2.3 Complementarity Problem Solver

dVC has implemented a wrapper plugin that uses the well-known PATH solver [43] to solve the subproblems. The PATH solver was introduced in 1995 and has since become the standard against which all others are compared [19]. In addition to solving the standard LCPs, the PATH solver can also solve mixed linear and nonlinear complementarity problems. PATH is closed source, but has a freely downloadable binary available from [42].

Our focus to date with dVC has been engineering applications, however other applications such as haptics or virtual reality may make use of iterative solvers. Iterative solvers improve solution accuracy monotonically and thus are appealing to these applications since an "early exit" often yields a solution that is believable, despite its inaccuracy. The flexibility of dVC allows the user to develop various solvers, and select them based on the current task at hand.

4.3 Simulated and Experimental Results

To illustrate the capabilities of dVC, several example systems are presented with experimental results. The first example describes how dVC can be used to help design dynamical systems with intermittent frictional contacts and uncertainty in the model. In the second example, dVC was used to design open loop plans for a planar micro-manipulation task. In the final example, dVC was used to analyze grasping strategies for a planar part; choosing strategies where success is likely even when "a guaranteed strategy does not exist" [20].

4.3.1 Design of a Part Reorienting Device

This example is taken from a parts feeding application [103]. It is a device with a cavity of complex geometry (as shown in Figure 4.2) designed to orient a cup-shaped part. Regardless of the part's initial orientation in the top chute, after it falls through the cavity it must enter the lower vertical chute with its center of gravity down.

The problem consists of a design parameter space P and an uncertainty space M. Design parameters are parameters we have control over (in this case, 12 param-

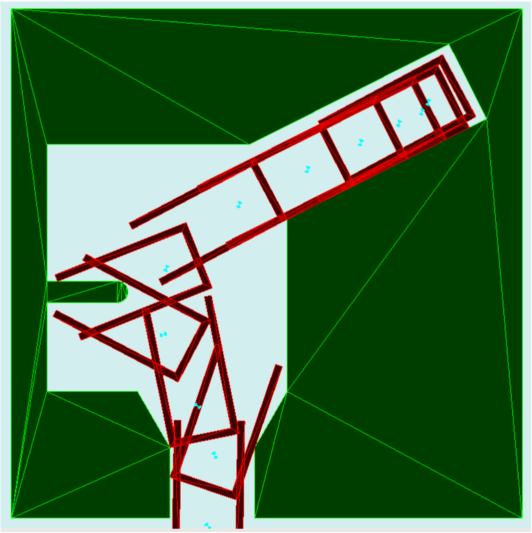


Figure 4.2: Snapshots of the gravity-fed part in the feeder.

eters) that define the geometry of the device. The uncertainty space exists because models are not perfect and certain physical properties (like friction coefficients) are difficult to measure and predict. The state of the system $x(t)$ was a function of: t, x_0, p, and m, where $p \in P$, $m \in M$ and x_0 is the initial condition. The goal was to find some $p \in P$ that works for all $m \in M$.

The solution approach was to randomly sample P for a design that was geometrically feasible (motion planning), then thoroughly sample M to verify the design. The dVC simulator lends itself nicely to this verification system through its modular design. Using dVC, a hierarchical algorithm was written and easily implemented:

1. Verify with geometric model
2. For all $m_i \in M_\mu$, verify with inelastic ST model
3. For all $m_i \in M$, verify with SPK model

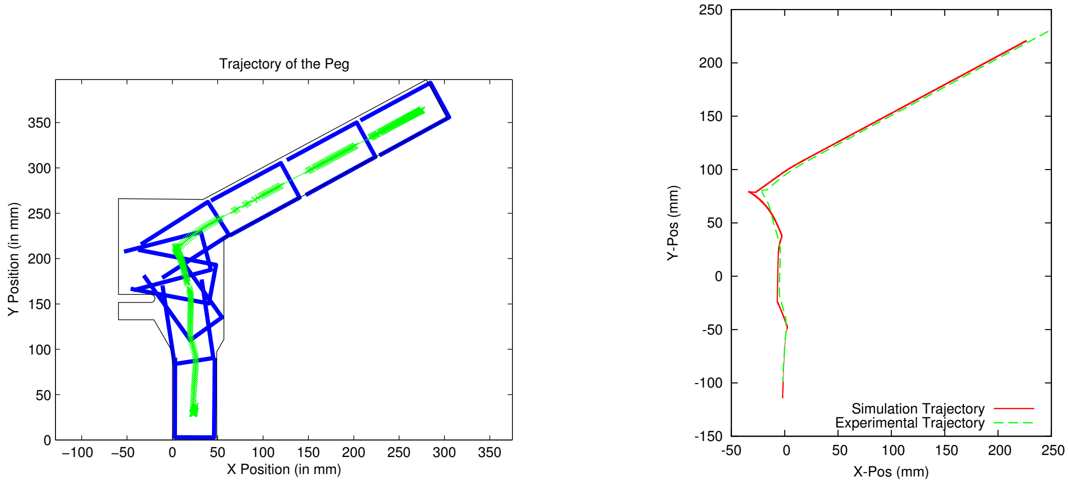
where the M_μ uncertainty space consists of the unknown initial cup orientation and friction coefficients and M adds the additional unknown parameters arising from the locally deforming viscoelastic frictional contacts.

This hierarchical planning through the use of dVC allowed for a significant savings in design time, since the simpler model could be used to prune away failed

	Naive Verification	Hierarchical Verification
Geometric	N/A	603 s
ST	5000 s	2935 s
Total	5000 s	3538 s

Table 4.1: Hierarchical approach to design

designs without the need of testing with the more accurate, but also more computationally expensive model. Table 4.1 displays the running times of the verification algorithms under the naive approach, and our hierarchical approach for the geometric and inelastic ST models. We saw of savings of 1462 seconds using our hierarchical planner because we did not run the more computationally expensive dynamic model on designs where no geometric solution existed.



(a) Trajectory of the Peg

(b) Comparison of the Simulated and Experimentally Captured Trajectories

Figure 4.3: The left figure shows time-elapsed image of the part passing through the orienting device. The green line with crosses is the trajectory of the center of gravity. The right figure shows a comparison of experimental data with the simulation results.

A similar experimental device was built in the GRASP Lab at the University of Pennsylvania and instrumented to compare real and simulated trajectories. A typical trajectory of the part is illustrated in Figure 4.3a. A motion capture system and basic image processing tools were used to extract the boundary of the part at a rate of 500Hz. The boundary images were fit to a simple graphical model for display purposes. The trajectories obtained via experimentation and the Stewart-Trinkle time-stepper are shown overlaid in Figure 4.3b. The path of the cup shaped part determined from the simulated feeder matches closely when experimentally verified.

4.3.2 Needle pushing a planar slider

The second experiment is an assembly problem taken from a planar micromanipulation experiment [22]. Figure 4.4 presents 2 pictures of the experimental setup as seen through a microscope. The goal in this problem is to find a pushing path for the needle (labeled probe in figure 4.4a) to push the block from the initial configuration to a goal configuration (shown in figure 4.4b). dVC was used to simulate candidate pushing plans from a space of possible pushing plans to find one capable of accomplishing the task.

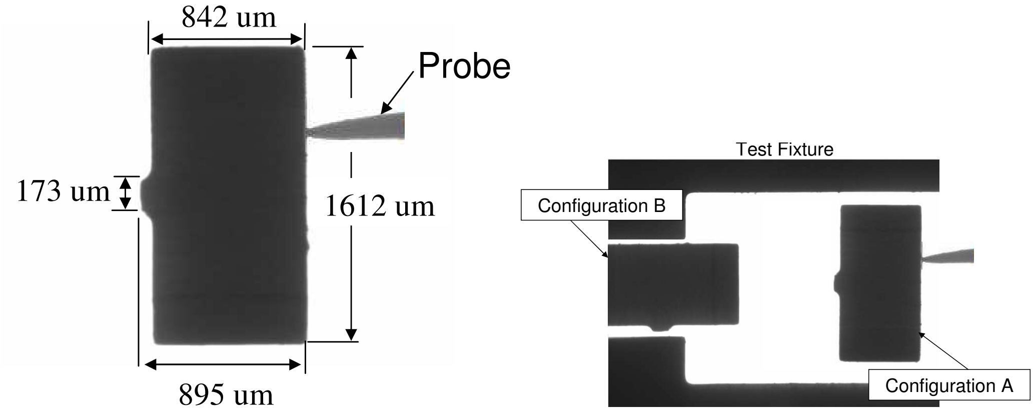
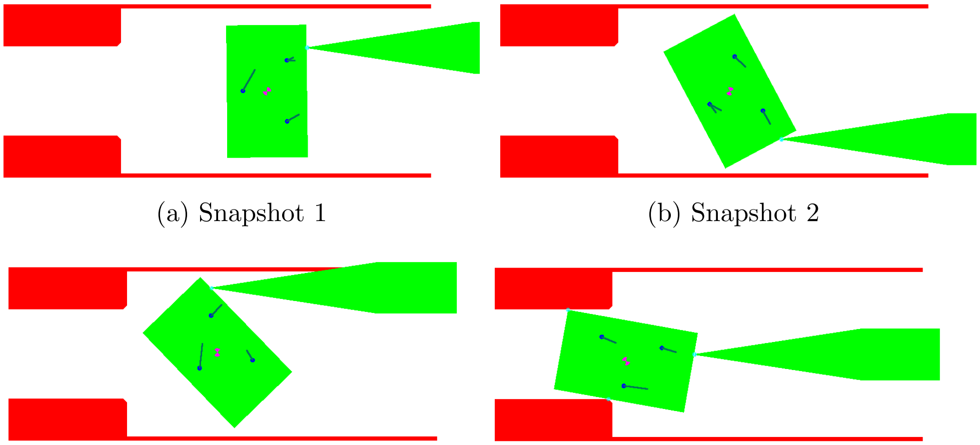


Figure 4.4: Images taken from a microscope of the experimental micro part and assembly. The left image shows the dimensions of the part and the right image shows the initial (A) and goal (B) configurations.

Since friction in the plane of motion was important and inertial forces were an order of magnitude smaller than the frictional due to dimensions, for this problem we used a 2.5D quasi-static model, with a position-controlled trajectory of the needle. The locations of the support points and coefficients of friction were identified through experimentation. Figure 4.5 illustrates several frames of simulation. The support points are the 3 small dark circles inside the peg, and the lines extending from them are a visualization of the friction force components resisting the motion of block.



(a) Snapshot 1

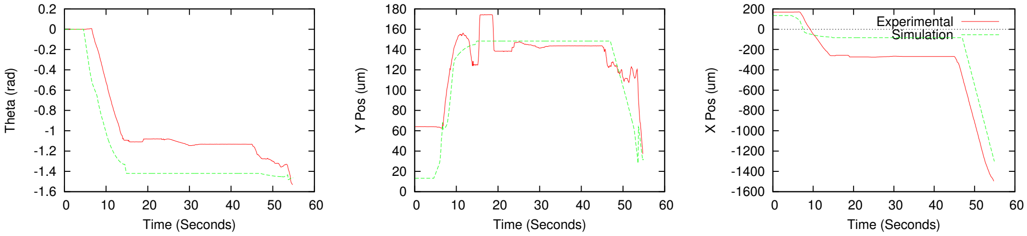
(b) Snapshot 2

(c) Snapshot 3

(d) Snapshot 4

Figure 4.5: Four screen shots from the simulated planar micromanipulation task. The small dark circles inside the peg are the support points shown with corresponding tangential friction force vectors.

Figure 4.6 shows a comparison of the simulated and experimental trajectories. The path of the peg determined from the simulated pushing path matches closely when experimentally verified. The other components of the trajectory also match closely. The flexibility of dVC allows us to choose the desired physical model and also to use it as an open loop path planning tool.



(a) θ Versus Time

(b) Y-Pos Versus Time

(c) X-Pos Versus Time

Figure 4.6: Comparison of 2.5D simulation with experimental data.

4.3.3 Probabilistic Grasp Planning

In this example, dVC was used to simulate an earlier grasping experiment [20]. The goal of the experiment was to study the probability of success of a given grasping strategy under uncertainty in the friction forces between the gripper and part and between the part and the support surface. The task is to grasp a planar part with a parallel-jaw gripper. The part and gripper are shown in figure 4.7, as well as the coordinate system used in the simulation. The part is slightly more than 40 mm in diameter. The midway point between the tips of the fingers is measured with respect to the center of the part (Figure 4.7). The goal is to estimate the probability of a successful grasp as a function of x and y.

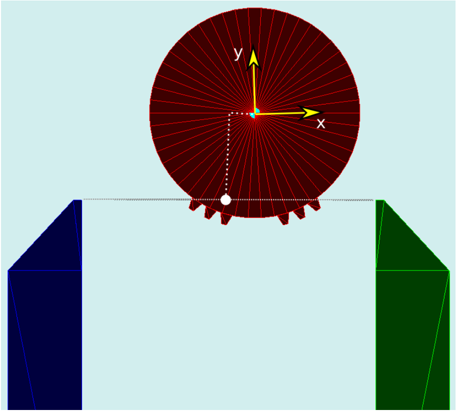
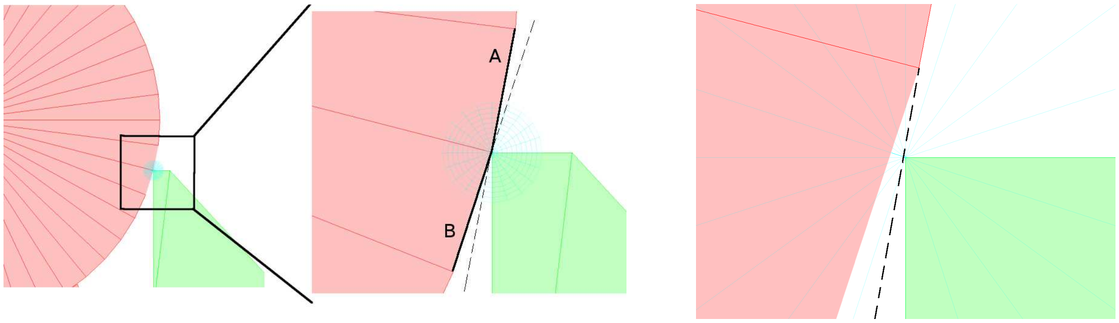


Figure 4.7: The parameters x and y describe the position of the center point between the fingertips, relative to the center of the gear.

Similar to the parts reorienting device example, this example also contains a large uncertainty space: the location of the support points of the part, coefficient of surface friction, and coefficient of gripper friction. The location of the part in the world was known exactly. Similar to the micro-manipulation experiment, friction in the plane of motion is again important, but here we cannot ignore the inertial forces. For this problem we used a 2.5D dynamic model, with force-controlled parallel jaw grippers. Then for each (x_i, y_i) in a 20mm by 20mm box of initial positions (sampled at 1mm resolution), assign random values to the unknown parameters and simulate the grasp. Estimate the success probability for each (x_i, y_i) pair as the total number of successful grasps divided by the number of trials. The goal configuration (as defined in [20]) is when the tips of the fingers lie on the outer edges of the outermost teeth.

When we used our simulator to repeat the grasp acquisition experiments performed by Brost and Christensen, we discovered a problem. The polyhedral approximation of the circular part and grippers led to non-penetration constraints that were non-convex, but the underlying mathematical model used in the simulation assumed that the non-penetration constraints are always convex. That is, the model assumes the set of non-penetration constraints can be accurately represented as a conjunction of linear inequalities. This assumption is violated whenever a vertex of the gripper is "close" to a vertex of the part. This led to the realization that the set of constraints should be represented as a disjunction of inequalities (discussed in section 3.2).



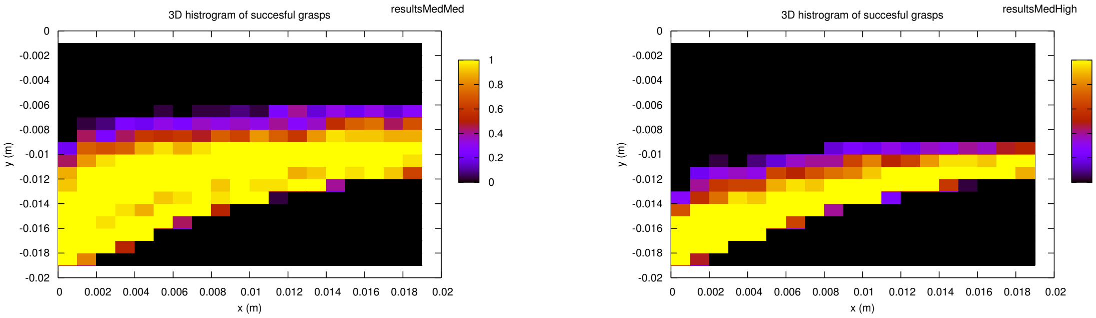
(a) The simulator imposes non-penetration constraints between the gripper vertex and edges A and B (each treated as a half space). Thus the vertex of the gripper is restricted to lie to the right of the fictitious edges (shown as dashed line extensions of the edges) rather than facets A and B (highlighted with bold solid lines).

(b) Position of the gripper and part at the end of the time step. Note that the gripper vertex is positioned on the lower fictitious edge, and is *not* in contact with the boundary of the part.

Figure 4.8: Effect of a non-convex non-penetration constraint in a gripping experiment.

Figure 4.8a shows a vertex of the gripper in contact with a vertex on the part. Because of the proximity of the gripper to two facets, A and B, of the part, the simulator imposes non-penetration constraints between the gripper vertex and edges A and B (each treated as a half space). Thus the vertex of the gripper is restricted to lie to the right of the fictitious edges (dashed line extensions of the edges) rather than facets A and B (highlighted with bold solid lines). Figure 4.8b shows the position of the gripper and part at the end of the time step. Note that the gripper vertex is positioned on the lower fictitious edge, *not* in contact with the boundary of the part. In preparing for the next time step, the collision detection algorithm will eliminate the facet A, and therefore its fictitious edge, from consideration, so the gripper will move toward facet B. Over multiple time-steps, the effect is that the gripper "bounces" rather than slides down the edge of the part. This behavior greatly affects the outcome of each grasp acquisition attempt. A finer polygonal approximation and smaller time steps will reduce the problem, but not remove it.

Modeling the non-penetration constraints as a disjunction of inequalities (discussed in section 3.2) allows us to produce simulation results. Figures 4.9a and 4.9b illustrate two typical simulation results. The figures are 2D projections of the 3D histogram of successful grasps. These figures were created assuming a fixed support point tripod location on the lock, and selecting random values for the coefficients of friction in the specified ranges. Figure 4.9a is the result for medium surface contact friction coefficients and medium gripper contact friction coefficients. Figure 4.9b is the result for medium surface contact friction coefficients and high gripper contact friction coefficients. For the presented results, coefficients of friction from the medium group range in values between: (0.33, 0.66], and high: (0.66, 0.99].



(a) medium surface and medium gripper friction coefficients.

(b) medium surface and high gripper friction coefficients.

Figure 4.9: 2D map of the 3D histogram of successful grasps for various friction coefficients.

Our initial results agree well with the experimental results presented in [20]. Current work focuses on developing a better understanding of the friction coefficient influences, developing better sampling strategies for the support point tripod, incorporating other physical models and developing hierarchical planning techniques similar to example 1.

4.4 Summary

We presented *daVinci Code*, which is a new software tool we designed and implemented to facilitate simulation, analysis, and virtual design of multibody systems with intermittent frictional unilateral contact. We are currently extending our simulator to work for 3D systems. We also would like to develop a suite of test problems, for which we can test the accuracy of various time-stepping methods. The next chapter performs one such study.

5. Sources of Error in a Simulation of Rigid Bodies

In this chapter we present initial results (published in [17]) verifying the accuracy of the Stewart-Trinkle time-stepping method [110] presented earlier, which lies at the core of the dVC physics engine [18] presented in chapter 4. One of the benefits of our time-stepper is that as the time step goes to zero, the solution trajectories converge to a solution of the original instantaneous-time problem [45, 109]. Additional results are presented showing the effects of various sources of model approximation errors. Most of the problems studied had time-stepping subproblems formulated as linear complementarity problems (LCPs). These were simulated using dVC. However, custom C-code was written for problems whose subproblems were formulated as nonlinear complementarity problems (NCPs). All the LCPs and NCPs were solved by the state-of-the-art complementarity problem solver, PATH [43].

All the results presented here were obtained through numerical studies of a challenging and important problem studied analytically and experimentally by Vose *et al.* [118, 119, 120]. In particular, they studied the motion of a particle on a nominally rigid plate moving with a specified periodic (high frequency) trajectory. When viewed on a long time scale (several cycles at a time), such periodic inputs cause the friction forces to generate velocity fields that can be used to move parts along specified paths on the plate. This work is motivated by problem of assembly of tiny parts that are very difficult for humans (or robots) to manipulate, but could possibly be assembled (possibly many at once) on a vibrating plate.

As will be seen below, the advantage of studying this problem via simulation is that after verifying our simulation results against their theoretical and (qualitative) experimental results, we were immediately able to study an extended array of problems that violated the assumptions made by Vose *et al.* to facilitate their analytical study. One of these assumptions was that the particle was always in sliding contact with the plate. Its main effect was to limit the location of the particle to a small region of the plate where the assumption held. We were able to study the motion of the particle in contact anywhere on the plate, observing motions with stick-slip behavior and loss of contact. In addition, we are able to study the effects of linear approximations of the friction cone at the contact point, a commonly applied simplification of the dynamic model. These results are discussed below.

A second result presented is in designing new plate motions to generate a desired part motion. This is done through an optimization framework, where a simulation of the part interacting with the plate (including the full dynamics of the system) is performed, and based on the results of the simulation the motion of the plate is modified.

5.1 Dynamic Model

The system Vose *et al.* studied [118, 119, 120] consists of two pieces (Figure 5.1); a kinematically controlled plate and a dynamical part interacting with the plate.

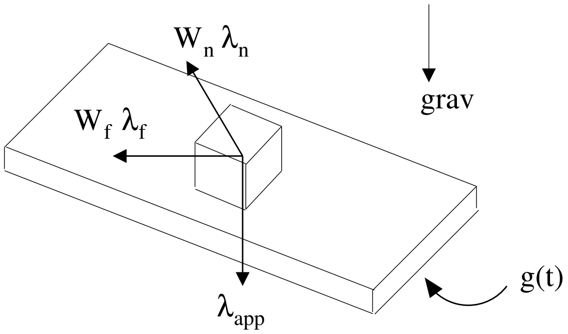


Figure 5.1: The plate is kinematically controlled by the vector function $g(t)$. There are three forces acting on the part: the force due to gravity $\boldsymbol{\lambda}_{\mathbf{app}}$, the non-penetration constraint force $\mathbf{W}_n\boldsymbol{\lambda}_n$, and the frictional force $\mathbf{W}_f\boldsymbol{\lambda}_f$.

Let $\mathbf{q} = [x\ y\ z\ e_s\ e_x\ e_y\ e_z]^T$ be the configuration of the part where $[x\ y\ z]^T$ is the location of the center of mass of the part in a fixed world frame and $[e_s\ e_x\ e_y\ e_z]^T$ is a unit quaternion representing the orientation of the part with respect to the fixed frame. $e_s = \cos(\theta/2)$ and $[e_x\ e_y\ e_z] = \sin(\theta/2)\hat{\mathbf{b}}$ where θ is the angle of rotation and $\hat{\mathbf{b}}$ is a unit vector parallel to the axis of rotation. Let $\boldsymbol{\nu} = [v_x\ v_y\ v_z\ \omega_x\ \omega_y\ \omega_z]^T$ be the velocity twist of the part in a fixed world frame, where $\mathbf{v} = [v_x\ v_y\ v_z]$ is the translational velocity of the center of mass of the part and $\boldsymbol{\omega} = [\omega_x\ \omega_y\ \omega_z]$ is the angular velocity. Let $\boldsymbol{\lambda}_{\text{app}}(\mathbf{q}, t) \in \mathbb{R}^6$ represent the resultant of the externally applied forces at time t, which for this problem is only gravity. Let $\hat{\mathbf{n}}_i$ be the unit vector normal to the tangent plane at contact i, and $\hat{\mathbf{t}}_i$ and $\hat{\mathbf{o}}_i$ be two orthogonal unit vectors both orthogonal to $\hat{\mathbf{n}}_i$ (*i.e.* basis vectors of the tangent plane) at contact i. Let $\lambda_{in} > 0$ be the magnitude of the normal contact force at the ith contact point, and λ_{it} and λ_{io} the corresponding orthogonal friction force components. Further, let $\boldsymbol{\lambda}_{\text{n}}$ be the concatenated vector of all the normal contact force magnitudes, and $\boldsymbol{\lambda}_{\text{t}}$ and $\boldsymbol{\lambda}_{\text{o}}$ the respective concatenated vectors for the tangential force magnitudes.

5.1.1 Instantaneous Dynamics

For this particular system of a part interacting with a kinematically controlled plate (figure 5.1), the DCP can be written as [24, 113, 112]:

$$\mathbf{M}(\mathbf{q})\dot{\boldsymbol{\nu}} = \mathbf{W}_{\text{n}}(\mathbf{q})\boldsymbol{\lambda}_{\text{n}} + \mathbf{W}_{\text{t}}(\mathbf{q})\boldsymbol{\lambda}_{\text{t}} + \mathbf{W}_{\text{o}}(\mathbf{q})\boldsymbol{\lambda}_{\text{o}} + \boldsymbol{\lambda}_{\text{app}}(\mathbf{q}, t) + \boldsymbol{\lambda}_{\text{vp}}(\mathbf{q},\ \boldsymbol{\nu}) \tag{5.1}$$

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\boldsymbol{\nu} \tag{5.2}$$

$$0 \leq \boldsymbol{\lambda}_{\text{n}} \perp \boldsymbol{\psi}_{\text{n}}(\mathbf{q}, t) \geq 0 \tag{5.3}$$

$$0 = (\mathbf{U}\boldsymbol{\lambda}_{\text{n}})\circ(\mathbf{v}_{\text{t}}) + \boldsymbol{\lambda}_{\text{t}}\circ\boldsymbol{\sigma} \tag{5.4}$$

$$0 = (\mathbf{U}\boldsymbol{\lambda}_{\text{n}})\circ(\mathbf{v}_{\text{o}}) + \boldsymbol{\lambda}_{\text{o}}\circ\boldsymbol{\sigma} \tag{5.5}$$

$$0 \leq \boldsymbol{\sigma} \perp (\mathbf{U}\boldsymbol{\lambda}_{\text{n}})\circ(\mathbf{U}\boldsymbol{\lambda}_{\text{n}}) - \boldsymbol{\lambda}_{\text{t}}\circ\boldsymbol{\lambda}_{\text{t}} - \boldsymbol{\lambda}_{\text{o}}\circ\boldsymbol{\lambda}_{\text{o}} \geq 0 \tag{5.6}$$

Equation (5.1) is the Newton-Euler equations of the system. The matrix $\mathbf{M}(\mathbf{q}) = \begin{bmatrix} m\mathbf{I}_{3\times 3} & 0 \\ 0 & \mathcal{I}(\mathbf{q}) \end{bmatrix}$ is the mass-inertia matrix of the part, where m is the mass of the part and $\mathcal{I}$ is the inertia tensor. The vector $\boldsymbol{\lambda}_{\text{vp}} = \begin{bmatrix} 0 \\ -\boldsymbol{\omega} \times \mathcal{I}(\mathbf{q})\boldsymbol{\omega} \end{bmatrix}$ is the velocity product term of Euler's equation. The matrices $\mathbf{W}_{(\cdot)} = \begin{bmatrix} \cdots \mathbf{W}_{i(\cdot)} \cdots \end{bmatrix} \in \mathbb{R}^{6\times n_c}$, where n_c is the number of contacts, are dependent on $\mathbf{q}$ and map the normal and frictional wrench magnitudes to the body reference frame:

$$\mathbf{W}_{in}(\mathbf{q}) = \begin{bmatrix} \hat{\mathbf{n}}_i(\mathbf{q}) \\ \mathbf{r}_i(\mathbf{q}) \times \hat{\mathbf{n}}_i(\mathbf{q}) \end{bmatrix} \quad \mathbf{W}_{it}(\mathbf{q}) = \begin{bmatrix} \hat{\mathbf{t}}_i(\mathbf{q}) \\ \mathbf{r}_i(\mathbf{q}) \times \hat{\mathbf{t}}_i(\mathbf{q}) \end{bmatrix} \quad \mathbf{W}_{io}(\mathbf{q}) = \begin{bmatrix} \hat{\mathbf{o}}_i(\mathbf{q}) \\ \mathbf{r}_i(\mathbf{q}) \times \hat{\mathbf{o}}_i(\mathbf{q}) \end{bmatrix} \tag{5.7}$$

where $\mathbf{r}_i$ is a vector from the center of mass of the part to contact i. The dependence on $\mathbf{q}$ may not be obvious as written, since for brevity we have not written the vectors with respect to a frame. Typically $\hat{\mathbf{n}}_i$ is known in the fixed world frame but $\mathbf{r}_i$ is known in the body fixed frame ($\mathbf{b}$). Therefore, the cross product term in the wrenches becomes $\mathbf{R}^{\mathbf{b}}\mathbf{r}_i \times \hat{\mathbf{n}}_i$, where $\mathbf{R}(\mathbf{q}) \in \mathbb{R}^{3\times3}$ is the rotation matrix converting the body frame to the fixed world frame. Similarly, the inertia matrix is known with respect to the body fixed frame and also must be converted to the fixed world frame, $\mathcal{I} = \mathbf{R}^{\mathbf{b}}\mathcal{I}\mathbf{R}^T$.

Equation (5.2) is the kinematic map of the system where $\mathbf{G}$ is the matrix mapping the generalized velocity of the body to the time derivative of the position and orientation. $\mathbf{G}(\mathbf{q}) = \begin{bmatrix} \mathbf{I}_{3\times3} & 0_{3\times3} \\ 0_{4\times3} & \mathbf{J}_{4\times3}(\mathbf{q}) \end{bmatrix}$ where $\mathbf{I}_{3\times3}$ is the identity matrix of given size and $\mathbf{J}(\mathbf{q}) = \frac{1}{2}\begin{bmatrix} -e_x & -e_y & -e_z \\ e_s & e_z & -e_y \\ -e_z & e_s & e_x \\ e_y & -e_x & e_s \end{bmatrix}$.

Equation (5.3) is the nonpenetration constraint for all contacts written as a complementarity condition where $\boldsymbol{\psi}_{\mathrm{n}}$ is a concatenated vector of all the signed distance functions for each contact i (*e.g.* the signed distance between the part's corners and the kinematic plate's face). The distance is positive at contact i when the vertex-face pair at this contact is separated, it becomes zero when the vertex-face pair is touching, and it becomes negative when the vertex has penetrated the face. Note that in general there is no closed form expression for $\psi_{in}(\mathbf{q}, t)$.

Equations (5.4)–(5.6) represent Coulomb's friction law, written compactly for all contacts, where $\mathbf{U}$ is a diagonal matrix with i^{th} diagonal element equal to μ_i, the coefficient of friction (in this paper, the kinetic and static coefficients of friction are equal, so no distinction is made) at contact i, σ_i is a Lagrange multiplier arising from the conversion of the maximum dissipation condition from its “argmax” form into the inequality form given above, $\mathbf{v}_{\mathrm{t}}$ and $\mathbf{v}_{\mathrm{o}}$ are the concatenated vectors of sliding velocities for all contacts, and $\circ$ connotes the Hadamard product (*i.e.* $\mathbf{a} \circ \mathbf{b} = [a_1b_1 \ a_2b_2 \dots a_nb_n]^T$). The value of σ_i also has a physical interpretation, it is the sliding speed at contact i.

The orthogonal sliding velocity components $v_{i\mathrm{t}}$ and $v_{i\mathrm{o}}$ for this system with a kinematically controlled plate can be written as:

$$v_{i\mathrm{t}} = \mathbf{W}_{it}^T(\mathbf{q})\boldsymbol{\nu} - \mathbf{W}_{it}^T(g)\mathbf{G}^T(g)\left(\frac{\partial g}{\partial t}\right)^T \quad v_{i\mathrm{o}} = \mathbf{W}_{io}^T(\mathbf{q})\boldsymbol{\nu} - \mathbf{W}_{it}^T(g)\mathbf{G}^T(g)\left(\frac{\partial g}{\partial t}\right)^T \tag{5.8}$$

where $g(t) : \mathbb{R} \rightarrow \mathbb{R}^7$ is the vector function providing the position and orientation (unit quaternion) of the plate at time t.

5.1.2 Discrete Time Dynamics

The DCP (equations (2.35)–(2.64)) is not solved directly, but instead a time-stepping scheme is employed and the resulting (possible nonlinear) MCP is solved at each time step.

In the NCP formulation (equation (2.77)), we approximate the distance function at time $t_{\ell+1}$ as $\psi_{in}(\mathbf{q}^{\ell+1}) \approx \psi_{in}(\mathbf{q}^{\ell}, t_{\ell+1}) + h\mathbf{W}_{\mathrm{in}}^T\boldsymbol{\nu}^{\ell+1}$. The notation $\psi_{in}(\mathbf{q}^{\ell}, t_{\ell+1})$ denotes that the collision detection is done with the part at time t_ℓ and the plate at time $t_{\ell+1}$. We can do this because the kinematically controlled plate has a known function of time, and when approximating the distance function between the part and plate at time $t_{\ell+1}$, we can use the location of the plate at time $t_{\ell+1}$. This allows for the only approximation in the gap function to be the motion of the part. Note, when the part is modeled as a particle there is no rotation and, therefore, no approximation. If we evaluate $\mathbf{W}_{(\cdot)}$ and $\boldsymbol{\lambda}_{\mathrm{vp}}$ at $\ell+1$, we have a fully implicit formulation [24] with an approximation in the gap function. If we evaluate $\mathbf{W}_{(\cdot)}$ and $\boldsymbol{\lambda}_{\mathrm{vp}}$ at ℓ, we recover the Stewart-Trinkle formulation [110] with quadratic friction law. In simulations with a particle, the rotation matrices in the wrenches no longer exist and neither does $\boldsymbol{\lambda}_{\mathrm{vp}}$, resulting in a fully implicit formulation for "free." In the results presented below, we used the geometrically explicit formulation.

5.2 Results

In this section we present various results of our simulations. The first set of results compares the simulated motion of a particle to its theoretically predicted motion. Next, results comparing the trajectory error as a function of step size and as a function of the friction approximation are presented. Lastly, timing results between NCPs and LCPs of various sizes are given. The details on how we calculated the orientation of the kinematic plate is given at the end of the chapter in section 5.3.

5.2.1 Simulation Verification

5.2.1.1 Analytical Results

In this section, we show that the results of simulation using the Stewart-Trinkle time-stepper agree with the analytical results obtained by Vose *et al.* They were able to obtain closed-form solutions of particle motion for a given a 2D version of the system (shown in Figure 5.2). In this system, the plate oscillates about an axis parallel to the plate's surface and d units below it. While a block is shown in the figure, the analysis was only for a particle. Similarly, the simulation results presented in this section were obtained with a particle body. In [118], Vose *et al.* showed that particles converge to a unique velocity for each position r on the plate, which they call the asymptotic velocity at r.

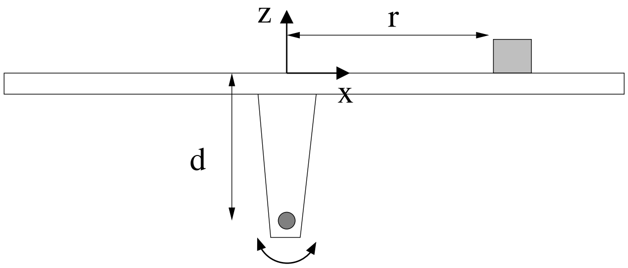


Figure 5.2: A part on a plate rotating about an axis below the plate. The fixed world frame is centered on the plate.

The particular plate motion analyzed in their paper is given by the following control function. Let $\theta(t)$ be the angle of the plate at time t with period T defined by:

$$\theta(t) = \begin{cases} \frac{1}{2}\alpha t^2 - \frac{1}{4}\alpha T t & \text{if } 0 \leq t < T/2, \\ -\frac{1}{2}\alpha t^2 + \frac{3}{4}\alpha T t - \frac{1}{4}\alpha T^2 & \text{if } T/2 \leq t < T \end{cases} \tag{5.9}$$

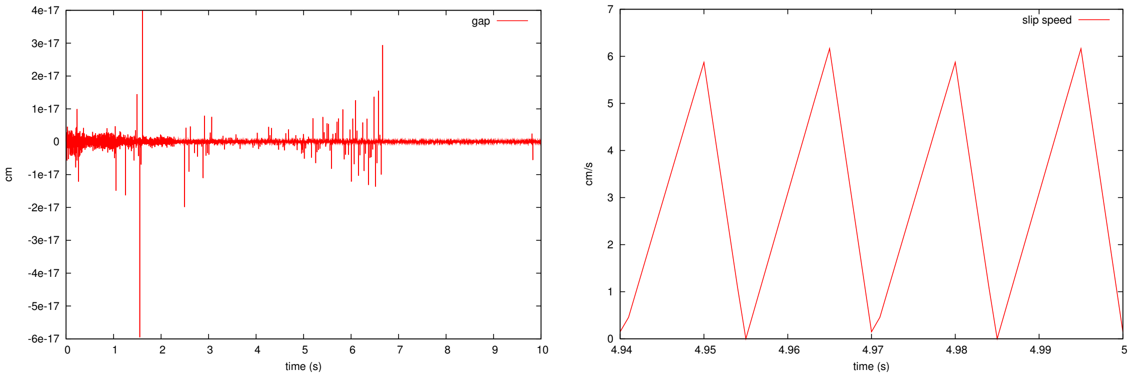
Vose *et al.* impose two restrictions in order to obtain a solution for the asymptotic velocity of the particle, it never loses contact with the plate and it never sticks to the plate. Under these assumptions, the average horizontal asymptotic velocity of the part v_{ss} can be computed as

$$v_{ss} = br + cr^3 \tag{5.10}$$

where b and c are constants defined in their paper.

The first test we performed was a comparison between their numerically computed asymptotic velocity of the part and the asymptotic velocity of the part found by our simulation. To compute the asymptotic velocity, we recorded the average velocity of the particle for each period of the plate's motion. Since the particle does not move very far in a single period of the plate's motion, we consider this average velocity an estimate of the asymptotic velocity. In addition, we also computed the average position of the particle (in the plate's frame) and used the x component of the average position in equation (5.10). In order to satisfy the required assumptions of their analysis, we set $\alpha = 180$, the period $T = .03$ seconds, and placed the axis of rotation 5cm below the plate.

Figures 5.3a and 5.3b empirically confirm that the assumptions of contact maintenance and not sticking are holding. In figure 5.3b, two cycles of the plate's periodic motion are plotted. Except for instantaneous points where the direction of slip changes, the magnitude of the slip speed is always positive. Figure 5.4 illustrates a comparison between the numerically computed horizontal asymptotic velocity determined in the work of Vose *et al.* to our simulated velocity. Note that initially the simulation velocity is much larger than the predicted steady state velocity, but over time, the particle's velocity appears to converge to the steady state velocity. This result suggests that our time-stepping method gives a faithful



(a) The distance between the particle and the plate.

(b) Slip speed between the particle and plate for a portion of the simulation.

Figure 5.3: The assumptions of contact maintenance and no sticking are satisfied.

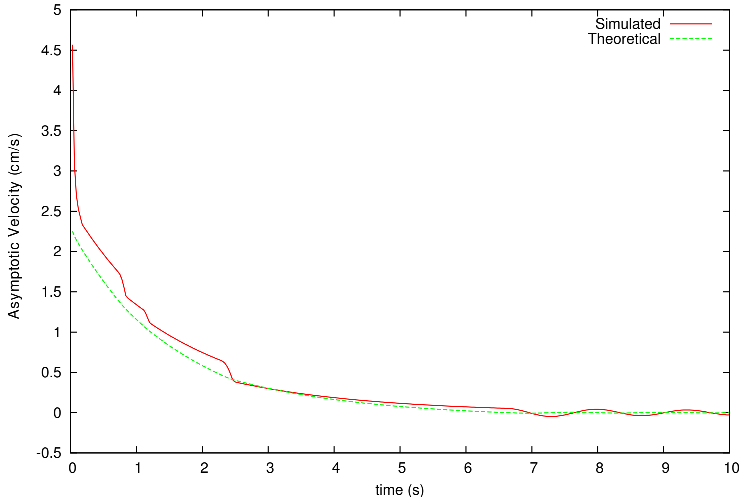
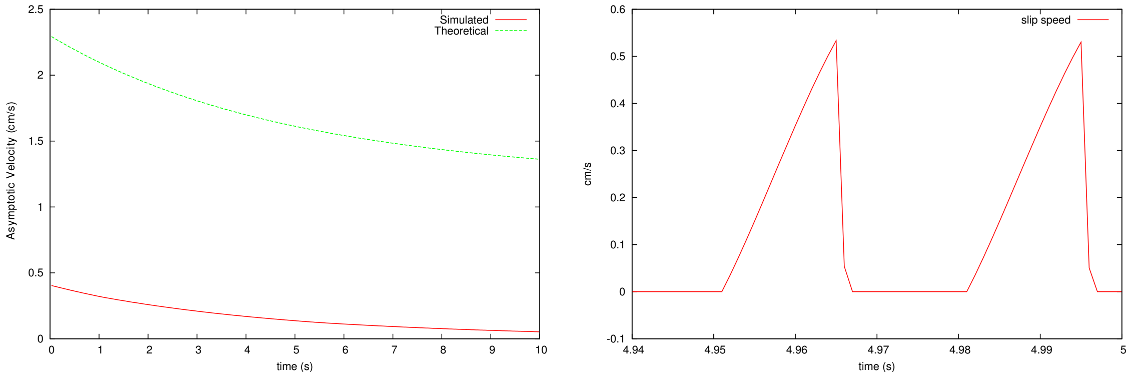


Figure 5.4: Comparison between the numerically computed asymptotic velocity (labeled simulated) to the value determined in the work of Vose *et al.* (labeled theoretical).

representation of the dynamic model of the system.

One last point of comparison of analytical and simulated results was obtained by varying the problem parameters. By simply moving the axis of rotation closer to the plate, our simulation and the experimental results of Vose *et al.* show that particle sticks during portions of each cycle (figure 5.5b) of the plate's motion. This violates the assumptions used in the analysis and thus analytical prediction of particle asymptotic motion for this case is invalid (figure 5.5a).



(a) Numerical and simulated steady state velocity

(b) Slip speed between the particle and plate for a portion of the simulation.

Figure 5.5: With the axis of rotation closer to the plate there are periods of sticking during the cycle, and the asymptotic velocity found during simulation does not match (as expected) the theoretical value determined in the work of Vose *et al.*

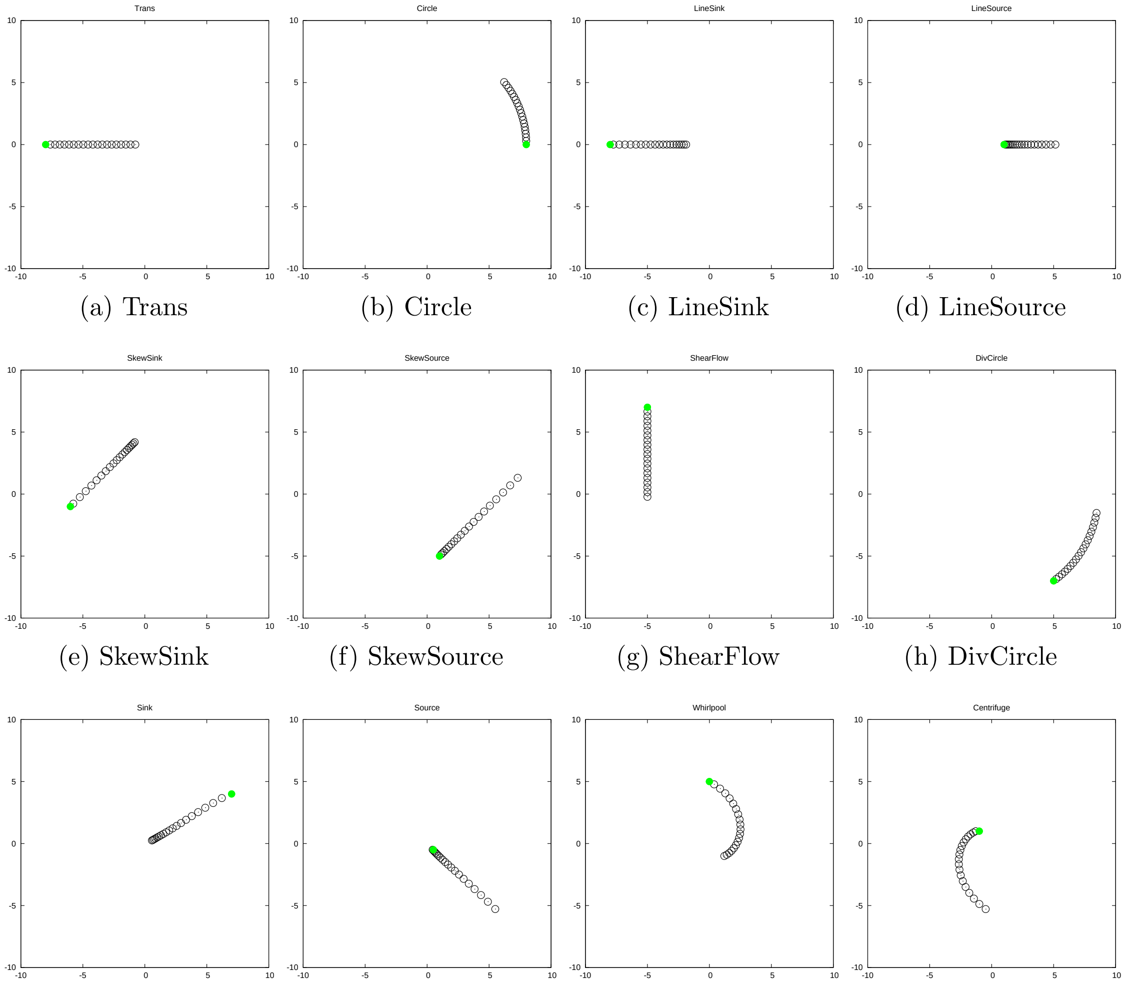
5.2.1.2 Qualitative Results

A second indication that our simulation method was producing correct results was done by qualitative comparison of our simulations with those produced by Vose *et al.* Our simulator qualitatively matched all the results in [119] (figure 5.6) and observations of their experimental system. No quantitative experimental results are available, however, Vose *et al.* kindly provided their simulation results for the Centrifuge plate motion for a direct comparison. Figure 5.7 compares the position of the particle and figure 5.8 compares the velocity of the particle for the first 0.14 seconds of simulation. There is a small discrepancy in the maximum velocity of the particle, leading to linear drift in the position. The maximum error in the particle's x-velocity is 0.0734 cm/s, which entirely accounts for the drifting in the x-position.

5.2.2 Trajectory Error as a function of step size

In this section, we look at how the trajectory of the part on the plate varies as a function of the step size used in the discretized mixed complementarity problem formulation. The part was modeled as a particle to remove surface friction effects.

We use the NCP formulation with the quadratic friction law (equation (2.77)) to remove the friction linearization error and simulate the same plate controller with



(a) Trans (b) Circle (c) LineSink (d) LineSource

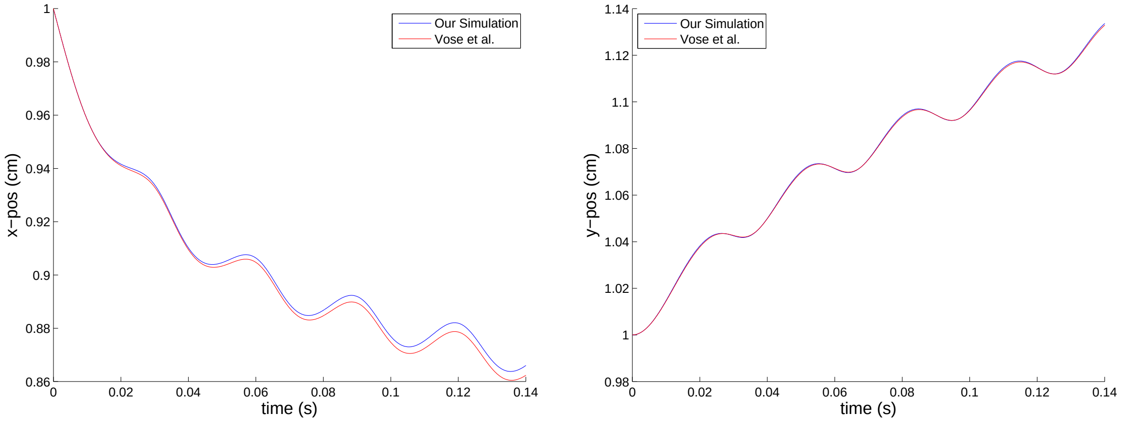
(e) SkewSink (f) SkewSource (g) ShearFlow (h) DivCircle

(i) Sink (j) Source (k) Whirlpool (l) Centrifuge

Figure 5.6: A six second simulation of all the fields described in figure 5 of [119] using a step size of 0.0001 seconds. The green circle indicates the starting location of the particle.

various values for the step size. We compute a baseline trajectory using a step size of $h = 0.00005$ seconds. Taking this trajectory as “correct”, we then compute the error in the other trajectories as the distance at time t from the base trajectory at time t. We chose to simulate the Centrifuge motion described in [119] since it has four non-zero spatial velocities (two angular and two translational) and it has been qualitatively verified. The trajectory and trajectory error as a function of step size for a 6 second simulation are shown in figure 5.9.

From figure 5.9a, the choice of a $h = 0.00005$ seconds for the baseline trajectory



(a) X-Position of particle

(b) Y-Position of particle

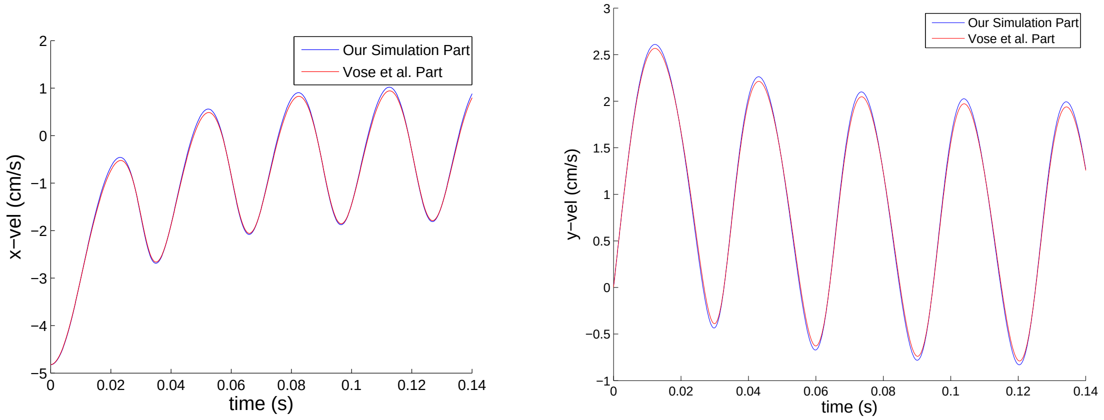
Figure 5.7: Direct comparison of the particle's position between our simulation results and those of Vose *et al.* for the Centrifuge motion.

is substantiated. Halving the step size from 0.001 to 0.0005 shows a significant improvement in the error, and reducing again (by a factor of 5) to 0.0001 also shows a large reduction in error. At this step size, the error is mostly removed and further reductions have little effect. The observation of $h = 0.0001$ seconds having small error associated with step size is why we chose to use this step size in many of the other simulations throughout this paper.

5.2.3 Trajectory Error as a function of Friction Cone linearization

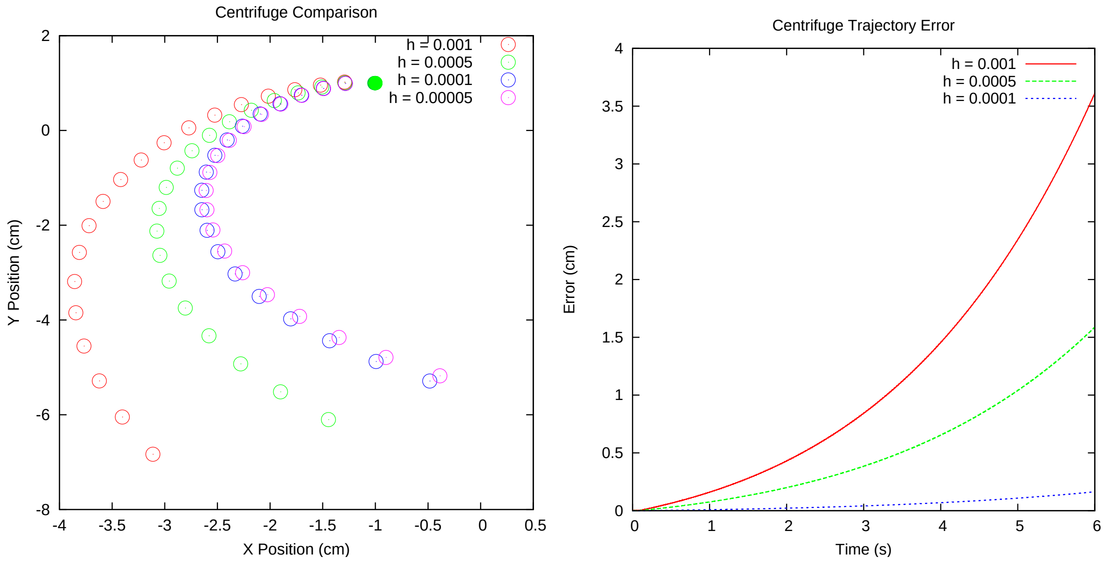
In this section, we look at how the trajectory of the part on the plate varies as a function of the friction linearization level used in the discretized mixed complementarity problem formulation. We use the same plate controllers and part model as in the previous section, but now keep the step size constant (at $h = 0.0001$ seconds) and and use a (regular) polyhedral approximation of the friction law at each tripod support point. The trajectories are parameterized by the number of edges in the polyhedral approximation. A baseline trajectory is computed by using the quadratic form of Coulomb's law at the contact point. Taking this trajectory as "correct", the error in the other trajectories is the distance at time t from the base trajectory at time t.

The first results are again for the Centrifuge motion. The trajectory and tra-



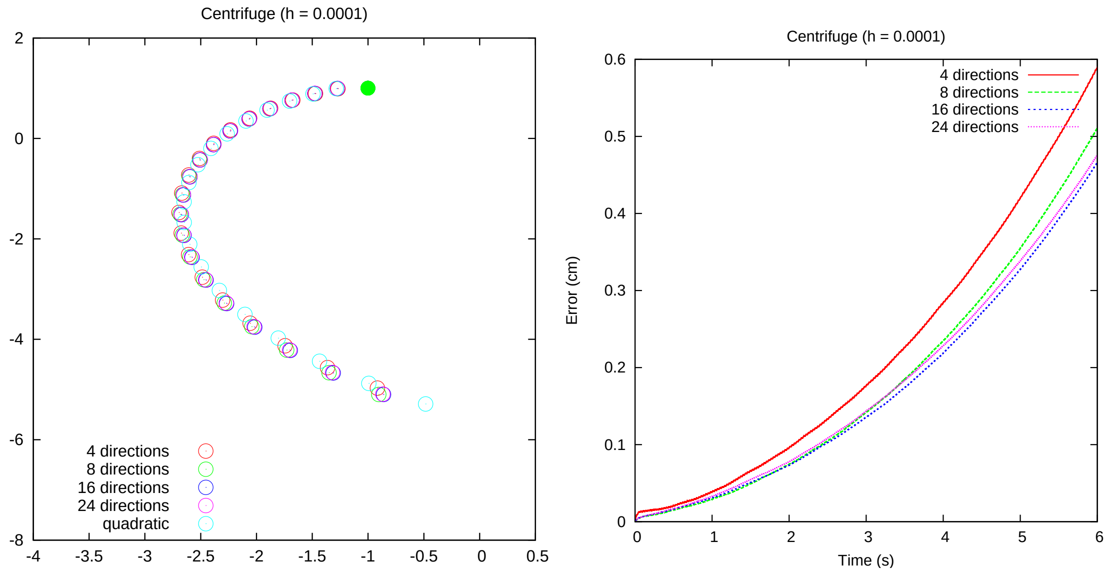
(a) X-Velocity of particle and contact point (b) Y-Velocity of particle and contact point

Figure 5.8: Direct comparison of the particle's velocity between our simulation results and those of Vose *et al.* for the Centrifuge motion.



(a) Trajectory (ticks placed at 0.3 second intervals) (b) Trajectory Error

Figure 5.9: Trajectory and error of a particle with the plate subject to the centrifuge motion. The green circle indicates the starting location of the particle. The error (Euclidean distance) is shown as a function of step size, assuming that a step size of 0.00005 is ground truth.



(a) Trajectory (ticks placed at 0.3 second intervals)

(b) Trajectory Error

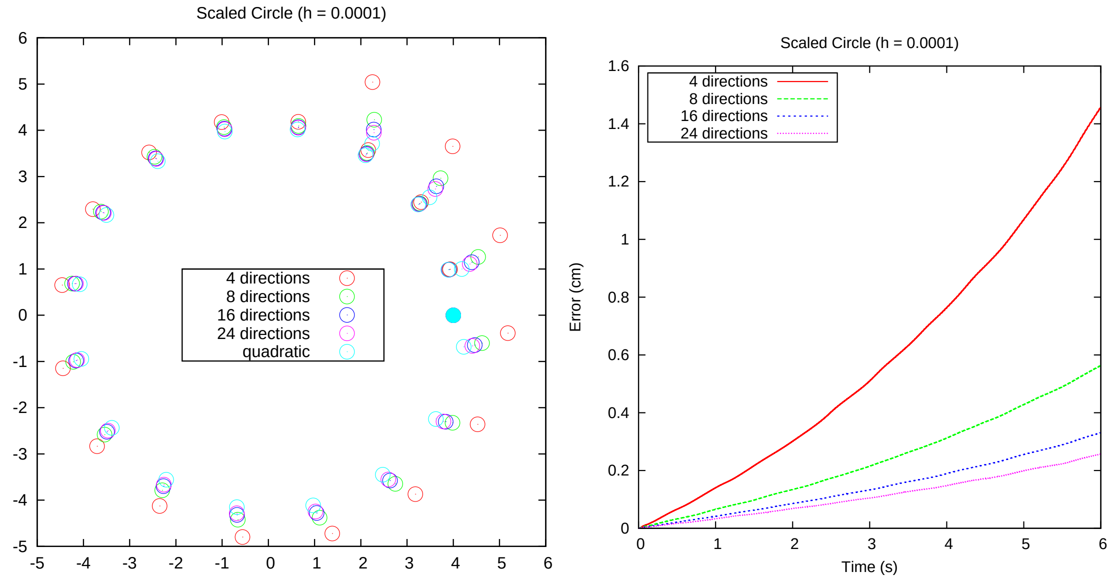
Figure 5.10: Trajectory and error caused by the polygonal approximation of the friction code. The green circle indicates the starting location of the particle. The error at time t is the Euclidean distance of the particle between the LCP and NCP formulations. All simulations used a step size of 0.0001 seconds.

jectory error as a function of the polyhedral approximation for a 6 second simulation are shown in figure 5.10. Interestingly, unlike the results parameterized by step size, increasing the number of sides in the polyhedral friction cone (figure 5.10b) does not converge to the baseline results using the "real" quadratic friction law. For this particular plate motion, is also appears that the results converge around 16 friction directions.

Based on these results we decided to generate a second set, this time using a scaled version of the Circle motion described in [119]. For the scaled circle we scaled the angular acceleration by a factor of 5, resulting in the following plate motion function:

$$\alpha_z = 5\big(100\sin(66\pi t)\big) \tag{5.11}$$

$$\ddot{p}_z = 8\sin\left(66\pi t + \frac{3}{2}\pi\right) \tag{5.12}$$



(a) Trajectory (ticks placed at 0.3 second intervals)

(b) Trajectory Error

Figure 5.11: Trajectory and error caused by the polygonal approximation of the friction code. The error at time t is the Euclidean distance of the particle between the LCP and NCP formulations. The initial position of the particle was (4,0) and all simulations used a step size of 0.0001 seconds.

Unlike the centrifuge results, in the scaled circle trajectory (figure 5.11a), increasing the number of friction directions increasingly improves the trajectory error (figure 5.11b).

5.2.4 Solution Time of Problem

Another consideration of analysis by simulation is the computation time. For example, a 5 second simulation of a tripod using the circle field controller and the LCP formulation (with 3 contacts and 32 friction directions at each contact) took slightly over 2 minutes to complete on a standard laptop (Intel Pentium M processor 1.60GHz). Conversely, the "harder" nonlinear complementarity problem formulation with quadratic friction law took a little over 20 seconds to complete. It is also clear from the simulations that even more friction directions are needed to visually duplicate the trajectory generated using quadratic friction.

The above results indicate that the NCP formulation is not only more ac-

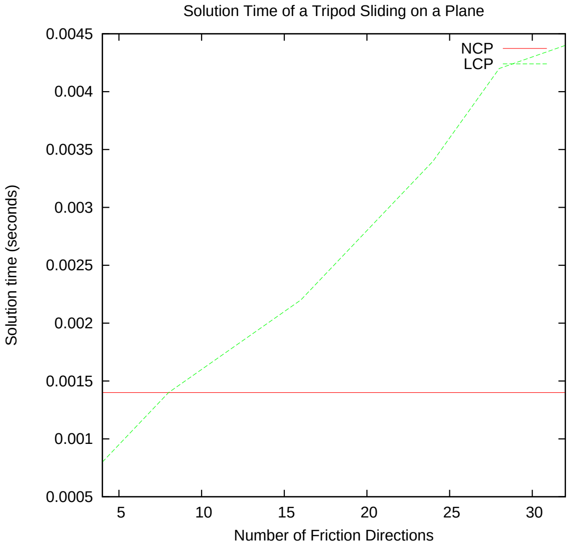


Figure 5.12: Timing of a single LCP function call for a block sliding on a plane as a function of number of friction directions. The solution time for an equivalent NCP formulation with quadratic friction law is also plotted. At approximately 8 friction directions, the solution time of the LCP is larger than the NCP.

curate, but also significantly faster than the the LCP formulation for reasonable levels of (friction cone linearization) accuracy. See the plot of the solution time of a single LCP (using the PATH solver) for various levels of friction approximation (Figure 5.12). The size of the LCP is $12 + 3d$, where d is the number of friction directions. For comparison, the solution time of the equivalent NCP is also plotted. At approximately 8 friction directions, the solution time of the LCP is larger than the NCP. This is an important finding given that for all the above simulations, 8 friction directions is insufficient for generating accurate simulations.

5.3 Computation of Plate's Orientation

The sinusoidal plate controllers in [119] are presented as acceleration trajectories. However, in our simulations we regard the plate as kinematic with a position specified solely as a function of time. Theoretically, we should be able to integrate these control laws to a position level, but in practice, when there is angular acceleration about multiple axes integrating the angular acceleration to a unit quaternion is not possible analytically. Let $\boldsymbol{\alpha}(t) = [a_1 \sin(b_1 t + c_1) \ldots a_3 \sin(b_3 t + c_3)]^T$ be the angular acceleration of the plate at time t where a_i, b_i, and c_i for $i = 1 \ldots 3$ are the parameters of the sinusoidal motion. The angular velocity $\boldsymbol{\omega}$ is trivially obtained by integrating $\boldsymbol{\alpha}$, $\boldsymbol{\omega}(t) = [-a_1 \cos(b_1 t + c_1)/b_1 \cdots - a_3 \cos(b_3 t + c_3)/b_3]^T$.

Let $\boldsymbol{\varepsilon}(t) = [e_s\ e_x\ e_y\ e_z]$ be the unit quaternion of the plate at time t. When there is only a single angular velocity component, the unit quaternion is easily (analytically) computed as

$$\boldsymbol{\varepsilon}(t) = \left[\cos\left(\frac{\theta}{2}\right)\ \sin\left(\frac{\theta}{2}\right)\mathbf{k}\right]^T \tag{5.13}$$

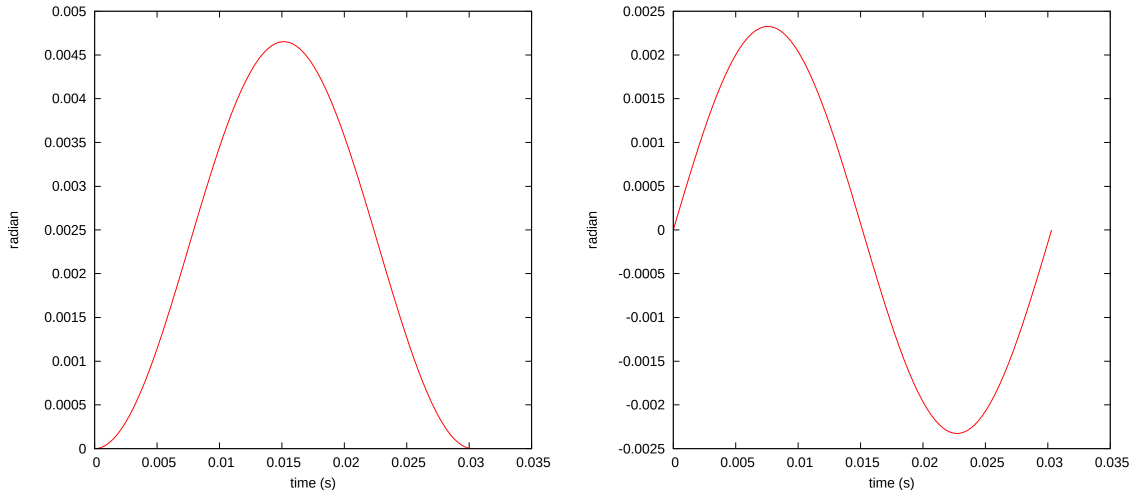
where $\theta = \dfrac{-A_i \sin(B_i t + C_i)}{B_i^2}$ and $\mathbf{k} = [1\ 0\ 0]$, $[0\ 1\ 0]$ or $[0\ 0\ 1]$ for $i = 1$, 2, or 3 respectively.

When there are multiple angular velocities, a numerical solution must be used to solve the ODE system defined in equation (2.15). In this paper, we used the standard fourth order Runge-Kutta method for solving this system. However, there is an additional requirement to this system; the rotations of the plate about each axis must be symmetric about zero. The initial value of the unit quaternion that satisfies this requirement is not known. For example, the *Centrifuge* field in [119] specifies the following angular acceleration for the plate:

$$\alpha_x = 100 \sin\left(66\pi t + \frac{1}{2}\pi\right)$$
$$\alpha_y = 100 \sin\left(66\pi t + \pi\right)$$

Figure 5.13 shows the resulting rotation angles about the x and y axes if we solve equation (2.15) with $\boldsymbol{\varepsilon}(0) = [1\ 0\ 0\ 0]$, the identity unit quaternion. The rotation about the y-axis (figure 5.13b) is symmetric about zero, but the rotation about the x-axis (figure 5.13a) is not.

To find $\boldsymbol{\varepsilon}(0)$ such that the symmetry about zero requirement is held, we add an initialization stage to the simulation. First, a simulation is performed for a single period of the plate's motion with $\boldsymbol{\varepsilon}(0)$ set to the identity unit quaternion. This trajectory is converted to Euler angles and the angle offset that makes each



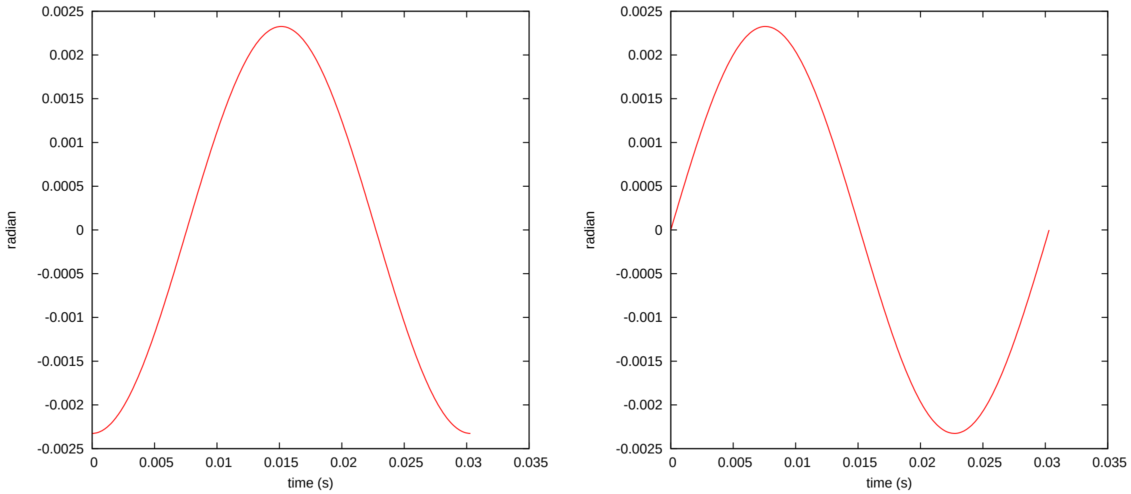
(a) Rotation About X-Axis (b) Rotation About Y-Axis

Figure 5.13: The x and y angles of rotation for one period of the centrifuge plate motion, when starting with the identity unit quaternion. The rotation about the x-axis is not symmetric about zero.

angle symmetric about zero is computed. Lastly, these three angle offsets are converted into a unit quaternion and $\boldsymbol{\varepsilon}(0)$ is the sum of the identity quaternion and the computed offset quaternion. For the Centrifuge field, $\boldsymbol{\varepsilon}(0)$ was computed to be $[0.999999,\ -0.00116298,\ 3.04478\times10^{-09},\ -4.24925\times10^{-06}]$. Figure 5.14 shows the resulting rotation angles about the x and y axes if we solve equation (2.15) with this computed initial unit quaternion.

5.4 Summary

We presented the results of a study on the accuracy of our time-stepping method on a system composed of a rigid plate and a single rigid body. It was verified that the simulation method matched the theoretical results and agreed qualitatively with simulations of problems for which only qualitative experimental results were available. After verifying our code in this manner, we studied the convergence behavior of particle trajectories as a function of step size, and found that the convergence properties of such algorithms predicted in past work of Stewart and others holds. Additionally, we studied the convergence behavior of particle trajectories as



(a) Rotation About X-Axis

(b) Rotation About Y-Axis

Figure 5.14: The x and y angles of rotation for one period of the centrifuge plate motion, when starting with the computed initial unit quaternion.

a function of the linearization of the friction cone. Our preliminary simulation experiments indicate that the nonlinear complementarity formulation is not only more accurate, but also significantly faster than the linear formulation for reasonable levels of (friction cone linearization) accuracy. This is an important finding and future work will be devoted to developing NCP solvers that are as robust and efficient as possible. With the accuracy of our time-stepper verified for this system, the next chapter addresses the design problem.

6. Planning New Motions

There are two approaches for designing new plate motions; approaches based on *initial value* problems and those based on *boundary value* problems. In both formulations, the goal is to find a feasible set of design parameters and initial conditions such that a desired part trajectory is achieved. This is related to the kinodynamic motion planning problem [33], where the goal is to solve a robot motion planning problem subject to both kinematic and dynamic constraints.

The initial value problem can be solved by breaking the large sequence of the discrete time equations of motion (equations (2.77) or equations (2.78) for $\ell = 0 \ldots T$) up into a finite number of smaller sub-problems, each solved iteratively. The design of new motions can then be done using an optimization framework where a set of initial conditions and design parameters is selected and a simulation of the current selection is done until the end time T is reached. Based on the results of the simulation, the objective function is computed and the set of initial conditions and design parameters is updated. This process continues until the objective function reaches a minimum value [103].

In the boundary value problem approach, however, the problem cannot be decoupled into a series of smaller problems. Instead, the constraints of the system are specified as boundary conditions and the entire system ($\ell = 0 \ldots T$) must be solved [99]. The boundary value approach can be further generalized by adding an objective function and considering the mixed complementarity problem as a constraint of an optimization problem with the boundary conditions as standard nonlinear programming constraints. This resulting problem is known as a mathematical program with equilibrium constraints (MPEC) [70]:

Definition 5 (Mathematical Program with Equilibrium Constraints)**.**

$$\min_{\mathbf{u}\in\mathbb{R}^{n_1},\mathbf{v}\in\mathbb{R}^{n_2}} f(\mathbf{u},\mathbf{v}) \tag{6.1}$$

$$\text{subject to: } (\mathbf{u},\mathbf{v}) \in Z, \text{ and} \tag{6.2}$$

$$\mathbf{v} \text{ solves } \mathrm{MCP}(g(\mathbf{u},\cdot),\mathbf{B}), \tag{6.3}$$

where f is a desired objective function, $Z \subseteq \mathbb{R}^{n_1+n_2}$ is a nonempty closed set (equation (6.2) represents standard nonlinear programming constraints), and equation (6.3) signifies $\mathbf{v}$ is a solution to the MCP defined by the function g and the bound set $\mathbf{B}$.

The boundary value problem and the MPEC problem are much harder to solve than the initial value problem. For one, the problem can no longer be decoupled in time and the resulting problems are much larger.

6.1 Results

In this section, two desired motions of the part on the plate are specified, a *circle* motion and a *saddle* motion. To simplify the dynamics of the problem in the optimization problem, the part is modeled as a particle. As in [119], the motion of the plate is restricted to sinusoidal functions with fixed frequency. The goal is to learn the parameters of the plate's sinusoidal functions such that the desired part motions are realized. There are 12 design parameters for this problem, the amplitude (A_i) and phase (C_i) for each of the 6 acceleration components of the plate:

$$\begin{aligned}\ddot{\mathbf{p}} &= [A_1 \sin(66\pi t + C_1) \dots A_3 \sin(66\pi t + C_3)]^T \\ \boldsymbol{\alpha} &= [A_4 \sin(66\pi t + C_4) \dots A_6 \sin(66\pi t + C_6)]^T\end{aligned}$$

where $\ddot{\mathbf{p}} = [\ddot{p}_x\ \ddot{p}_y\ \ddot{p}_z]^T$ is the translational acceleration of the plate, and $\boldsymbol{\alpha} = [\alpha_x\ \alpha_y\ \alpha_z]^T$ is the angular acceleration.

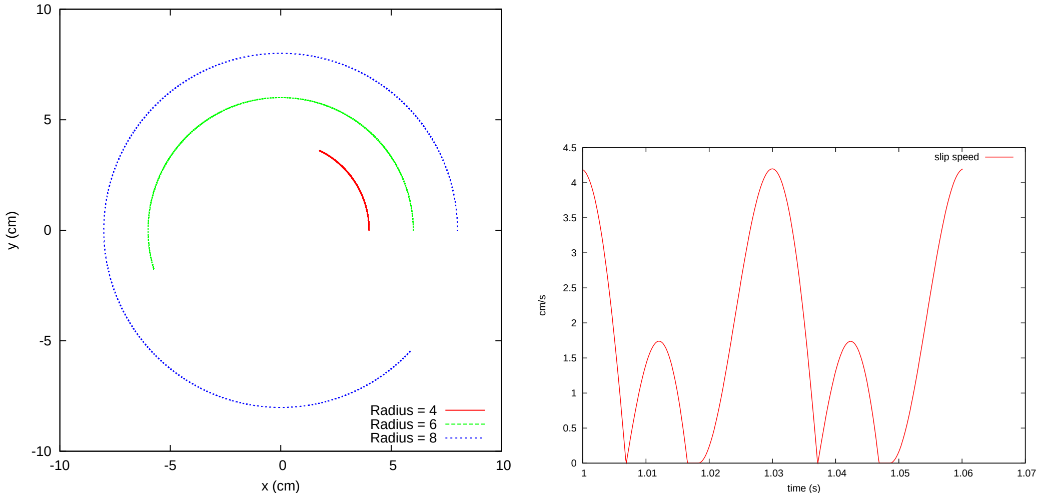
We used the initial value problem approach to find the plate motions for achieving a desired particle trajectory. The step size used in the simulation was $h = 0.0005$ and friction was modeled using Coulomb's quadratic law, resulting in an NCP. Given a selection of the 12 design parameters, a simulation is performed and a trajectory for the particle is recorded. Every 0.25 seconds the position and velocity of the particle is recorded and the error from the desired trajectory is computed. The total trajectory error is the cumulative sum of these errors. Treating the trajectory error of a simulation for a given selection of parameters as the objective function to minimize, the Nelder-Mead optimization method was used to search over the parameters and find a local minimum.

6.1.1 Circle Motion

In [119], a plate controller was presented that results in a velocity field of a circle. However, in their solution the radius of the circle must be greater than approximately 6cm, otherwise sticking occurs between the plate and the particle and the particle does not move very far. The goal in this section is to learn a plate motion such that a particle placed at a radius of 4-8cm obtains a asymptotic velocity of approximately $(-0.15y, 0.15x)$. With a desired speed of $0.15R$ cm/s, where R is the radius, it should take approximately 41.89 seconds for the particle to traverse one complete circle. The results of using the controller described in [119] with a particle placed are various initial radii are shown in figure 6.1.

Our goal is to learn a sinusoidal plate motion such that the desired velocity magnitude of $(-0.15y, 0.15x)$ is reached for a particle starting with a radius of 4-8cm. In a similar spirit to their work, we limited the search space to only three of the parameters: the amplitude of α_z, the amplitude of $\ddot{p}_z$, and phase of $\ddot{p}_z$. The other 9 parameters were set to 0. Every 0.25 seconds we recorded the position and asymptotic velocity of the particle on the plate, and the error was the sum of the position and velocity errors at each record point. The position error was the radial distance from the original radius and the asymptotic velocity error was the difference between the asymptotic velocity of the particle and the desired asymptotic velocity of $(-0.15y, 0.15x)$.

We ran two separate optimizations, one penalizing separation between the part and the plate and one not penalizing separation. The 3 unknown parameters were



(a) Trajectory

(b) Slip Speed (zoomed)

Figure 6.1: Results of the circle trajectory described in [119] with the particle starting at various radii. The simulation used a constant step size of .0001 seconds and ran for for 41.89 seconds. When the particle starts at $(4,\ 0)$, there are periods of sticking (6.1b) during a cycle of the plate's motion, and the part does not travel with the desired speed.

initialized with the values given in [119]:

$$\alpha_z = 100\sin(66\pi t) \longrightarrow A_6 = 100 \tag{6.4}$$

$$\ddot{p}_z = 8\sin(66\pi t + \frac{3}{2}\pi) \longrightarrow A_3 = 8, C_3 = \frac{3}{2}\pi \tag{6.5}$$

The first results, with the penalty on particle plate separation, took 191 iterations to converge to a local minimum with $A_6 = 62.913$, $A_3 = 9.81585$, and $C_3 = 6.33218$ with an objective function value of 29.14. The desired particle motion will have an objective value of 0, which means with the no separation constraint imposed on the system, the solver was unable to find a plate motion that achieved the desired particle motion. It was still able to find a solution better than the initial guess, but the desired goal has not been met. The sinusoidal motion of the plate found with the no separation constraint is given by:

$$\alpha_z = 62.913 \sin(66\pi t) \tag{6.6}$$

$$\ddot{p}_z = 9.81585 \sin(66\pi t + 6.33218) \tag{6.7}$$

Figure 6.2 shows the results of simulation with the learned plate motion. The learned plate motion was able to complete a full circle (plus some overshoot) when the initial radius is 8cm, however, with the restrictions of contact maintenance and limited search space it was unable to find a plate motion that worked for smaller initial radii.

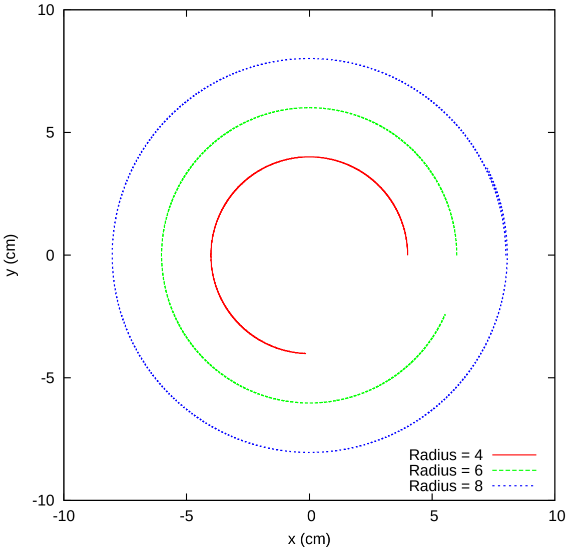


Figure 6.2: Results of the learned circle trajectory with a constant step size of .0001 and a penalty for particle plate separation.

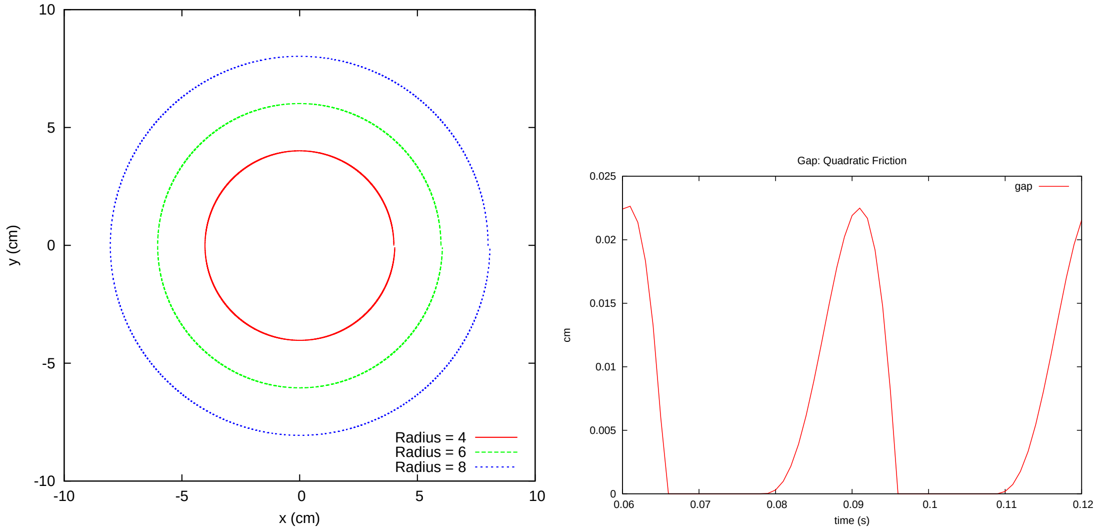
Unlike the analysis done in [119], we can allow the particle to lose and regain contact with the plate (in the presented formulation (equation (2.77)), the impacts are perfectly inelastic). Removing this restriction increases the number of possible sinusoidal plate motions available, thus providing an opportunity for the optimization algorithm to find a better solution. With this restriction removed, the solver took 163 iterations to converge to a local minimum with $A_6 = 40.7791$, $A_3 = 16.3305$, and $C_3 = 6.91229$ and an objective function value of 1.03644. This objective value is much less than the previous value found with the restrictions in place and is very close to satisfying the goal.

The corresponding sinusoidal motion of the plate is given by:

$$\alpha_z = 40.8332 \sin(66\pi t) \tag{6.8}$$

$$\ddot{p}_z = 15.3654 \sin(66\pi t + 6.5374) \tag{6.9}$$

The results of the simulation are shown in figure 6.3. With this plate motion, the particle is both sticking and breaking contact (figure 6.3b) during the simulation. However, the desired asymptotic velocity is nearly obtained for all initial starting locations of the particle and the trajectory of the particle is as desired. It's worth pointing out that this plate motion is not possible to analyze in the work of [119] since the part is both separating and sticking during the course of this particular plate motion.



(a) Trajectory

(b) Gap

Figure 6.3: Results of the learned circle velocity field without a contact maintenance requirement using a constant step size of 0.0001 seconds. The optimization problem took advantage of contact separation in the solution found.

6.1.2 Saddle Motion

In this section we present the results for learning a sinusoidal plate motion that creates a saddle velocity field, similar to the field presented in [119]. In particular, the goal was to have the y-axis act as a sink, and the x-axis act as a source. We allowed the optimization algorithm to search over all 12 parameters.

The objective function was to find a asymptotic velocity of the particle equal to $(-0.2x, 0.2y)$. The initial guess for the plate's controller parameters was:

$$A = [8\ 0.5\ -5\ -1.5\ 0\ 0]$$
$$C = [150\ 1.5\ 80\ 5.5\ 0\ 0]$$

The solver too took 3731 iterations to converge to a local minimum with $A_1 = 3.88479$, $C_1 = -1.08465$, $A_2 = -4.31188$, $C_2 = -2.0938$, $A_4 = 155.816$, $C_4 = 0.430032$, $A_5 = 204.171$, $C_5 = 5.77261$ an objective function value of 2.56985. The corresponding plate motion is:

$$\alpha_x = 3.88479 \sin(66\pi t - 1.08465) \tag{6.10}$$
$$\alpha_y = -4.31188 \sin(66\pi t - 2.0938) \tag{6.11}$$
$$\alpha_z = 0 \tag{6.12}$$
$$\ddot{p}_x = 155.816 \sin(66\pi t + 0.430032) \tag{6.13}$$
$$\ddot{p}_y = 204.171 \sin(66\pi t + 5.77261) \tag{6.14}$$
$$\ddot{p}_z = 0 \tag{6.15}$$

Figure 6.4 illustrates the results of the learned plate motion; a six second simulation of the plate motion is run for various initial staring locations of the particle.

6.2 Summary

We designed new plate motions to generate a desired part motion. We were able to specify two different velocity fields, and using an optimization problem framework, learn plate motions that realized these respective fields. By relaxing the contact restrictions required in [119], we were able to construct an improved circle field plate motion. In future work, we plan to study other sources of error and quantify their impact on simulation accuracy. For example, we will study the effect

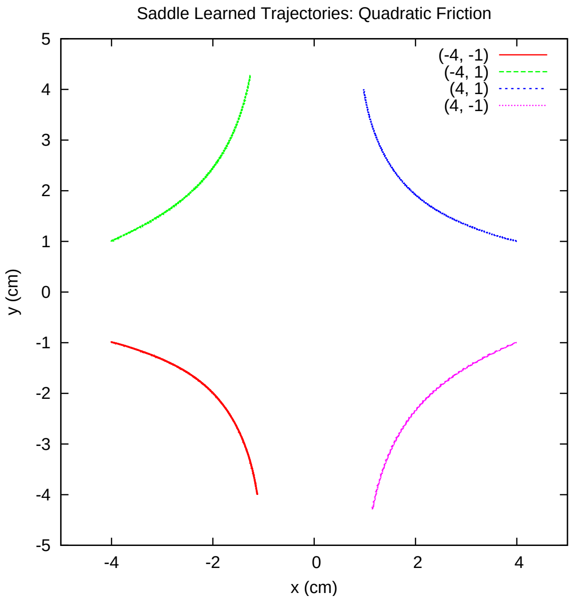


Figure 6.4: Results of the learned saddle trajectory for various initial positions and a step size of .0001.

of distributed contacts on the trajectories of 3D parts. This will lead to methods for developing adaptive step size selection for controlling simulation error. In fact, this work will go even further. We will gain a thorough understanding of errors caused by linearization of the friction cones, body geometry, and distance functions. Ultimately, we plan to develop a method that adaptively makes model linearization decisions to bound errors from those sources with an acceptable solution time. The virtual design problem is not possible without the ability to predict the parameters of the underlying complementarity problem. The next chapter addresses this problem.

7. An MPEC Formulation of Nonrecursive Filtering

In the previous chapters, we have assumed the simulation parameters are known values. In applications involving intermittent contact, such as robotic surgery or human assisting robots, these values may not be accurately known and improving the identification ability of a robot is essential. Identifying the values of these parameters increases the accuracy of simulations. Estimating the state of a dynamic system and identifying the parameters of a dynamic model are common problems in robotics, since a robot's autonomy is greatly improved when taking into account the dynamic behavior of the system.

The goal of the estimation problem is to estimate some unknown values using only a model of the system and noisy input signals. One example is estimating the position and orientation of a device from a model of the device's motion and noisy sensors (e.g. accelerometer, sonar, etc) [32]. Another example is online identification of a the parameters of a dynamic model from interacting with the environment [38].

7.1 Model

The nonlinear filtering problem is to estimate the values of a state vector $\mathbf{x}(t) \in \mathbb{R}^n$ at time t, given a set of noisy measurements $\mathbf{Y}^\ell = \{\mathbf{y}(t_1), \mathbf{y}(t_2), \ldots \mathbf{y}(t_\ell)\}$, where $\mathbf{y}^k \in \mathbb{R}^m$. The assumptions are that $\mathbf{x}(t)$ and $\mathbf{y}(t)$ evolve according to known nonlinear functions and unknown noise,

$$\mathbf{x}^{\ell+1} = F(\mathbf{x}^\ell) + G(\mathbf{x}^\ell)\boldsymbol{\varepsilon}^\ell \tag{7.1}$$

$$\mathbf{y}^\ell = H(\mathbf{x}^\ell) + \mathbf{n}^\ell \tag{7.2}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$, $G : \mathbb{R}^n \to \mathbb{R}^n$, and $H : \mathbb{R}^n \to \mathbb{R}^m$ are known nonlinear vector-valued functions (in our case F is a NCP), $\boldsymbol{\varepsilon}^\ell$ is the process noise, and $\mathbf{n}^\ell$ represents the observation or measurement noise. The process noise is a random disturbance added to effect the model propagation step due to unmodeled function parameters. Often, the noise vectors are assumed to be independent Gaussian white noise.

7.2 Filtering approaches

There are many popular filtering methods available for state estimation of constrained nonlinear systems, including Extended Kalman Filter [105], Unscented Kalman Filter [57], Particle Filter [50, 62], and Moving Horizon Estimation [93]. A full discussion of these methods is beyond the scope of this thesis, so we refer readers to the following papers for more details: [34, 121, 52, 95, 94, 31].

The basic framework of the filtering approach can be summarized in the following form,

$$\hat{\mathbf{x}}^{\ell+1} = F(\hat{\mathbf{x}}^{\ell}) + K^{\ell}(\mathbf{y}^{\ell} - H(\hat{\mathbf{x}}^{\ell})) \tag{7.3}$$

where $\hat{\mathbf{x}}^{\ell}$ is the current prediction of $\mathbf{x}^{\ell}$ and K^{ℓ} is the filter gain. Kalman famously found the optimal value for K^{ℓ} when the dynamic system is linear, and since then there has been much research improving and extending this method. For example, the EKF is an extension of the Kalman filter to nonlinear systems. It works by first linearizing the nonlinear system, and then applying the Kalman filter to the linearized system.

The linearization of the system at the current state may not accurately capture the dynamics of the nonlinear system. For example, in the EKF, the mean is propagated through the full nonlinear model, but the covariance matrix is propagated through the linearized equations. The UKF filter was introduced to avoid some of the pitfalls associated with the linearization at a single point. The UKF assumes the probability density of the state vector is Gaussian, and samples it at several specially chosen points, called sigma points. Particle filtering is related to the UKF, but instead of sampling the density function in a deterministic fashion, the points are chosen by random sampling.

7.3 Non-Recursive Filters

Unlike the recursive filters, non-recursive filters store all the measurement data and process it as a single “batch” of data. In addition, nonrecursive filters may or may not model process noise like the Kalman filters. With modern hardware the storage cost is becoming cheaper and the increased processing power is making these methods applicable to a broader range of problems.

In the moving horizon estimation approach, the computational burden is fixed by only considering a finite number, N, of previous estimates. Taking the solution of a batch of data for $x^T \dots x^{T+N}$ and determining its "arrival cost" to the solution of $x^{T+N}...x^{T+2N}$ is still an active area of research.

7.4 MPEC Formulation

In contrast to equation 7.1 where random disturbances are used to effect the model propagation step, we assume the model is a function of a set of time-invariant model parameters, $\mathbf{p}$, and simultaneously solve for the state and parameters of the system,

$$\mathbf{x}^{\ell+1} = F(\mathbf{x}^{\ell}, \mathbf{p}) \tag{7.4}$$

$$\mathbf{y}^{\ell} = H(\mathbf{x}^{\ell}) + \mathbf{n}^{\ell} \tag{7.5}$$

We can now present our non-recursive formulation for the estimation problem of a NCP system. We consider N points in a trajectory as part of one large optimization problem, which attempts to find the parameters and state that minimizes the observation error:

$$\min_{\mathbf{n}^0,\dots,\mathbf{n}^N,\ \mathbf{x}^0,\dots,\mathbf{x}^N,\ \mathbf{p}} (\mathbf{x}^0 - \bar{\mathbf{x}}^0)^T(\mathbf{x}^0 - \bar{\mathbf{x}}^0) + \sum_{\ell=0}^{T} (\mathbf{n}^{\ell})^T \mathbf{n}^{\ell} \tag{7.6}$$

$$\text{subject to:} \quad \mathbf{p} \in \mathcal{P}, \mathbf{n} \in \mathcal{N} \tag{7.7}$$

$$\mathbf{x}^{\ell+1} \in SOL(\text{MCP}(\mathbf{x}^{\ell}, \mathbf{p})) \tag{7.8}$$

$$\mathbf{y}^{\ell} = [\mathtt{I}\ \mathbf{0}]\mathbf{x}^{\ell} + \mathbf{n}^{\ell} \tag{7.9}$$

where $\bar{\mathbf{x}}^0$ is the initial state estimate, $\mathbf{n}$ is a slack variable representing the error between observation and prediction, $\mathtt{I}$ is an identity matrix of appropriate size, MCP is the mixed complementarity problem arising from the discrete time dynamics model, and $\mathcal{P}$ and $\mathcal{N}$ are the sets of possible parameter values and max observations error respectively. In this framework, enforcing the physical constraints on parameters is natural. It also removes the common assumption of Gaussian noise in the observation error, replacing it with any algebraic error bounds.

It turns out that this optimization problem is well studied in the mathematical programming community. It is known as a mathematical program with equilibrium constraints (MPEC), which we defined in chapter 6 as an alternative approach to the design problem (see definition 5).

If equations (7.6) and (7.7) are linear and equation (7.8) is an LCP, we arrive at a specialization of the MPEC known as a linear program with equilibrium constraints (LPEC). However, for the identification problem, removing the nonlinearities would force us to make unrealistic assumptions. For one, rotations would be restricted to "small" values, since the contact Jacobians are functions of the bodies' states. In addition, we would be forced to assume that either the parameters or state values are known to eliminate the bilinear constraints (*e.g.* the friction model contains the constraint $\mu\lambda_\mathrm{n}$). This last restriction in particular implies that the MPEC formulation for the estimation problem can never be an LPEC for any but the most contrived problems.

7.5 Problems with Filtering Approaches

There are two problems with the filtering techniques that our new formulation can handle easily. The first problem is filtering approaches implicitly assume that the parameters evolve with time. However, many parameters evolve with respect to the state of an object not with time. For example, if trying to predict the compliance at a contact between time-independent materials, these values will vary with the position of the contact, not with time. Another problem with these filtering techniques is they have difficulty enforcing physical constraints on parameters. For example, the coefficient of friction is always positive for physical systems, but the filtering process can set this value negative when adding Gaussian noise. There has been recent work in attempting to overcome these issues in filtering by using problem specific reparameterizations and a Johnson distribution instead of Gaussian [117]. However, these techniques are problem-specific and non-trivial to implement.

7.6 Examples

We present two dual estimation examples; the first example is solving a controlled simulation experiment of a particle falling and sliding on a frictional surface. In this example we know the exact trajectory of the particle and the coefficient of friction (since the trajectory was generated in simulation) allowing us to verify the solution. The second example is from a lab experiment of a sliding block. The block trajectory was captured from an overhead camera, and the coefficient of surface friction was unknown. Since the actual solution to this problem is unknown, we compared the results to a solution found by a standard particle filter.

7.6.1 Simulation Example

7.6.1.1 Single Particle

In this example, a simulation of a particle falling subject to a known applied force is performed. The particle begins initially at rest, eventually makes contact with the surface, and lastly slides along the surface (see figure 7.1). The simulated trajectory of the particle was recorded and noise was artificially added to each point in the trajectory. The noise was added by uniformly sampling from $[-\epsilon,\ \epsilon]$. We performed five simulations with the same initial conditions and parameters, but for each subsequent trial we would increase the value of ϵ. The goal of the experiment is to simultaneously solve for the noise-free trajectory of the particle and for the coefficient of friction, μ, between the particle and surface from the noisy trajectory data.

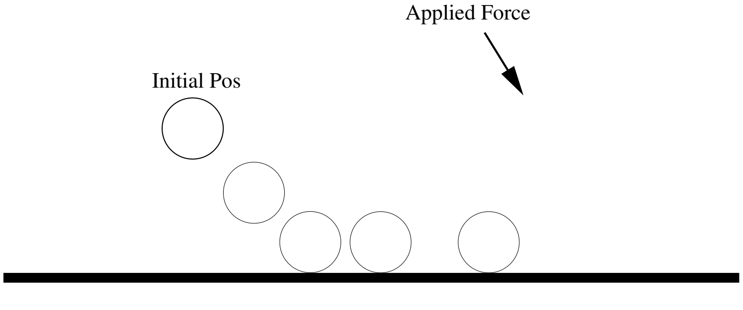


Figure 7.1: A particle begins initially at rest and not in contact with the surface. It is subject to a known applied force and eventually makes contact with the surface. Afterwards, it slides along the surface.

The observable state of the system consists of the x and y position of the particle, denoted x_p and y_p, respectively. We can also label the velocity of the particle, $\dot{x}_p$ and $\dot{y}_p$, observable since we can approximate the velocity given two adjacent position readings. The unobservable state (which must be estimated from the observed state) consists of the nonpenetration force, λ_n, friction forces $\boldsymbol{\lambda}_\mathrm{f}$, and the slip speed at the contact σ. The parameter to estimate is the coefficient of friction, μ. Taken together, the representation of this example becomes:

$$\begin{aligned}\text{State: } \mathbf{x} &= [\overbrace{x_p\ y_p\ \dot{x}_p\ \dot{y}_p}^{\text{Observable}}\ |\ \overbrace{\lambda_n\ \boldsymbol{\lambda}_\mathrm{f}\ \sigma}^{\text{Unobservable}}\ ]\\ \text{Observations: } \mathbf{y} &= [x_p\ y_p\ \dot{x}_p\ \dot{y}_p]\\ \text{Parameters: } \mathbf{p} &= [\mu]\end{aligned}$$

Mathematically, we can write the state and observation update functions, and apply bounds on our parameter,

$$\mathbf{x}^{\ell+1} \in SOL(\mathrm{MCP}(\mathbf{x}^\ell)) \tag{7.10}$$

$$\mathbf{y}^\ell = \begin{bmatrix}\mathrm{I}_{4\times4} & \mathbf{0}_{4\times4}\end{bmatrix}\mathbf{x}^\ell + \mathbf{n}^\ell \tag{7.11}$$

$$\mathcal{P} = 0 \le \mu \le 1 \tag{7.12}$$

$$\mathcal{N} = (\mathbf{n}^\ell)^T\mathbf{n}^\ell \le \epsilon^2 \tag{7.13}$$

where the MCP in equation (7.10) is given by:

$$\begin{aligned}
0 &= \mathbf{q}^{\ell+1} - \mathbf{q}^\ell - h\boldsymbol{\nu}^{\ell+1}\\
0 &= \mathbf{M}(\boldsymbol{\nu}^{\ell+1} - \boldsymbol{\nu}^\ell) - h(\mathbf{W}_\mathrm{n}\lambda_n^{\ell+1} + \mathbf{W}_\mathrm{f}\boldsymbol{\lambda}_\mathrm{f}^{\ell+1} + \boldsymbol{\lambda}_\mathrm{app})\\
&\left.0 \le \lambda_n^{\ell+1} \perp y_p^{\ell+1} \ge 0\ \right\}\ \text{Normal contact model}\\
&\left.\begin{aligned}0 &\le \boldsymbol{\lambda}_\mathrm{f}^{\ell+1} \perp \mathbf{E}\sigma^{\ell+1} + \mathbf{W}_\mathrm{f}^\mathrm{T}\boldsymbol{\nu}^{\ell+1} \ge 0\\ 0 &\le \sigma^{\ell+1} \perp \mu\lambda_n^{\ell+1} - \mathbf{E}^\mathrm{T}\boldsymbol{\lambda}_\mathrm{f}^{\ell+1} \ge 0\end{aligned}\right\}\ \text{Friction Model}
\end{aligned}$$

where $\mathbf{q} = \begin{bmatrix}x_p\\ y_p\end{bmatrix}$, $\boldsymbol{\nu} = \begin{bmatrix}\dot{x}_p\\ \dot{y}_p\end{bmatrix}$, $\mathbf{M} = \begin{bmatrix}m & 0\\ 0 & m\end{bmatrix}$, $\mathbf{W}_\mathrm{n} = \begin{bmatrix}0\\ 1\end{bmatrix}$, $\mathbf{W}_\mathrm{f} = \begin{bmatrix}-1 & 1\\ 0 & 0\end{bmatrix}$, $\mathbf{E} =$

$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$, and $\boldsymbol{\lambda}_{\text{app}} = \begin{bmatrix} x_{\text{app}} \\ -mg \end{bmatrix}$. There are four equations and four complementarity constraints per time step.

The MPEC formulation of the estimation problem can be written as:

$$\min_{\mathbf{x}^0,\ldots,\mathbf{x}^N,\mathbf{n}^0,\ldots,\mathbf{n}^N,\mu} (\mathbf{x}^0 - \bar{\mathbf{x}}^0)^T(\mathbf{x}^0 - \bar{\mathbf{x}}^0) + \sum_{\ell=0}^{N} \mathbf{n}^{\ell^T}\mathbf{n}^{\ell}$$

subject to:

$$\left.\begin{aligned} &0 = \mathbf{q}^{\ell+1} - \mathbf{q}^{\ell} - h\boldsymbol{\nu}^{\ell+1} \\ &0 = \mathbf{M}(\boldsymbol{\nu}^{\ell+1} - \boldsymbol{\nu}^{\ell}) - h(\mathbf{W}_{\text{n}}\lambda_n^{\ell+1} + \mathbf{W}_{\text{f}}\boldsymbol{\lambda}_{\text{f}}^{\ell+1} + \boldsymbol{\lambda}_{\text{app}}) \end{aligned}\right\} 4N \text{ equations}$$

$$\left.\begin{aligned} &0 \le \lambda_n^{\ell+1} \perp y_p^{\ell+1} \ge 0 \\ &0 \le \boldsymbol{\lambda}_{\text{f}}^{\ell+1} \perp \mathbf{E}\sigma^{\ell+1} + \mathbf{W}_{\text{f}}^{\text{T}}\boldsymbol{\nu}^{\ell+1} \ge 0 \\ &0 \le \sigma^{\ell+1} \perp \mu\lambda_n^{\ell+1} - \mathbf{E}^{\text{T}}\boldsymbol{\lambda}_{\text{f}}^{\ell+1} \ge 0 \end{aligned}\right\} 4N \text{ complementarity}$$

$$\left.\begin{aligned} &0 \le \mu \le 1 \\ &(\mathbf{n}^{\ell})^T\mathbf{n}^{\ell} \le \epsilon^2 \end{aligned}\right\} 4N + 2 \text{ inequalities}$$

$$\left.\mathbf{y}^{\ell} = \begin{bmatrix} \mathrm{I}_{4\times 4} & \mathbf{0}_{4\times 4} \end{bmatrix} \mathbf{x}^{\ell} + \mathbf{n}^{\ell} \right\} 4N \text{ equations}$$

We used the AMPL mathematical modeling language [44] to formulate the MPECs, and the nonlinearly constrained optimization solver KNITRO [21] to solve them. The values used in simulation were: $\boldsymbol{\mu} = \mathbf{0.2}$, $h = .05$, $\mathbf{q}^0 = [0, 3]$, $\boldsymbol{\lambda}_{app} = [5, -9.81m]$, $m = 1$, $N = 100$. Again, the goal was to simultaneously determine the value of μ and the real trajectory, from the simulated trajectory with artificial noise added. The results of KNITRO for increasing levels of artificial noise are shown in table 7.1. The initial guess provided to KNITRO was: $\mathbf{q}^{\ell} = \tilde{\mathbf{q}}^{\ell}$, $\lambda_n^{\ell} = 0$, $\boldsymbol{\lambda}_{\text{f}}^{\ell} = 0$, $\sigma^{\ell} = 0$, $\mu = 1$. KNITRO was able to recover the correct value of μ for increasing levels of observation error. The objective value of the MPEC was also driven to zero, which implies that KNITRO also found the correct trajectory. With an observation error of 0.5, KNITRO was unable to solve the problem. Even with this much noise, the solver found a very good estimate of μ (remember that μ was initialized at 1.0). Future work is required to investigate why this level of noise causes solver failure, even when such a good estimate of the parameter is found.

Observe. Error (ϵ)	μ	Obj. Val	Iterations
5.00E-05	**0.2**	2.03e-07	301
5.00E-04	**0.2**	1.61e-05	443
5.00E-03	**0.2**	1.467e-03	19
5.00E-02	**0.200022**	1.61e-01	149
5.00E-01	**0.199873**	1.48e+01	96 (infeasible pt)

Table 7.1: Results of the estimation problem for a single particle. The correct value of μ is 0.2.

7.6.1.2 Multiple Particles

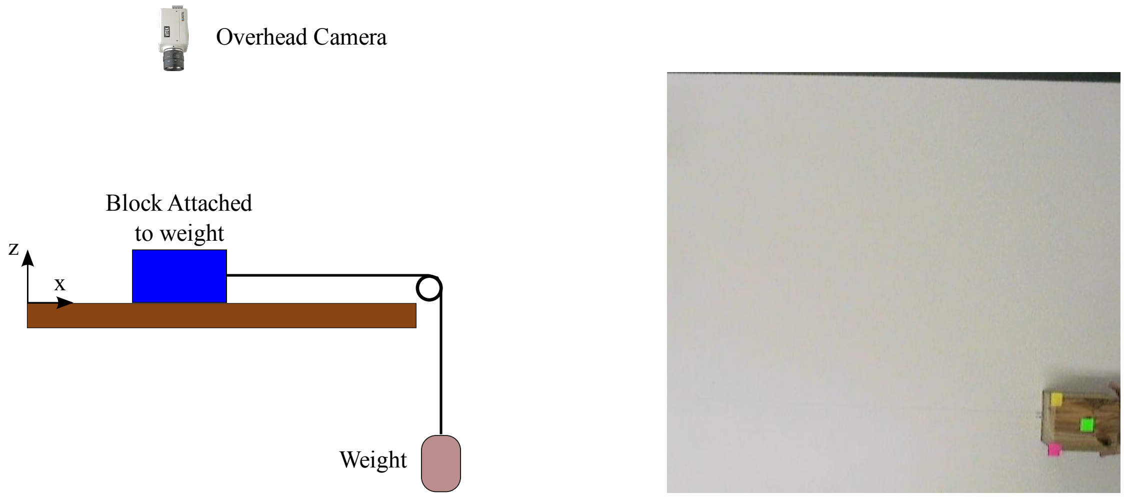
For this example, we extended the single particle example to now include multiple particles starting with random initial positions ($\mathbf{x}_{p_i} \in [-10, 10]$, $\mathbf{y}_{p_i} \in [0, 5]$) and random coefficients of friction ($\mu_i \in (0, 0.5]$). The measurement noise was uniformly sampled from the range $[-0.005, 0.005]$. The results of KNITRO for increasing number of particles are shown in table 7.2. The initial guess provided to KNITRO was: $\mathbf{q}^\ell = \tilde{\mathbf{q}}^\ell$, $\lambda_n^\ell = 0$, $\boldsymbol{\lambda}_\mathrm{f}^\ell = 0$, $\sigma^\ell = 0$, $\mu_i = 1$. Even up to 10 particles, which produces a problem with over 10,000 variables, KNITRO was able to solve for the trajectory and correct coefficient of friction for each particle. The parameter error, "μ error" is the root mean squared error for each μ_i.

# Part.	Num. Var.	**$\boldsymbol{\mu}$ Error**	Obj. Val	Iters
2	2006	**0**	4.09e-03	334
3	3009	**0**	4.95e-03	185
5	5015	**0.000006**	1.05e-02	364
10	10030	**0.0000072**	1.66e-02	281

Table 7.2: Results of the estimation problem with multiple particles. "$\boldsymbol{\mu}$ Error" is the root mean squared error.

7.6.2 Sliding Block Example

While the previous examples validated our new estimation method for a simulated example, this example tests our method on a lab experiment. In this example, a block with a known weight attached to it is sliding across a table (see figure 7.2).



(a) Schematic

(b) Camera Grab

Figure 7.2: Experimental set up. The left figure is a schematic of the experiment. The right image is a single frame from the overhead camera. The colored markers on the block were used to compute its position and orientation.

An overhead camera captures and records the trajectory of the block on the table. The block was not allowed to tumble, so we were able to model this system using our $2.5D$ formulation (section 3.3). The goal of the experiment is to simultaneously solve for the noise-free trajectory of the block and for the surface friction coefficient between the block and the table's surface from the noisy trajectory data. We modeled the surface contact as a support tripod fixed to the sliding block with linearized friction cones at each of the three contact points (see figure 7.3).

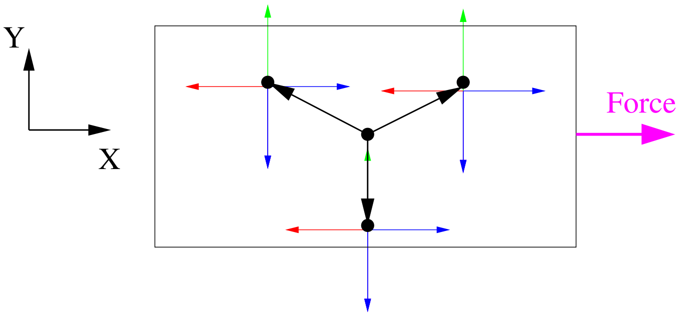


Figure 7.3: Surface friction support tripod

The observable state of the system consists of the x, y position of a frame attached to a location on the block, and the angle, θ of the frame in a fixed world frame. These values are concatenated into the configuration vector $\mathbf{q}$. We can also label the velocity of the block, $\boldsymbol{\nu}$, observable since we can approximate the velocity given two adjacent configuration readings from the camera. The unobservable state consists of the nonpenetration forces at each tripod support point, $\boldsymbol{\lambda}_{\text{n}}$, friction forces $\boldsymbol{\lambda}_{\text{f}}$ at each tripod support point, and the slip speed, $\boldsymbol{\sigma}$, at each tripod support point. The single parameter to estimate for this problem is the coefficient of friction, μ, which is assumed the same and constant at all three points. Taken together, the representation of this example becomes:

$$\begin{aligned}
\text{State: } \mathbf{x} &= [\ \overbrace{\mathbf{q}\ \boldsymbol{\nu}}^{\text{Observable}}\ \mid \overbrace{\boldsymbol{\lambda}_{\text{n}}\ \boldsymbol{\lambda}_{\text{f}}\ \boldsymbol{\sigma}}^{\text{Unobservable}}\ ] \\
\text{Observations: } \mathbf{y} &= [\mathbf{q}\ \boldsymbol{\nu}] \\
\text{Parameters: } \mathbf{p} &= [\mu]
\end{aligned}$$

The MPEC formulation of the estimation problem can be written as:

$$\min_{\mathbf{x}^0,\ldots,\mathbf{x}^N,\mathbf{n}^0,\ldots,\mathbf{n}^N,\mu} (\mathbf{x}^0 - \bar{\mathbf{x}}^0)^T(\mathbf{x}^0 - \bar{\mathbf{x}}^0) + \sum_{\ell=0}^{N} \mathbf{n}^{\ell^T}\mathbf{n}^{\ell}$$

subject to:

$$\left.\begin{aligned}
&0 = \mathbf{q}^{\ell+1} - \mathbf{q}^{\ell} - h\boldsymbol{\nu}^{\ell+1} \\
&0 = \mathbf{M}(\boldsymbol{\nu}^{\ell+1} - \boldsymbol{\nu}^{\ell}) - h(\mathbf{W}_{\text{n}}\boldsymbol{\lambda}_{\text{n}}^{\ell+1} + \mathbf{W}_{\text{f}}\boldsymbol{\lambda}_{\text{f}}^{\ell+1} + \boldsymbol{\lambda}_{\text{app}})
\end{aligned}\right\} 6N \text{ equations}$$

$$\left.\begin{aligned}
&0 \le \boldsymbol{\lambda}_{\text{n}}^{\ell+1} \perp y^{\ell+1} \ge 0 \\
&0 \le \boldsymbol{\lambda}_{\text{f}}^{\ell+1} \perp \mathbf{E}\boldsymbol{\sigma}^{\ell+1} + \mathbf{W}_{\text{f}}^{\text{T}}\boldsymbol{\nu}^{\ell+1} \ge 0 \\
&0 \le \boldsymbol{\sigma}^{\ell+1} \perp \mu\boldsymbol{\lambda}_{\text{n}}^{\ell+1} - \mathbf{E}^{\text{T}}\boldsymbol{\lambda}_{\text{f}}^{\ell+1} \ge 0
\end{aligned}\right\} \quad (6+3d)N \text{ complementarity}$$

$$\left.\begin{aligned}
&0 \le \mu \le 1 \\
&(\mathbf{n}^{\ell})^T\mathbf{n}^{\ell} \le \epsilon^2
\end{aligned}\right\} \quad 6N + 1 \text{ inequalities}$$

$$\mathbf{y}^{\ell} = \begin{bmatrix}\mathtt{I}_{6\times 6} & \mathbf{0}_{6\times 6}\end{bmatrix}\mathbf{x}^{\ell} + \mathbf{n}^{\ell} \quad \Big\} \quad 6N \text{ equations}$$

where d is the number of friction directions in the polyhedral approximation of the friction cone.

We used the AMPL to formulate the MPEC, and KNITRO to solve it. KNI-

TRO was able to find a solution, and estimated the coefficient of friction $\mu = 0.330311$. This value appears plausible, but we would like to be able to compare this solution to a well established method. Therefore, we also solved for the trajectory of the block and the coefficient of surface friction using a standard particle filtering method. Unlike the MPEC approach, there is no natural way to enforce the non-negativity constraint on μ. During the course of the filtering, if the filter ever tried to set μ to a negative number, we simply set the value to 0. This may break any theoretical convergence guarantees, but it allows us to make a quantitative comparison with another method.

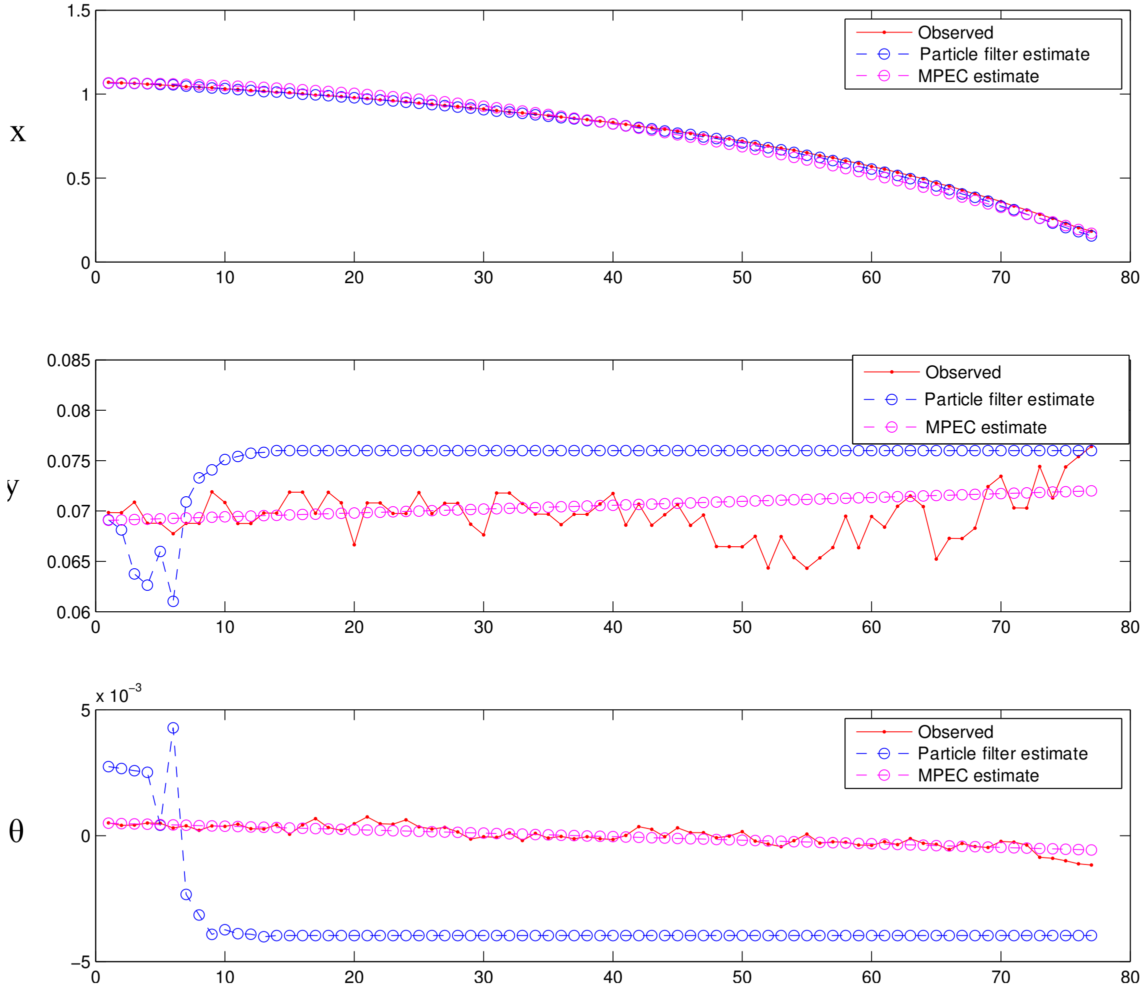


Figure 7.4: Comparison of trajectory between particle filter, MPEC and observation

In figure 7.4, a plot of the block's trajectory versus time is given for the raw camera data, the MPEC's solution, and the particle filter's solution. For all three components of the configurations, there is good agreement between the MPEC and filtered results.

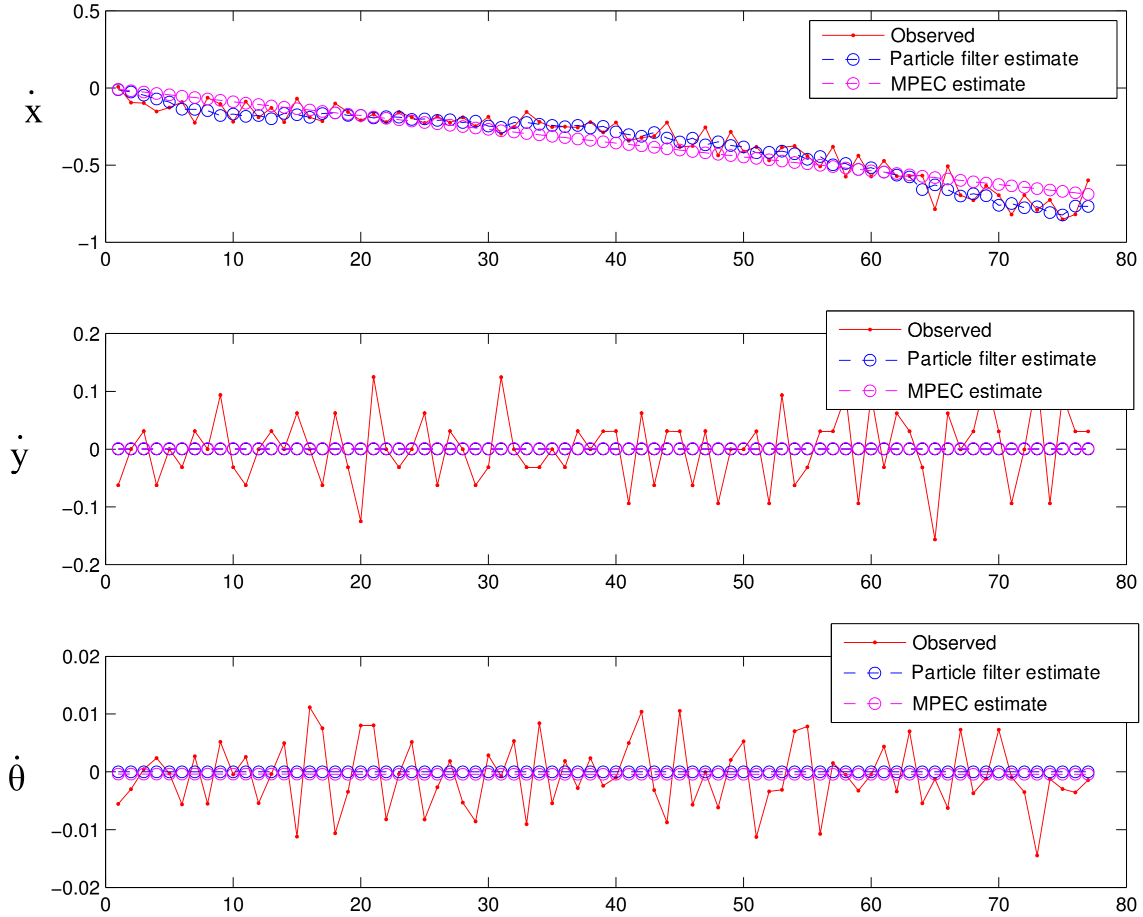


Figure 7.5: Comparison of velocity between particle filter, MPEC and observation

In figure 7.5, a plot of the block's velocity versus time is given for the raw camera data, the MPEC's solution, and the particle filter's solution. There is no distinguishable difference between the results of the MPEC and particle filter for the $\dot{y}$ and $\dot{\theta}$ components of the velocity. Comparing the $\dot{x}$ component, the two solutions agree well yet again the MPEC's solution appears qualitatively better.

We showed that the MPEC's solution for the real trajectory qualitatively matched a solution found by another method. Next we need to compare the value found for μ. In figure 7.6, the particle filter's time varying estimate of μ is plotted

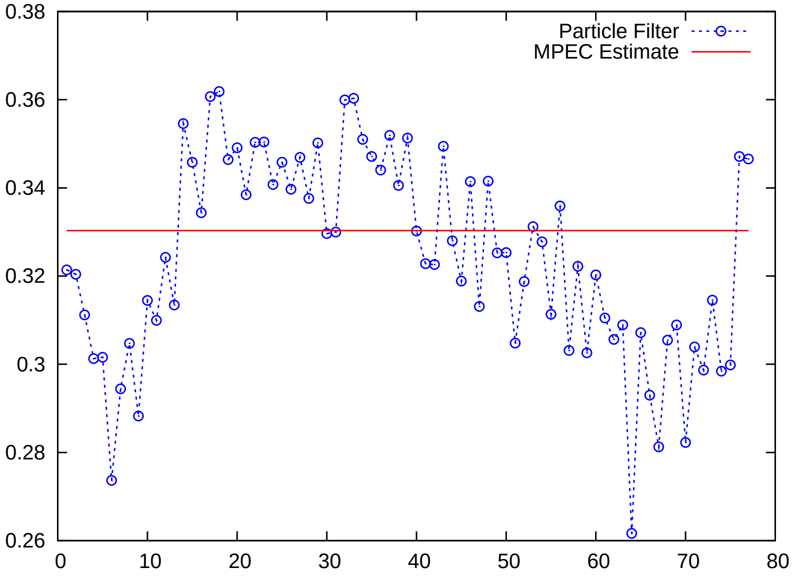


Figure 7.6: Comparison of the surface friction estimate between the particle filter and MPEC

against the MPEC's time invariant solution. The mean of the particle filter's estimate of μ, that is the average value of μ over the trajectory, is 0.3246. The MPEC's solution produced a value of $\mu = 0.330311$, qualitatively matching the value found by the filter.

The MPEC formulation presented in equations (7.6)–(7.9) does not require the entire trajectory to be included in the formulation. At some time t_ℓ, we can formulate the MPEC using only the previous k observations. Figure 7.7 shows a comparison of the MPEC's solution for levels of $k = 5$, 20, and 50 to the particle filter's solution and the full history MPEC's solution. Increasing the value of k decreases the variance in the solution found by the MPEC. This makes sense, since increasing the window size forces the parameters to become less and less time varying.

One main difference between the recursive filtering and MPEC filtering approaches is that the filtering approach is a stochastic process whereas the MPEC approach with moving window is purely algebraic. Future work will compare these two filtering approaches and better understand the similarities and differences between them.

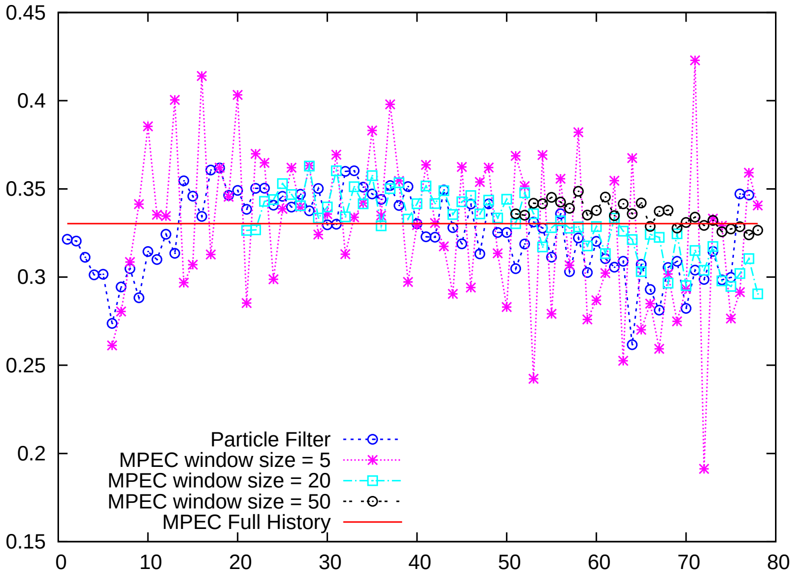


Figure 7.7: Comparison of the surface friction estimate between the particle filter, MPECs with various levels of history windows, and the MPEC with full history.

7.7 Summary

We presented our initial results of a new nonrecursive nonlinear filter for systems with complementarity constraints. In future work we plan to incorporate ideas from the moving horizon framework to this method in order to restrict the problem size. We would also like to address the question of observability of these nonlinear non-smooth systems. We would also like to design control laws that excite the parameters of interest, improving the identification problem.

8. Discussion and Conclusions

In Chapter 3, four new time-stepping formulations were presented. These formulations were constructed for a variety of reasons, including accuracy, performance, problem class, and design. We see several directions for future work. We would like to address the existence and uniqueness of solutions for the fully implicit mixed NCP we developed. For our work with non-convex polygonal constraints, we would like to extend the method to 3D bodies. In addition, when interpenetration has occurred, there is nondeterminism in the formulation that we would like to remove. We have already begun implementing these extensions, details can be found in the following tech report [82].

For all the time-stepping methods, we also wish to perform more extensive numerical experimentation. In addition, we have be restricted (in a distribution sense) by the robustness of available solvers. Even though PATH is an excellent solver capable of solving a large class of mixed complementarity problems, it is not open source and not freely distributable. We would like to develop new open-source solvers specially designed to solve the LCPs and NCPs that arise from these time-stepping formulations. It may be possible to develop time-stepper specific optimizations, which would be impossible to implement in general purpose solvers.

In Chapter 4, we presented *daVinci Code* (dVC), which is a new software tool we designed and implemented to facilitate simulation, analysis, and virtual design of multibody systems with intermittent frictional unilateral contact. We are currently extending our simulator to work for 3D systems. We also would like to develop a suite of test problems, for which we can test the accuracy of various time-stepping methods.

In Chapter 5, we presented the results of a study on the accuracy of our time-stepping method on a system composed of a rigid plate and a single rigid body. It was verified that the simulation method matched the theoretical results and agreed qualitatively with simulations of problems for which only qualitative experimental results were available. After verifying our code in this manner, we studied the convergence behavior of particle trajectories as a function of step size, and found that the convergence properties of such algorithms predicted in past work of Stewart and others holds. Additionally, we studied the convergence behavior of particle trajectories as a function of the linearization of the friction cone. Our preliminary simulation experiments indicate that the nonlinear complementarity formulation is not only more accurate, but also significantly faster than the linear formulation for reasonable levels of (friction cone linearization) accuracy. This is an important finding and future work will be devoted to developing NCP solvers that are as robust and efficient as possible.

In Chapter 6, we designed new plate motions to generate a desired part motion. We were able to specify two different velocity fields, and using an optimization problem framework, learn plate motions that realized these respective fields. By relaxing the contact restrictions required in [119], we were able to construct an improved circle field plate motion. In future work, we plan to study other sources of error and quantify their impact on simulation accuracy. For example, we will study the effect of distributed contacts on the trajectories of 3D parts. This will lead to methods for developing adaptive step size selection for controlling simulation error. In fact, this work will go even further. We will gain a thorough understanding of errors caused by linearization of the friction cones, body geometry, and distance functions. Ultimately, we plan to develop a method that adaptively makes model linearization decisions to bound errors from those sources with an acceptable solution time.

In Chapter 7, we presented our initial results of a new nonrecursive nonlinear filter for systems with complementarity constraints. In future work we plan to incorporate ideas from the moving horizon framework to this method in order to restrict the problem size. We would also like to address the question of observability of these nonlinear non-smooth systems. We would also like to design control laws that excite the parameters of interest, improving the identification problem.

REFERENCES

[1] M. Anitescu. A fixed time-step approach for multibody dynamics with contact and friction. In *International Conference on Intelligent Robots and Systems*, pages 3725–3731, Oct 2003. 61

[2] M. Anitescu. Optimization-based simulation of nonsmooth rigid multibody dynamics. *Mathematical Programming*, 105(1):113–143, 2006. 61

[3] M. Anitescu and G.D. Hart. A fixed-point iteration approach for multibody dynamics with contact and small friction. *Mathematical Programming*, 101(1):3–32, 2004. 62

[4] M. Anitescu and F.A. Potra. A time-stepping method for stiff multibody dynamics with contact and friction. *International Journal for Numerical Methods in Engineering*, 55(7):753–784, 2002. 61

[5] M. Anitescu, F.A. Potra, and D.E. Stewart. Time-stepping for three-dimensional rigid body dynamics. *Computer Methods in Applied Mechanics and Engineering*, 177(3-4):183–197, 1999. 61

[6] M. Anitescu and F.R. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *ASME Journal of Nonlinear Dynamics*, 14:231–247, 1997. 61

[7] Mihai Anitescu, James F. Cremer, and Florian A. Potra. Formulating 3D contact dynamics problems. *Mechanics of structures and machines*, 24(4):405, 1996. 80

[8] Mihai Anitescu and Alessandro Tasora. An iterative approach for cone complementarity problems for nonsmooth dynamics. *Computational Optimization and Applications*. To Appear. 21

[9] Uri M Ascher, Hongsheng Chin, Linda R Petzold, and Sebastian Reich. Stabilization of constrained mechanical systems with daes and invariant manifolds. *Mechanics of Structures and Machines*, 23(2):135–157, 1995. 37, 38

[10] D. Baraff. Issues in computing contact forces for non-penetrating rigid bodies. *Algorithmica*, pages 292–352, October 1993. 40, 61

[11] D. Baraff and A. Witkin. Dynamic simulation of non-penetrating flexible bodies. *ACM SIGGRAPH Computer Graphics*, 26(2):303–308, 1992. 62

[12] David Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. *Computer graphics*, pages 223–232, July 1989. 40

[13] David Baraff. Coping with friction for non-penetrating rigid body simulation. *Computer graphics (SIGGRAPH 1991)*, pages 31–41, 1991. 61

[14] David Baraff. Fast contact force computation for nonpenetrating rigid bodies. 28(Annual Conference Series):23–34, July 1994. 61

[15] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer methods in applied mechanics and engineering*, 1(1):1–16, 1972. 36

[16] M. S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. Wiley-Interscience, 2006. 28

[17] Stephen Berard, Binh Nguyen, Kurt Anderson, and J.C. Trinkle. Sources of error in a simulation of rigid parts on a vibrating rigid plate. *ASME Computational and Nonlinear Dynamics*. Submitted. 4, 117

[18] Stephen Berard, Binh Nguyen, Benjamin Roghani, Jeff Trinkle, Jon Fink, and Vijay Kumar. daVinci Code: A multi-model simulation and analysis tool for multi-body systems. In *IEEE International Conference on Robotics and Automation*, 2007. 1, 3, 62, 80, 102, 117

[19] S. Billups, S. Dirkse, and M. Ferris. A comparison of large scale mixed complementarity problem solvers. *Computational Optimization and Applications*, 7:3–25, 1997. 108

[20] Randy C. Brost and Alan D. Christiansen. Probabilistic analysis of manipulation tasks: A conceptual framework. *International Journal of Robotics Research*, 15(1):1–23, February 1996. ix, 2, 93, 100, 101, 103, 108, 113, 114, 115

[21] R. H. Byrd, J. Nocedal, and R. A. Waltz. KNITRO: An Integrated Package for Nonlinear Optimization. *Large-Scale Nonlinear Optimization*, pages 35–59, 2006. 149

[22] David J. Cappelleri, Jonathan Fink, and Vijay Kumar. Modeling uncertainty for planar meso-scale manipulation and assembly. In *ASME International Design Engineering Technical Conference*, 2006. 111

[23] N. Chakraborty, S. Berard, S. Akella, and J.C. Trinkle. An implicit compliant model for multibody systems with frictional intermittent contact. In *ASME International Design Engineering Technical Conferences*, October 2007. 3, 62, 80

[24] N. Chakraborty, S. Berard, S. Akella, and J.C. Trinkle. An implicit time-stepping method for multibody systems with intermittent contact. In *Robotics: Science and Systems*, June 2007. **Best Student Paper Award.** 1, 3, 42, 61, 62, 80, 119, 121

[25] A. Chatterjee. On the Realism of Complementarity Conditions in Rigid Body Collisions. *Nonlinear Dynamics*, 20(2):159–168, 1999. 61

[26] M. Cherif and K.K. Gupta. Planning quasi-static fingertip manipulation for reconfiguring objects. 15(5):837–848, 1999. 1

[27] J.C.K. Chou. Quaternion kinematic and dynamic differential equations. *IEEE Transactions on Robotics*, 8(1), 1992. 13

[28] Michael B. Cline. Rigid body simulation with contact and constraints. Master's thesis, University of British Columbia, November 2002. 24, 61

[29] R. W. Cottle, J.S. Pang, and R. E. Stone. *The Linear Complementarity Problem.* Academic Press, 1992. 5, 18, 19

[30] R.W. Cottle and G.B. Dantzig. Complemetary Pivot Theory of Mathematical Programming. *Linear Algebra and Its Applications*, 1:103–125, 1968. 19

[31] F. Daum and R. Co. Nonlinear filters: beyond the Kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8 Part 2):57–69, 2005. 144

[32] M. Dissanayake, P. Newman, S. Clark, HF Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001. 143

[33] Bruce Donald, Patrick Xavier, John Canny, and John Reif. Kinodynamic motion planning. *Journal of the ACM*, 40(5):1048–1066, 1993. 135

[34] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000. 144

[35] Christian Duriez, Frédéric Dubois, Abderrahmane Kheddar, and Claude Andriot. Realistic Haptic Rendering of Interacting Deformable Objects in Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):36–47, 2006. 21

[36] K.T. Egan, Stephen Berard, and J.C. Trinkle. Modeling nonconvex constraints using linear complementarity. Technical Report 03-13, Department of Computer Science, Rensselaer Polytechnic Institute, 2003. 3, 62, 64, 66, 101

[37] Michael Erdmann. On a Representation of Friction in Configuration Space. *International Journal of Robotics Research*, 13(3):240–271, 1994. 61

[38] D. Erickson, M. Weber, and I. Sharf. Contact stiffness and damping estimation for robotic systems. *The International Journal of Robotics Research*, 22(1):41, 2003. 143

[39] Kenny Erleben. *Stable, Robust, and Versatile Multibody Dynamics Animation.* PhD thesis, University of Copenhagen (DIKU), March 2005. 24, 61

[40] Kenny Erleben. Velocity-based shock propagation for multibody dynamics animation. *ACM Trans. Graph.*, 26(2):12, 2007. 20, 21, 61

[41] Joel M. Esposito, Vijay Kumar, and George J. Pappas. Accurate event detection for simulating hybrid systems. In *International Workshop on Hybrid Systems*, pages 204–217, London, UK, 2001. Springer-Verlag. 34

[42] Michael C. Ferris. Cpnet software, 2005. [accessed 20-Sept-2005]. 108

[43] Michael C. Ferris and Todd S. Munson. Complementarity problems in GAMS and the PATH solver. *Journal of Economic Dynamics and Control*, 24(2):165–188, 2000. 108, 117

[44] Robert Fourer, David M. Gay, and Brian W. Kernighan. *A Modeling Language for Mathematical Programming.* Duxbury Press, 2003. 149

[45] B. Gavrea, M. Anitescu, and F.A. Potra. Convergence of a class of semi-implicit time-stepping schemes for nonsmooth rigid multibody dynamics. *SIAM Journal on Optimization*, To Appear. 104, 117

[46] Frank Génot and Bernard Brogliato. New results on Painlevé paradoxes . *European journal of mechanics*, 18(4):653–677, 1999. 61

[47] C. Glocker and F. Pfeiffer. An lcp-approach for multibody systems with planar friction. In *Contact Mechanics Int. Symposium*, pages 13–30, 1992. 61

[48] C. Glocker and F. Pfeiffer. Multiple impacts with friction in rigid multibody systems. *Nonlinear Dynamics*, 7:471–497, 1995. 61

[49] O.C. Goemans, K. Goldberg, and A.F. Stappen. Blades for feeding 3D parts on vibratory tracks. *Assembly Automation*, 26(3):221–226, 2006. 1

[50] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, Apr 1993. 144

[51] E. Guendelman, R. Bridson, and R. Fedkiw. Nonconvex rigid bodies with stacking. In *International Conference on Computer Graphics and Interactive Techniques*, pages 871–878. ACM Press New York, NY, USA. 21, 61

[52] E.L. Haseltine and J.B. Rawlings. Critical evaluation of extended Kalman filtering and moving-horizon estimation. *Industrial and Engineering Chemistry Research*, 44(8):2451–2460, 2005. 144

[53] E. J. Haug, S. C. Wu, and S. M. Yang. Dynamics of mechanical systems with Coulomb friction, stiction, impact and constraint addition-deletion. III. *Mechanism and Machine Theory*, 21(5):365–446, 1986. 35

[54] K.K. Hauser, C. Shen, and J.F. OBrien. Interactive Deformation Using Modal Analysis with Constraints. In *Graphics Interface Proceedings 2003: Canadian Human-Computer Communications Society*. AK Peters, Ltd., 2003. 62

[55] E. Johnson and T.D. Murphey. Discrete and continuous mechanics for tree representations of mechanical systems. In *IEEE International Conference on Robotics and Automation*, 2008. 1

[56] F. Jourdan, P. Alart, and M. Jean. A Gauss-Seidel like algorithm to solve frictional contact problems. *Computer Methods in Applied Mechanics and Engineering*, 155(1-2):31–47, 1998. 21, 61

[57] Simon J. Julier and Jeffrey K. Uhlmann. A New Extension of the Kalman Filter to Nonlinear Systems. pages 182–193, 1997. 144

[58] Thomas R. Kane and David A. Levinson. *Dynamics: Theory and Applications*. McGraw-Hill, 1985. 12

[59] Yoshihiko Nakamura Katsu Yamane. A numerically robust LCP solver for simulating articulated rigid bodies in contact. In *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland, June 2008. 19, 20

[60] Danny M. Kaufman, Shinjiro Sueda, Doug L. James, and Dinesh K. Pai. Staggered projections for frictional contact in multibody systems. *ACM Trans. Graph.(Proc. SIGGRAPH Asia)*, 27, 2008. 21, 61, 80

[61] E. Klavins. Programmable self-assembly. *Control Systems Magazine*, pages 43–56, August 2007. 1

[62] A. Kong, J.S. Liu, and W.H. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, pages 278–288, 1994. 144

[63] Peter R. Kraus, Anders Fredriksson, and Vijay Kumar. Modeling of frictional contacts for dynamic simulation. In *IROS Workshop on Dynamic Simulation: Methods and Applications*, 1997. 33, 88

[64] C. Lacoursière. Splitting Methods for Dry Frictional Contact Problems in Rigid Multibody Systems: Preliminary Performance Results. In *Conference Proceedings from SIGRAD2003, November*, pages 20–21, 2003. 61

[65] C. E. Lemke. Bimatrix equilibrium points and mathematical programming. *Management science*, 11(7):681–689, 1965. 19

[66] T. Liu and M. Y. Wang. Computation of three-dimensional rigid-body dynamics with multiple unilateral contacts using time-stepping and Gauss-Seidel methods. *IEEE Transactions on Automation Science and Engineering*, 2(1):19–31, January 2005. 35, 61, 80, 95, 96

[67] Per Lötstedt. Coulomb friction in two-dimensional rigid-body systems. *Zeitschrift für angewandte mathematik und mechanik*, 61:605–615, 1981. 60

[68] Per Lötstedt. Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM Journal on Applied Mathematics*, 42(2):281–296, April 1982. 60

[69] Per Lötstedt. Numerical simulation of time-dependent contact and friction problems in rigid body mechanics. *SIAM journal on scientific and statistical computing*, 5(2):370–393, 1984. 61

[70] Z.Q. Luo, J.S. Pang, and D. Ralph. *Mathematical Programs With Equilibrium Constraints*. Cambridge University Press, Cambridge, England, 1996. 5, 135

[71] M. T. Mason. *Manipulator Grasping and Pushing Operations*. PhD thesis, Massachusetts Institute of Technology, June 1982. Reprinted in *Robot Hands and the Mechanics of Manipulation*, MIT Press, Cambridge, Massachusetts, 1985. 66

[72] Matthew T. Mason. *Mechanics of robotic manipulation*. MIT Press, 2001. 8, 17

[73] Matthew T. Mason and Yu Wang. On the inconsistency of rigid-body frictional planar mechanics. In *IEEE International Conference on Robotics and Automation*, 1988. 61

[74] V.J. Milenkovic and H. Schmidl. Optimization-based animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 37–46. ACM New York, NY, USA, 2001. 61

[75] Andrew T. Miller and Peter K. Allen. GraspIt!: A versatile simulator for grasp analysis. In *Proceedings ASME International Mechanical Engineering Congress and Exposition*, pages 1251–1258, Orlando, FL, 2000. 1

[76] Brian Mirtich and John Canny. Impulse-based simulation of rigid bodies. In *Proceedings of the 1995 symposium on Interactive 3D graphics*. ACM Press, 1995. 40

[77] Todd Murphey and Vijay Kumar. Workshop on Contact Models for Manipulation and Locomotion. personal communications, 2008. 2

[78] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994. 17

[79] Katta G. Murty. Note on a Bard-Type Scheme for Solving the Complementarity Problem. *Opsearch*, 11(2–3):123–130, 1974. 19

[80] Katta G. Murty. *Linear Complementarity, Linear and Nonlinear Programming Internet Edition*. Helderman-Verlag, 1988. 19, 20

[81] D Negrut, EJ Haug, and HC German. An implicit rungekutta method for integration of differential algebraic equations of multibody dynamics. *Multibody system dynamics*, 9(2):121–142, 2003. 39

[82] Binh Nguyen, Stephen Berard, and J.C. Trinkle. Modeling nonconvex configuration space using linear complementarity problems. Technical Report 08-11, Department of Computer Science, Rensselaer Polytechnic Institute, 2008. 64, 66, 103, 157

[83] P. Painlevé. Sur les lois du frottement de glissement. *Comptes Rendus de l'Academie des Sciences*, 121:112–115, 1895. 61

[84] Jong-Shi Pang. On the Convergence of a Basic Iterative Method for the Implicit Complementarity Problem. *Optimization Theory and Applications*, 37(2), 1982. 21

[85] Jong-Shi Pang and Francisco Facchinei. *Finite-Dimensional Variational Inequalities and complementarity Problems (I)*. Springer Verlag, New York, 2003. 18

[86] J.S. Pang and J.C. Trinkle. Complementarity formulations and existence of solutions of dynamic multi-rigid-body contact problems with coulomb friction. *Mathematical Programming*, 73:199–226, 1996. 61

[87] J.S. Pang, J.C. Trinkle, and G. Lo. A complementarity approach to a quasistatic rigid body motion problem. *Journal of Computational Optimization and Applications*, 5(2):139–154, March 1996. 62, 70

[88] Mark Pauly, Dinesh K. Pai, and Leonidas J. Guibas. Quasi-rigid objects in contact. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, August 2004. 34, 62, 81

[89] F. Pfeiffer and C. Glocker. *Multibody Dynamics with Unilateral Contacts.* Wiley Series in Nonlinear Science, New York, 1996. 61

[90] Erion Plaku, Kostas E. Bekris, and Lydia E. Kavraki. Oops for motion planning: An online, open-source, programming system. In *IEEE International Conference on Robotics and Automation*, 2007. 1

[91] Florian A. Potra. Implementation of linear multistep methods for solving constrained equations of motion. *SIAM Journal on Numerical Analysis*, 30(3):774–789, 1993. 36

[92] F.R. Potra, M. Anitescu, B. Gavrea, and J.C. Trinkle. Linearly implicit trapezoidal method for integrating stiff multibody dynamics with contact, joints, and friction. *International Journal for Numerical Methods in Engineering*, 2007. 61

[93] C.V. Rao and J.B. Rawlings. Nonlinear moving horizon state estimation. *Nonlinear Model Predictive Control*, page 45, 2000. 144

[94] CV Rao, JB Rawlings, and DQ Mayne. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE Transactions on Automatic Control*, 48(2):246–258, 2003. 144

[95] J.B. Rawlings and B.R. Bakshi. Particle filtering and moving horizon estimation. *Computers and chemical engineering*, 30(10-12):1529–1541, 2006. 144

[96] DS Reznik and JF Canny. C'mon part, do the local motion! In *IEEE International Conference on Robotics and Automation*, 2001. 1

[97] J. Sauer and E. Schömer. A constraintbased approach to rigid body dynamics for virtual reality applications. In *ACM Symposium on Virtual Reality Software and Technology*, 1998. 61

[98] R. Smith. Open dynamics engine, 2008. http://www.ode.org. 1

[99] P. Song, V. Kumar, and J. S. Pang. A two-point boundary-value approach for planning manipulation tasks. In *Robotics: Science and Systems*, Cambridge, Massachusetts, June 2005. 1, 135

[100] Peng Song, Peter Kraus, Vijay Kumar, and Pierre Dupont. Analysis of rigid-body dynamic models for simulation of systems with frictional contacts. *ASME Journal of Applied Mechanics*, 68(1):118–128, 2001. 33, 81

[101] Peng Song and Vijay Kumar. Distributed compliant model for efficient dynamic simulation of systems with frictional contacts. In *ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Sept 2003. 62, 81

[102] Peng Song, Jong-Shi Pang, and Vijay Kumar. A semi-implicit time-stepping model for frictional compliant contact problems. *International Journal for Numerical Methods in Engineering*, 60:2231–2261, 2004. 33, 62, 80, 81, 90

[103] Peng Song, J.C. Trinkle, Vijay Kumar, and Jong-Shi Pang. Design of part feeding and assembly processes with dynamics. In *International Conference on Robotics and Automation*, pages 39–44, 2004. 1, 28, 108, 135

[104] K. W. Spring. Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: a review. *Mechanism and machine theory*, 21(5), 1986. 13

[105] R.F. Stengel. *Optimal control and estimation.* Courier Dover Publications, 1994. 144

[106] David E. Stewart. Existence of solutions to rigid body dynamics and the Painlevé paradoxes. *Comptes Rendus de l'Acadmie des Sciences - Series I - Mathematics*, 325(6):689 – 693, 1997. 61

[107] David E. Stewart. Rigid-body dynamics with friction and impact. *SIAM Rev.*, 42(1):3–39, 2000. 25, 61

[108] David E. Stewart. Finite-dimensional contact mechanics. *Philosophical transactions- Royal Society. Mathematical, physical and engineering sciences*, 359(1789):2467–2482, 2001. 62

[109] D.E. Stewart. Convergence of a timestepping scheme for rigidbody dynamics and resolution of Painlevé's problem. *Archive for Rational Mechanics and Analysis*, 145(3):215–260, 1998. 4, 61, 104, 117

[110] D.E. Stewart and J.C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal of Numerical Methods in Engineering*, 39:2673–2691, 1996. 30, 41, 42, 60, 61, 80, 117, 121

[111] Shinjiro Sueda, Andrew Kaufman, and Dinesh K. Pai. Musculotendon simulation for hand animation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 27(3), 2008. 1

[112] J.C. Trinkle, Stephen Berard, and J.S. Pang. A Time-Stepping Scheme for Quasistatic Multibody Systems. In *IEEE International Symposium on Assembly and Task Planning*, July 2005. 3, 62, 70, 119

[113] J.C. Trinkle, J.S. Pang, S. Sudarsky, and G. Lo. On dynamic multi-rigid-body contact problems with Coulomb friction. *Zeitschrift für Angewandte Mathematik und Mechanik*, 77(4):267–279, 1997. 28, 35, 41, 61, 119

[114] J.C. Trinkle, R.C. Ram, A.O. Farahat, and P.F. Stiller. Dexterous manipulation planning and execution of an enveloped slippery workpiece. In *Proceedings, IEEE International Conference on Robotics and Automation*, volume 2, pages 442–448, May 1993. 1

[115] J.C. Trinkle, J.A. Tzitzouris, and J.S. Pang. Dynamic multi-rigid-body systems with concurrent distributed contacts. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 359(1789):2575–2593, December 2001. 27, 61, 95, 96

[116] J.A. Tzitzouris. *Numerical resolution of frictional multi-rigid-body systems via fully implicit time-stepping and nonlinear complementarity.* PhD thesis, The John Hopkins Univerisy, 2001. 61

[117] D. Verscheure, J. Swevers, H. Bruyninckx, and J. Schutter. On-line identification of contact dynamics in the presence of geometric uncertainties. *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 851–856, May 2008. 146

[118] T. H. Vose, P. Umbanhowar, and K. M. Lynch. Vibration-induced frictional force fields on a rigid plate. In *IEEE International Conference on Robotics and Automation*, 2007. 117, 118, 122

[119] T. H. Vose, P. Umbanhowar, and K. M. Lynch. Friction-induced velocity fields for point parts sliding on a rigid oscillated plate. In *Robotics: Science and Systems*, 2008. x, xi, 1, 117, 118, 125, 126, 129, 131, 132, 136, 137, 138, 139, 140, 141, 158

[120] T. H. Vose, P. Umbanhowar, and K. M. Lynch. Friction-induced lines of attraction and repulsion for parts sliding on an oscillated plate. *IEEE Transactions on Automation Science and Engineering*, To Appear. 117, 118

[121] E.A. Wan and R. Van Der Merwe. The Unscented Kalman Filter for nonlinear estimation. In *Proceedings of Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control*, pages 153–158, 2000. 144

[122] Yu Wang and Matthew Mason. Modeling impact dynamics for robotic operations. In *IEEE International Conference on Robotics and Automation*, 1987. 40

[123] Alan Watt and Mark Watt. *Advanced Animation and Rendering Techniques.* ACM, 1992. 8, 9

[124] N.B. Zumel and M.A. Erdmann. Nonprehensible two palm manipulation with non-equilbrium transitions between stable states. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 3317–3323, April 1996. 1