Sources of Error in a Simulation of Rigid Parts on a Vibrating Rigid Plate [*]

Stephen Berard[†] **Binh Nguyen**[†] **Kurt Anderson**[‡] **J.C. Trinkle**[†]

[†]Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

[‡]Department of Mechanical, Aerospace
and Nuclear Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

Abstract

We present a simulation study of an important rigid body contact problem. The system in question is composed of a rigid plate and a single rigid body (or particle). The plate follows a prescribed periodic motion of small amplitude and high frequency, such that the net force applied to the part appears 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 the Stewart-Trinkle 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. Specifically, we were able to demonstrate the convergence of the method with decreasing step size (as predicted theoretically by Stewart). Further analytical and numerical studies will be carried out in the future to develop and select robust simulation methods that best satisfy the speed and accuracy requirements of different applications.

1 Introduction

As in many other fields, simulation is becoming an increasingly important tool for supporting research in robotics. Not surprisingly, many of the important problems that could yield to closed-form analysis have been solved and studied thoroughly. Problems characterized by intermittent contact is one particularly important type of robotics problems for which research must rely on simulation techniques [1, 2, 3, 4, 5, 6]. Evidence of the need for simulation is the recent trend of robotics researchers studying grasping, assembly, and dexterous manipulation problems using Open Dynamic Engine (ODE) [7]. ODE was developed for computer game applications, which has led its developers to trade physical accuracy for simulation speed and believable results. The fact is that researchers using ODE (hoping that it will work accurately enough for their particular problems) are constantly asking for more physically accurate simulation tools.

In this paper, we present initial results verifying the accuracy of the Stewart-Trinkle time-stepping method [8], which lies at the core of the dVC physics engine [2]. This method is derived directly from the instantaneous dynamic model written as dynamic complementarity problem [9] using Euler approximations of derivatives (described below) and including constraint stabilization terms. The resulting simulation method requires the solution of one complementarity problem per time step. 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 [10, 11]. Additional results are presented showing the effects of various sources

[*] This work was supported by the National Science Foundation under grants CCF-0729161 and RCV-0413227. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

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 [12].

All the results presented here were obtained through numerical studies of a challenging and important problem studied analytically and experimentally by Vose *et al.* [13, 14, 15]. 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. It's 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 contribution of this paper 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.

2 Dynamic Model

The system Vose *et al.* studied [13, 14, 15] consists of two pieces (Figure 1); a kinematically controlled plate and a dynamical part interacting with the plate.

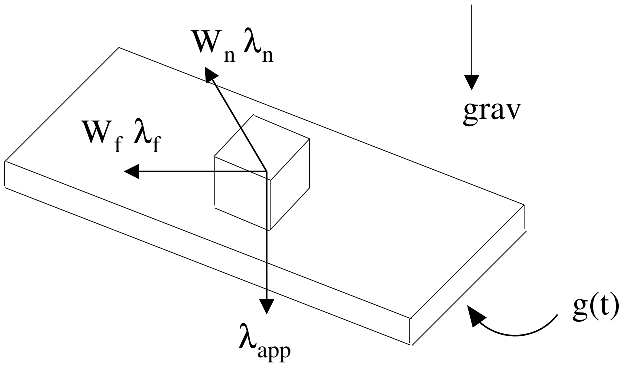


Figure 1: The plate is kinematically controlled by the vector function $g(t)$. There are three forces acting on the part: the force due to gravity $\boldsymbol{\lambda}_{\text{app}}$, the non-penetration constraint force $\mathbf{W}_n\boldsymbol{\lambda}_n$, and the frictional force $\mathbf{W}_f\boldsymbol{\lambda}_f$.

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.

2.1 Instantaneous Dynamics

One natural way to model the dynamics of such a system with frictional intermittent contact is as a differential complementarity problem (DCP) [16, 17].

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*

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

For this particular system of a part interacting with a kinematically controlled plate (figure 1), the DCP can be written as [1, 9, 18]:

$$\mathbf{M}(\mathbf{q})\dot{\boldsymbol{\nu}} = \mathbf{W}_{\text{n}}(\mathbf{q})\boldsymbol{\lambda}_{\text{n}} + \mathbf{W}_{\text{t}}(\mathbf{q})\boldsymbol{\lambda}_{\text{t}} + \mathbf{W}_{\text{o}}(\mathbf{q})\boldsymbol{\lambda}_{\text{o}} + \boldsymbol{\lambda}_{\text{app}}(\mathbf{q}, t) + \boldsymbol{\lambda}_{\text{vp}}(\mathbf{q}, \ \boldsymbol{\nu}) \tag{1}$$

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\boldsymbol{\nu} \tag{2}$$

$$0 \leq \boldsymbol{\lambda}_{\text{n}} \perp \boldsymbol{\psi}_{\text{n}}(\mathbf{q}, t) \geq 0 \tag{3}$$

$$0 = (\mathbf{U}\boldsymbol{\lambda}_{\text{n}}) \circ (\mathbf{v}_{\text{t}}) + \boldsymbol{\lambda}_{\text{t}} \circ \boldsymbol{\sigma} \tag{4}$$

$$0 = (\mathbf{U}\boldsymbol{\lambda}_{\text{n}}) \circ (\mathbf{v}_{\text{o}}) + \boldsymbol{\lambda}_{\text{o}} \circ \boldsymbol{\sigma} \tag{5}$$

$$0 \leq \boldsymbol{\sigma} \perp (\mathbf{U}\boldsymbol{\lambda}_{\text{n}}) \circ (\mathbf{U}\boldsymbol{\lambda}_{\text{n}}) - \boldsymbol{\lambda}_{\text{t}} \circ \boldsymbol{\lambda}_{\text{t}} - \boldsymbol{\lambda}_{\text{o}} \circ \boldsymbol{\lambda}_{\text{o}} \geq 0 \tag{6}$$

Equation (1) is the Newton-Euler equations of the system. The matrix $\mathbf{M}(\mathbf{q}) = \begin{bmatrix} m\mathbf{I}_{3\times3} & 0 \\ 0 & \mathcal{I}(\mathbf{q}) \end{bmatrix}$ 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:

$$\mathbf{W}_{in}(\mathbf{q}) = \begin{bmatrix} \hat{\mathbf{n}}_i(\mathbf{q}) \\ \mathbf{r}_i(\mathbf{q}) \times \hat{\mathbf{n}}_i(\mathbf{q}) \end{bmatrix} \qquad \mathbf{W}_{it}(\mathbf{q}) = \begin{bmatrix} \hat{\mathbf{t}}_i(\mathbf{q}) \\ \mathbf{r}_i(\mathbf{q}) \times \hat{\mathbf{t}}_i(\mathbf{q}) \end{bmatrix} \qquad \mathbf{W}_{io}(\mathbf{q}) = \begin{bmatrix} \hat{\mathbf{o}}_i(\mathbf{q}) \\ \mathbf{r}_i(\mathbf{q}) \times \hat{\mathbf{o}}_i(\mathbf{q}) \end{bmatrix} \tag{7}$$

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 (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 (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 (4)–(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.* $\mathbf{a}\circ\mathbf{b} = [a_1b_1 \; a_2b_2 \ldots a_nb_n]^T$). The value of σ_i also has a physical interpretation, it is the sliding speed at contact i.

The orthogonal sliding velocity components v_{it} and v_{io} for this system with a kinematically controlled plate can be written as:

$$v_{it} = \mathbf{W}_{it}^T(\mathbf{q})\boldsymbol{\nu} - \mathbf{W}_{it}^T(g)\mathbf{G}^T(g)\left(\frac{\partial g}{\partial t}\right)^T \qquad v_{io} = \mathbf{W}_{io}^T(\mathbf{q})\boldsymbol{\nu} - \mathbf{W}_{it}^T(g)\mathbf{G}^T(g)\left(\frac{\partial g}{\partial t}\right)^T \tag{8}$$

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

2.2 Discrete Time Dynamics

The DCP (equations (1)–(6)) is not solved directly, but instead a time-stepping scheme is employed and the resulting (possible nonlinear) mixed complementarity problem (MCP) is solved at each time step.

Definition 2 (Mixed Complementarity Problem) *Let $g(\mathbf{u},\mathbf{v}) : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^{n_1}$ and $f(\mathbf{u},\mathbf{v}) : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^{n_2}$ 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*

$$\begin{aligned} 0 &= g(\mathbf{u},\mathbf{v}), \qquad \mathbf{u}, \textit{free} \\ 0 \le \mathbf{v} \;&\perp\; f(\mathbf{u},\mathbf{v}) \ge 0 \end{aligned}$$

For completeness, we now present two velocity-level discrete time formulations of equations (1)-(6). Let t_ℓ denote the current time and h be the time step. 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 $\dot{\mathbf{q}} \approx (\mathbf{q}^{\ell+1} - \mathbf{q}^{\ell})/h$, we get the following nonlinear and linear discrete time systems.

2.2.1 Nonlinear Complementarity Problem Formulation

$$\begin{aligned} \mathbf{M}\boldsymbol{\nu}^{\ell+1} &= \mathbf{M}\boldsymbol{\nu}^{\ell} + h(\mathbf{W}_{\mathrm{n}}\boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1} + \mathbf{W}_{\mathrm{t}}\boldsymbol{\lambda}_{\mathrm{t}}^{\ell+1} + \mathbf{W}_{\mathrm{o}}\boldsymbol{\lambda}_{\mathrm{o}}^{\ell+1} + \boldsymbol{\lambda}_{\mathrm{app}} + \boldsymbol{\lambda}_{\mathrm{vp}}) \\ \mathbf{q}^{\ell+1} &= \mathbf{q}^{\ell} + h\mathbf{G}^{\ell}\boldsymbol{\nu}^{\ell+1} \\ 0 &= (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1})\circ(\mathbf{v}_{\mathrm{t}}^{\ell+1}) + \boldsymbol{\lambda}_{\mathrm{t}}^{\ell+1}\circ\boldsymbol{\sigma}^{\ell+1} \\ 0 &= (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1})\circ(\mathbf{v}_{\mathrm{o}}^{\ell+1}) + \boldsymbol{\lambda}_{\mathrm{o}}^{\ell+1}\circ\boldsymbol{\sigma}^{\ell+1} \\ 0 \le \boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1} &\perp \psi_{\mathrm{n}}(\mathbf{q}^{\ell+1}) \ge 0 \\ 0 \le \boldsymbol{\sigma}^{\ell+1} &\perp (\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1})\circ(\mathbf{U}\boldsymbol{\lambda}_{\mathrm{n}}^{\ell+1}) - \boldsymbol{\lambda}_{\mathrm{t}}^{\ell+1}\circ\boldsymbol{\lambda}_{\mathrm{t}}^{\ell+1} - \boldsymbol{\lambda}_{\mathrm{o}}^{\ell+1}\circ\boldsymbol{\lambda}_{\mathrm{o}}^{\ell+1} \ge 0 \end{aligned} \tag{9}$$

where we approximate the distance function at time t as $\psi_{\text{in}}(\mathbf{q}^{\ell+1}) \approx \psi_{in}(\mathbf{q}^{\ell}, t_{\ell+1}) + h\mathbf{W}_{\text{in}}^T\boldsymbol{\nu}^{\ell+1}$. The notation $\psi_{\text{in}}(\mathbf{q}^{\ell}, t_{\ell+1})$ 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 $\boldsymbol{\lambda}_{\text{vp}}$ at $\ell+1$, we have a fully implicit formulation [1] 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 [8] with quadratic friction law. In simulations with a particle, the rotation matrices in the wrenches no longer exist and neither does $\boldsymbol{\lambda}_{\text{vp}}$, resulting in a fully implicit formulation for "free."

2.2.2 Linear Complementarity Problem Formulation

In order to obtain a scheme based on mixed LCPs [8], a piecewise linear approximation of the quadratic friction cone with nonnegative force variables is needed (figure 2). For completeness, we present the derivation below.

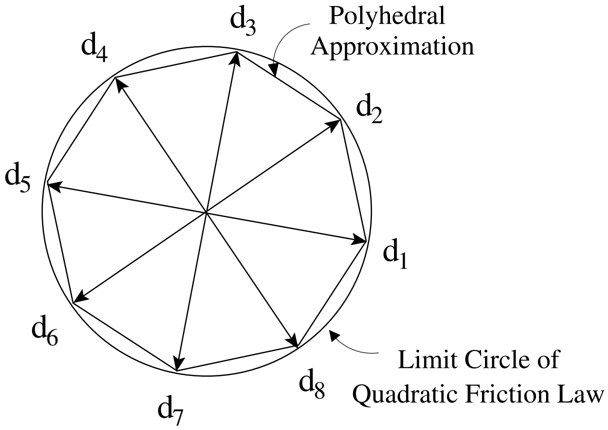


Figure 2: Friction cone approximated by an eight-sided pyramid defined by friction direction vectors d_j.

Let n_{d} friction force direction vectors $\mathbf{d}_j$ be chosen such that they positively span the space of possible friction forces, and let $(\lambda_{if})_j$ be the friction force components in those directions. Let $\mathbf{W}_{\text{f}} = \begin{bmatrix} \dots & \mathbf{W}_{if} & \dots \end{bmatrix}$ where $\mathbf{W}_{if} = \begin{bmatrix} \mathbf{d}_1 & \dots & \mathbf{d}_{n_d} \\ \mathbf{r}_i \times \mathbf{d}_1 & \dots & \mathbf{r}_i \times \mathbf{d}_{n_d} \end{bmatrix}$. Define orthogonal sliding velocity component

$$(v_{\text{if}})_j = (\mathbf{W}_{\text{if}})_j^T(\mathbf{q})\boldsymbol{\nu} - (\mathbf{W}_{\text{if}})_j^T(g)\mathbf{G}^T(g)\left(\frac{\partial g}{\partial t}\right)^T \tag{10}$$

The maximum dissipation law for contact i can now be written

$$0 \le (\lambda_{\text{if}})_j \perp (v_{\text{if}})_j + \sigma_i \ge 0, \quad j = 1 \dots n_d \tag{11}$$
$$0 \le \sigma_i \perp \mu_i\lambda_{\text{in}} - \mathbf{e}^T\boldsymbol{\lambda}_{\text{if}} \ge 0 \tag{12}$$

where $\mathbf{e} \in \mathbb{R}^{n_d}$ is a vector of ones and σ_i *approximates* the sliding speed at contact i.

The maximum dissipation for all contacts can be written compactly as:

$$0 \le \boldsymbol{\lambda}_{\text{f}} \perp \mathbf{v}_{\text{f}} + \mathbf{E}\boldsymbol{\sigma} \ge 0 \tag{13}$$
$$0 \le \boldsymbol{\sigma} \perp \mathbf{U}\boldsymbol{\lambda}_{\text{n}} - \mathbf{E}^T\boldsymbol{\lambda}_{\text{f}} \ge 0 \tag{14}$$

where $\mathbf{U}$ is a diagonal matrix with i^{th} element equal to μ_i and $\mathbf{E}$ is a block diagonal matrix with i^{th} block on the main diagonal given by $\mathbf{e}$.

Putting it all together, we arrive at the following mixed LCP formulation:

$$\begin{aligned}
\mathbf{M}\boldsymbol{\nu}^{\ell+1} &= \mathbf{M}\boldsymbol{\nu}^{\ell} + h(\mathbf{W}_{\text{n}}\boldsymbol{\lambda}_{\text{n}}^{\ell+1} + \mathbf{W}_{\text{f}}\boldsymbol{\lambda}_{\text{f}}^{\ell+1} + \boldsymbol{\lambda}_{\text{app}}^{\ell} + \boldsymbol{\lambda}_{\text{vp}}^{\ell}) \\
\mathbf{q}^{\ell+1} &= \mathbf{q}^{\ell} + h\mathbf{G}^{\ell}\boldsymbol{\nu}^{\ell+1} \\
0 \leq \boldsymbol{\lambda}_{\text{n}}^{\ell+1} &\perp \psi_{\text{n}}(\mathbf{q}^{\ell+1}) \geq 0 \\
0 \leq \boldsymbol{\lambda}_{\text{f}}^{\ell+1} &\perp \mathbf{v}_{\text{f}}^{\ell+1} + \mathbf{E}\boldsymbol{\sigma}^{\ell+1} \geq 0 \\
0 \leq \boldsymbol{\sigma}^{\ell+1} &\perp \mathbf{U}\boldsymbol{\lambda}_{\text{n}}^{\ell+1} - \mathbf{E}^T\boldsymbol{\lambda}_{\text{f}}^{\ell+1} \geq 0
\end{aligned} \tag{15}$$

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

3 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 LCP's of various sizes are given. The details on how we calculated the orientation of the kinematic plate is given in appendix A.

3.1 Simulation Verification

3.1.1 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 3). 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 [13], 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.

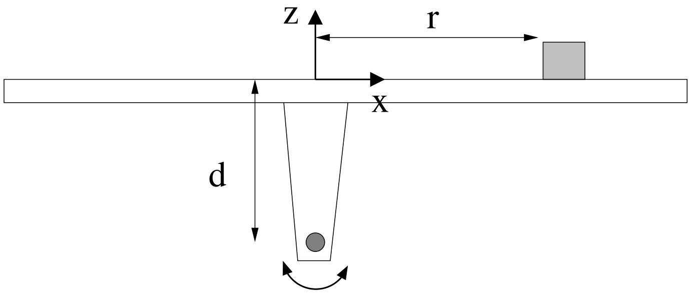


Figure 3: A part on a plate rotating about an axis below the plate. The fixed world frame is centered on the plate.

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:

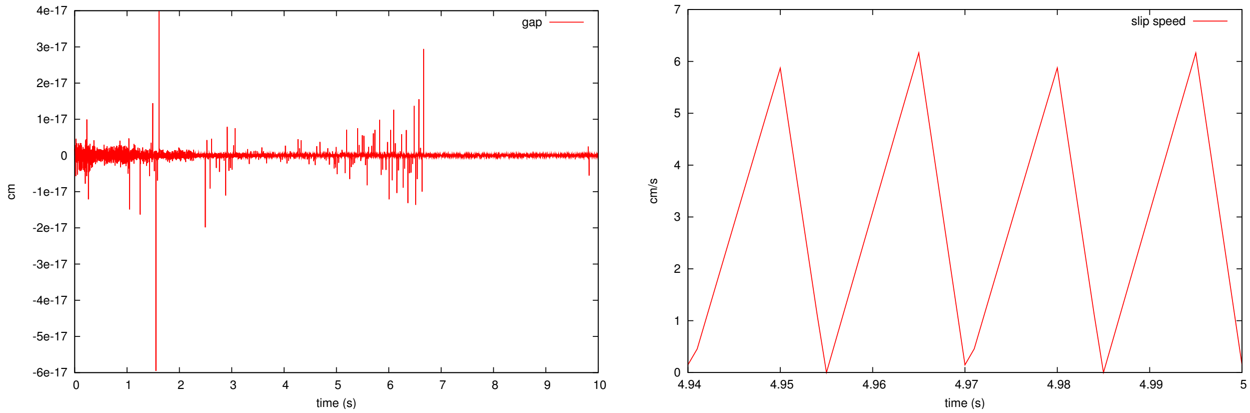
$$\theta(t) = \begin{cases} \frac{1}{2}\alpha t^2 - \frac{1}{4}\alpha T t & \text{if } 0 \leq t < T/2, \\ -\frac{1}{2}\alpha t^2 + \frac{3}{4}\alpha T t - \frac{1}{4}\alpha T^2 & \text{if } T/2 \leq t < T \end{cases} \tag{16}$$

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

$$v_{ss} = br + cr^3 \tag{17}$$

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 (17). 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.



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

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

Figure 4: The assumptions of contact maintenance and no sticking are satisfied.

Figures 4a and 4b empirically confirm that the assumptions of contact maintenance and not sticking are holding. In figure 4b, 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 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 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 6b) 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 6a).

3.1.2 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

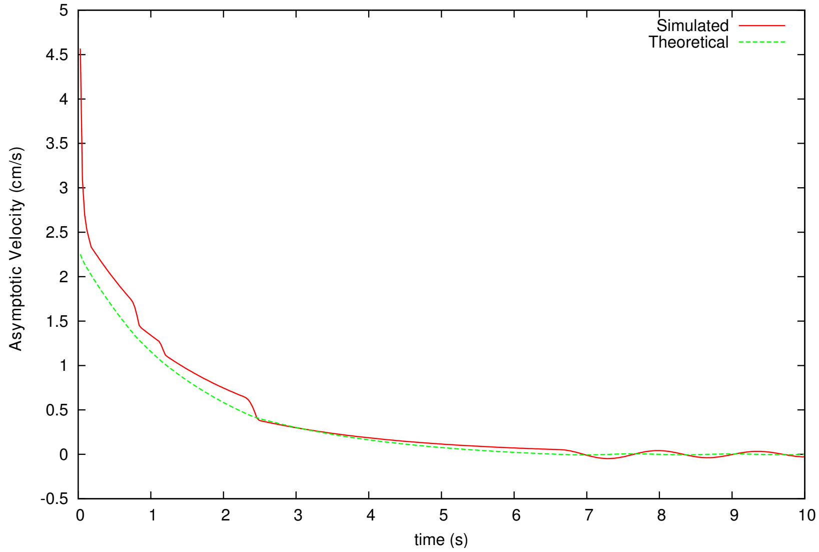
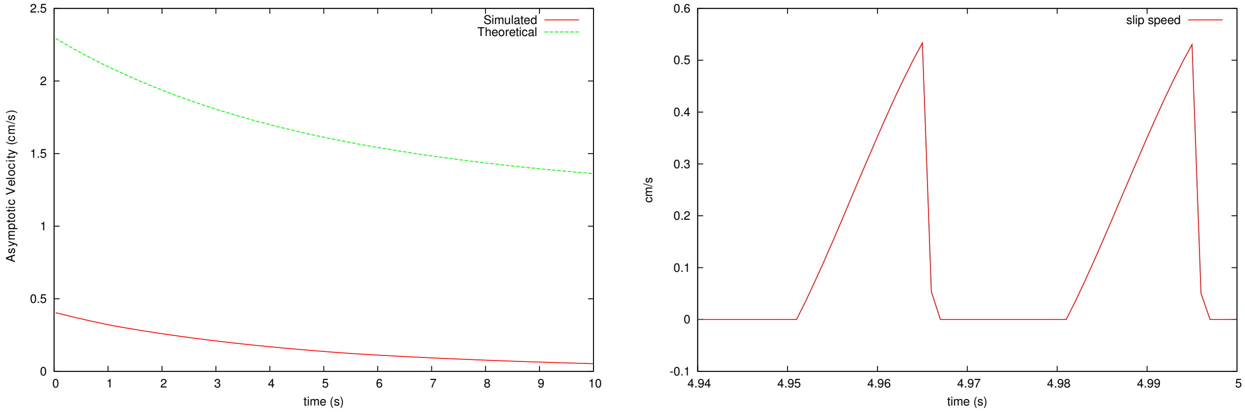


Figure 5: Comparison between the numerically computed asymptotic velocity (simulated) to the value determined in the work of Vose *et al.* (theoretical).

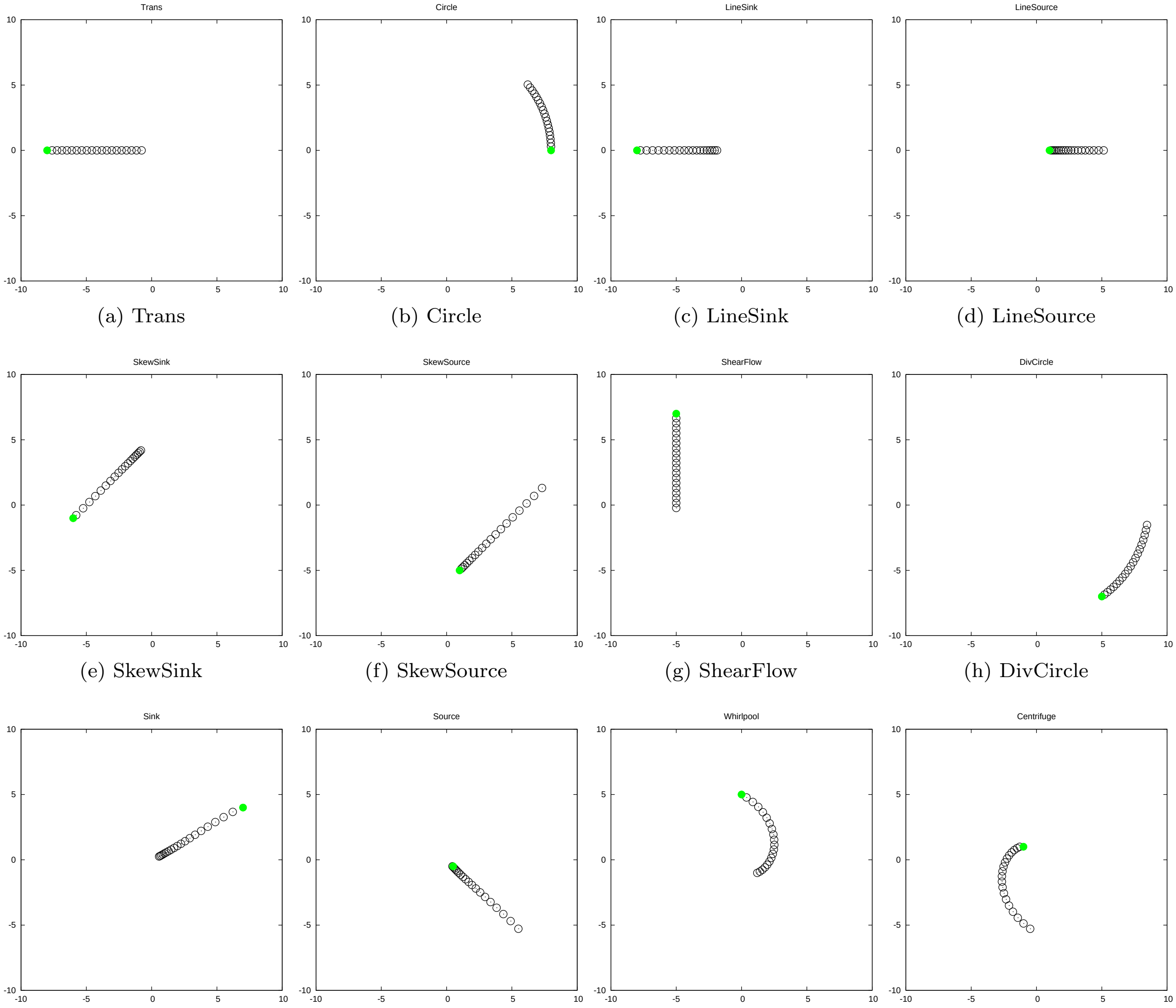


(a) Numerical and simulated steady state velocity

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

Figure 6: 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*.

the results in [14] (figure 7) 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 8 compares the position of the particle and figure 9 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.



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

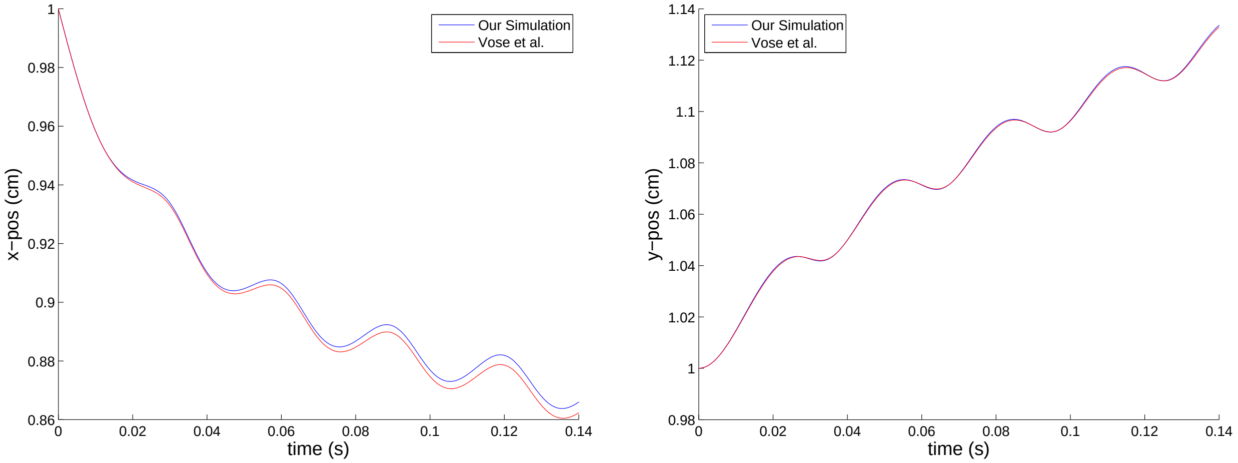
(e) SkewSink (f) SkewSource (g) ShearFlow (h) DivCircle

(i) Sink (j) Source (k) Whirlpool (l) Centrifuge

Figure 7: A six second simulation of all the fields described in figure 5 of [14] using a step size of 0.0001 seconds. The green circle indicates the starting location of the particle.

3.2 Trajectory Error as a function of step size

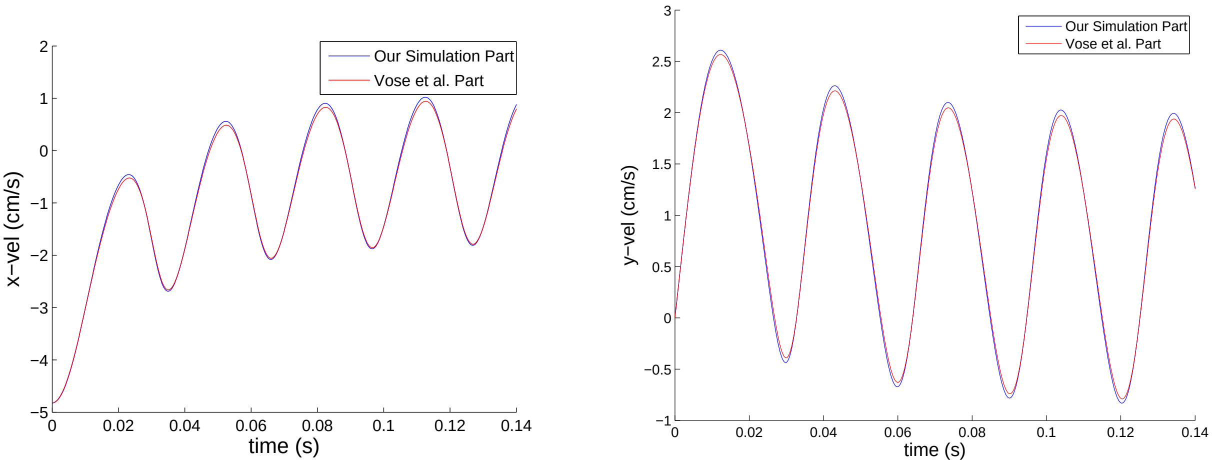
In this section, we look at how the trajectory of the part on the plate varies as a function of the step size used in the discretized mixed complementarity problem formulation. The part was modeled as a particle to remove surface friction effects.



(a) X-Position of particle

(b) Y-Position of particle

Figure 8: Direct comparison of the particle's position between our simulation results and those of Vose *et al.* for the Centrifuge motion.

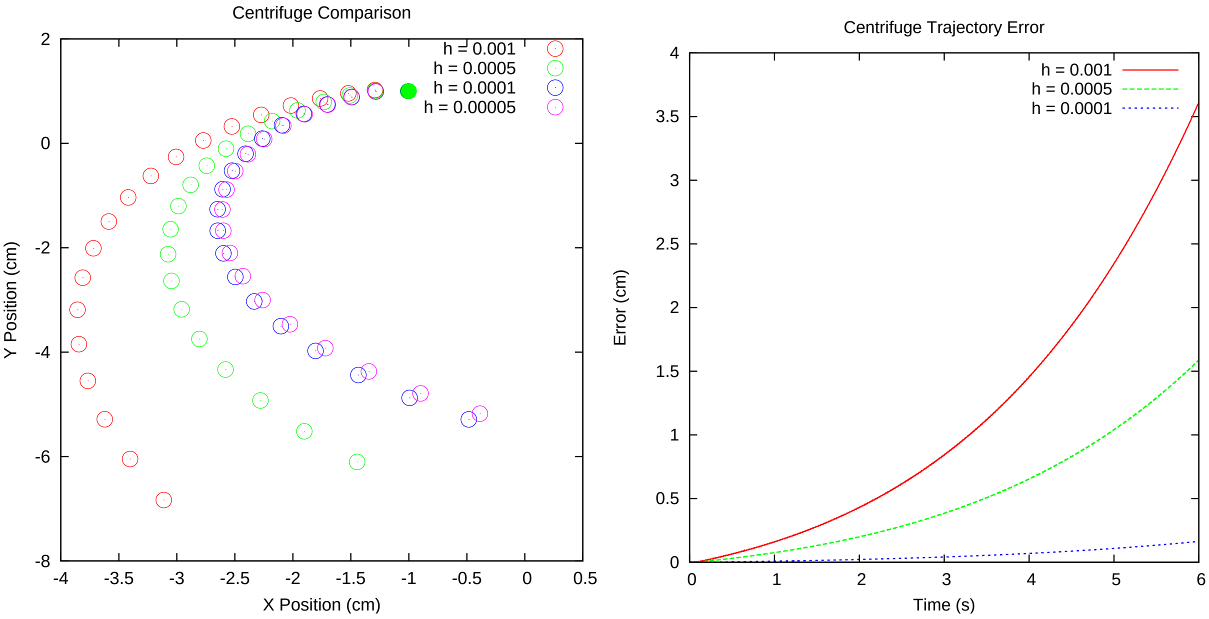


(a) X-Velocity of particle and contact point

(b) Y-Velocity of particle and contact point

Figure 9: Direct comparison of the particle's velocity between our simulation results and those of Vose *et al.* for the Centrifuge motion.

We use the NCP formulation with the quadratic friction law (equation (9)) to remove the friction linearization error and simulate the same plate controller with 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 [14] 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 10.



(a) Trajectory (ticks placed at 0.3 second intervals)

(b) Trajectory Error

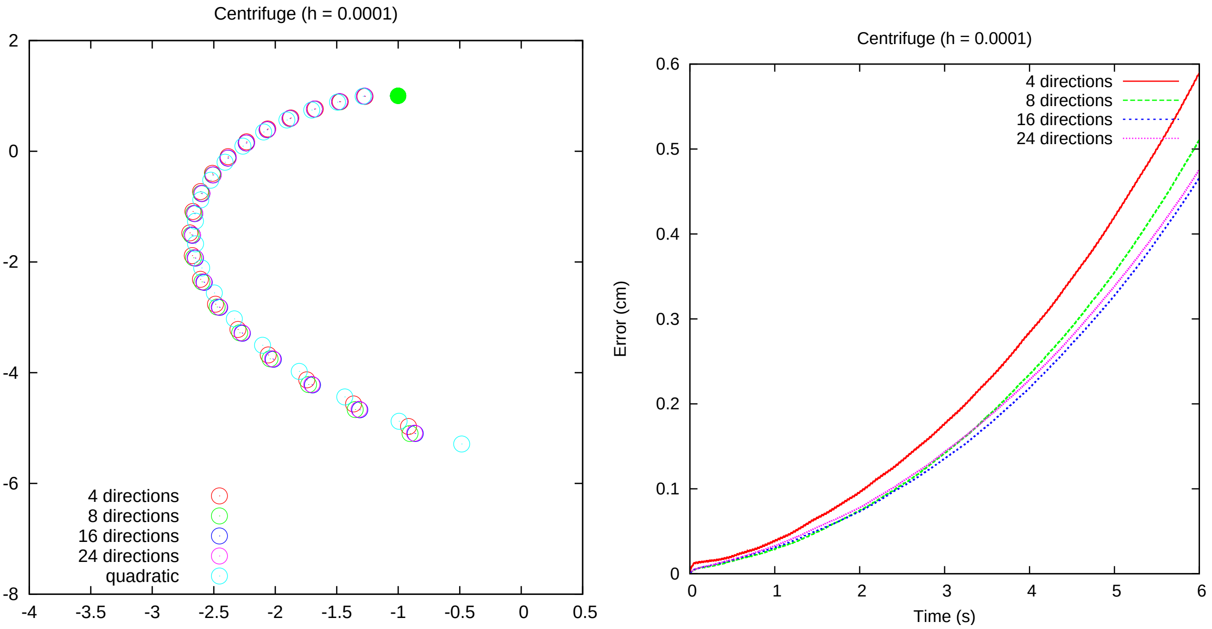
Figure 10: Trajectory and error of a particle with the plate subject to the centrifuge motion. The green circle indicates the starting location of the particle. The error (Euclidean distance) is shown as a function of step size, assuming that a step size of 0.00005 is ground truth.

From figure 10a, the choice of a $h = 0.00005$ seconds for the baseline trajectory 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.

3.3 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 trajectory error as a function of the polyhedral approximation for a 6 second simulation are shown in figure 11. Interestingly, unlike the



(a) Trajectory (ticks placed at 0.3 second intervals)

(b) Trajectory Error

Figure 11: 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.

results parameterized by step size, increasing the number of sides in the polyhedral friction cone (figure 11b) 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 [14]. For the scaled circle we scaled the angular acceleration by a factor of 5, resulting in the following plate motion function:

$$\alpha_z = 5\big(100\sin(66\pi t)\big) \tag{18}$$

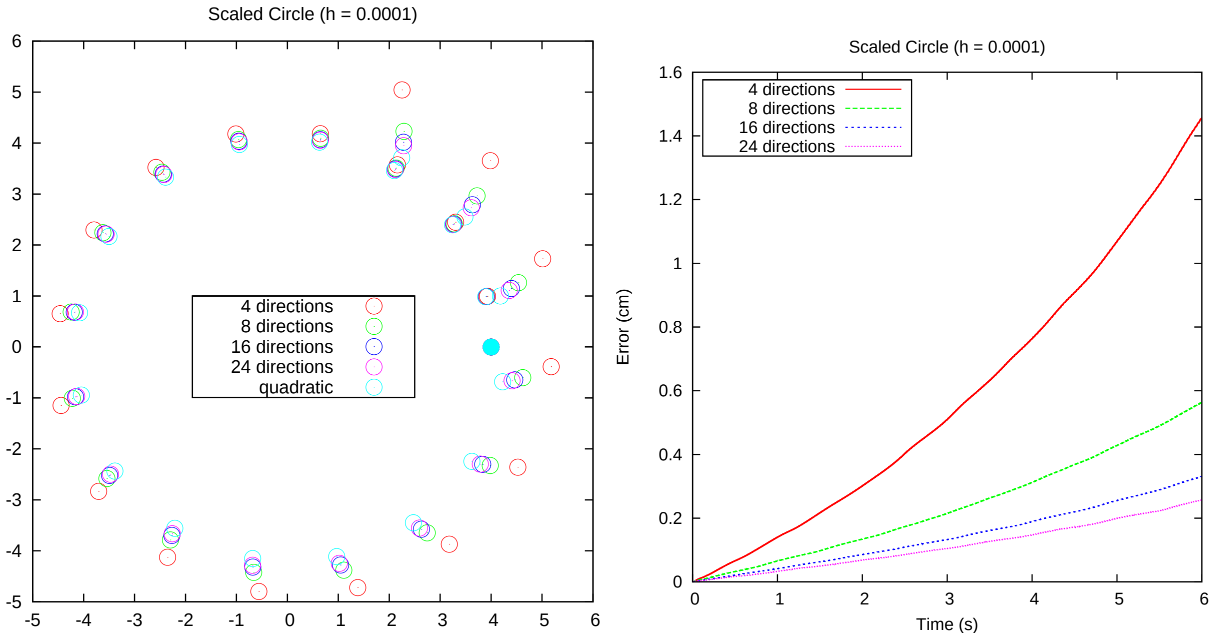
$$\ddot{p}_z = 8\sin\left(66\pi t + \frac{3}{2}\pi\right) \tag{19}$$

Unlike the centrifuge results, in the scaled circle trajectory (figure 12a), increasing the number of friction directions increasingly improves the trajectory error (figure 12b).

3.4 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 accurate, 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 13). The size of the LCP is $12+3d$, where d is the number of friction directions. For comparison, the



(a) Trajectory (ticks placed at 0.3 second intervals)

(b) Trajectory Error

Figure 12: 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.

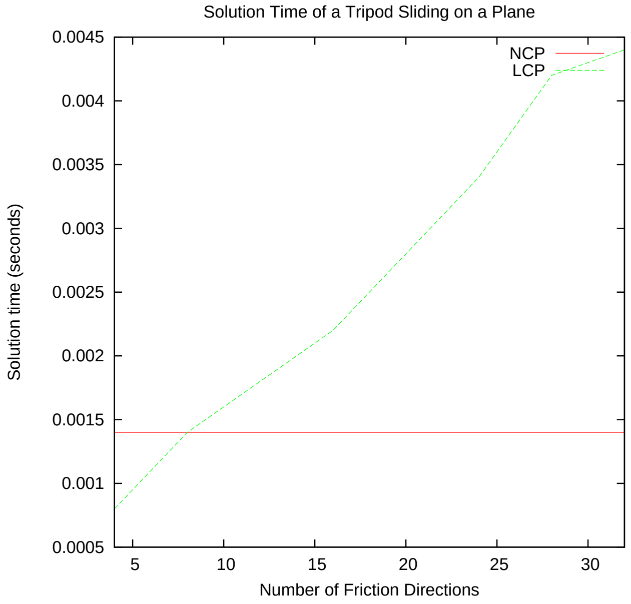


Figure 13: Timing of a single LCP function call for a block sliding on a plane as a function of number of friction directions. The solution time for an equivalent NCP formulation with quadratic friction law is also plotted. At approximately 8 friction directions, the solution time of the LCP is larger than the NCP.

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.

4 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 [19], 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 (9) or equations (15) for $\ell = 0 \dots T$) 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 [20].

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 ($\ell = 0 \dots T$) must be solved [3]. 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 non-linear programming constraints. This resulting problem is known as a mathematical program with equilibrium constraints (MPEC) [21]:

Definition 3 (Mathematical Program with Equilibrium Constraints)

$$\min_{\mathbf{u}\in\mathbb{R}^{n_1},\mathbf{v}\in\mathbb{R}^{n_2}} f(\mathbf{u},\mathbf{v}) \tag{20}$$

$$\textit{subject to: } (\mathbf{u},\mathbf{v}) \in Z, \textit{ and} \tag{21}$$

$$\mathbf{v} \textit{ solves } \textit{MCP}(g(\mathbf{u},\cdot),\mathbf{B}), \tag{22}$$

where f is a desired objective function, $Z \subseteq \mathbb{R}^{n_1+n_2}$ is a nonempty closed set (equation (21) *represents standard nonlinear programming constraints), and equation* (22) *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.

5 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 [14], 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:

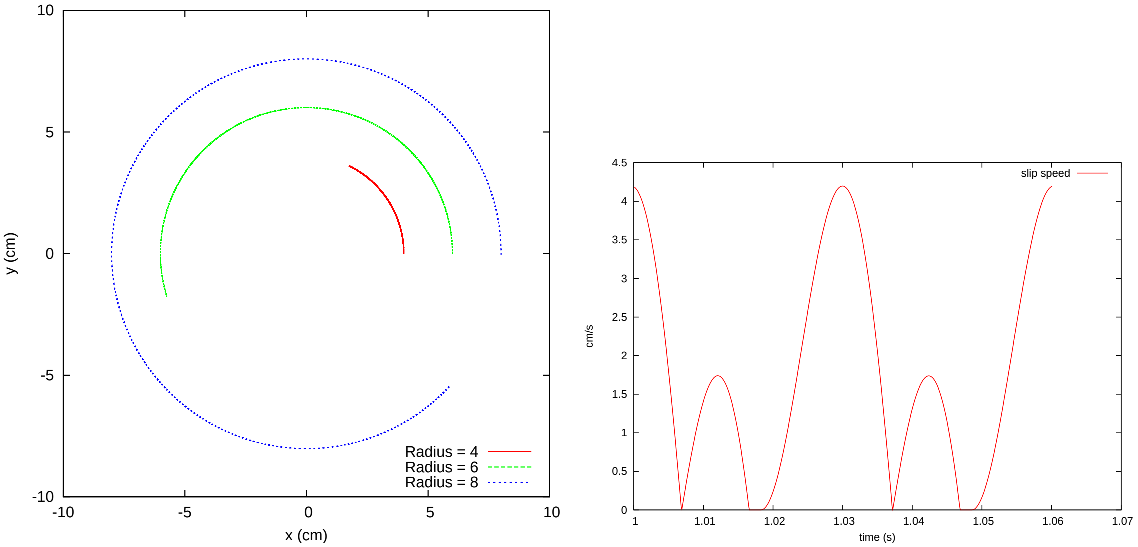
$$\ddot{\mathbf{p}} = [A_1 \sin(66\pi t + C_1) \dots A_3 \sin(66\pi t + C_3)]^T$$
$$\boldsymbol{\alpha} = [A_4 \sin(66\pi t + C_4) \dots A_6 \sin(66\pi t + C_6)]^T$$

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.

5.1 Circle Motion

In [14], 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 [14] with a particle placed are various initial radii are shown in figure 14.



(a) Trajectory

(b) Slip Speed (zoomed)

Figure 14: Results of the circle trajectory described in [14] 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 (14b) during a cycle of the plate's motion, and the part does not travel with the desired speed.

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 initialized with the values given in [14]:

$$\alpha_z = 100\sin(66\pi t) \longrightarrow A_6 = 100 \tag{23}$$

$$\ddot{p}_z = 8\sin(66\pi t + \frac{3}{2}\pi) \longrightarrow A_3 = 8, C_3 = \frac{3}{2}\pi \tag{24}$$

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:

$$\alpha_z = 62.913\sin(66\pi t) \tag{25}$$

$$\ddot{p}_z = 9.81585\sin(66\pi t + 6.33218) \tag{26}$$

Figure 15 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 maintenace and limited search space it was unable to find a plate motion that worked for smaller initial radii.

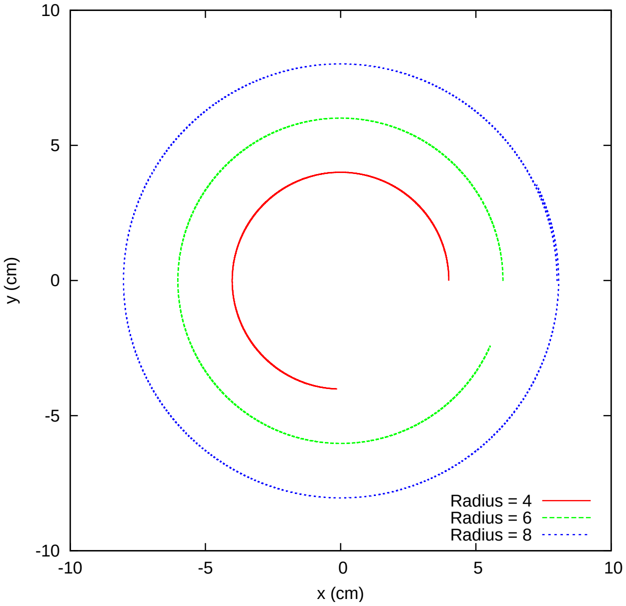


Figure 15: Results of the learned circle trajectory with a constant step size of .0001 and a penalty for particle plate separation.

Unlike the analysis done in [14], we can allow the particle to lose and regain contact with the plate (in the presented formulation (equation (9)), 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

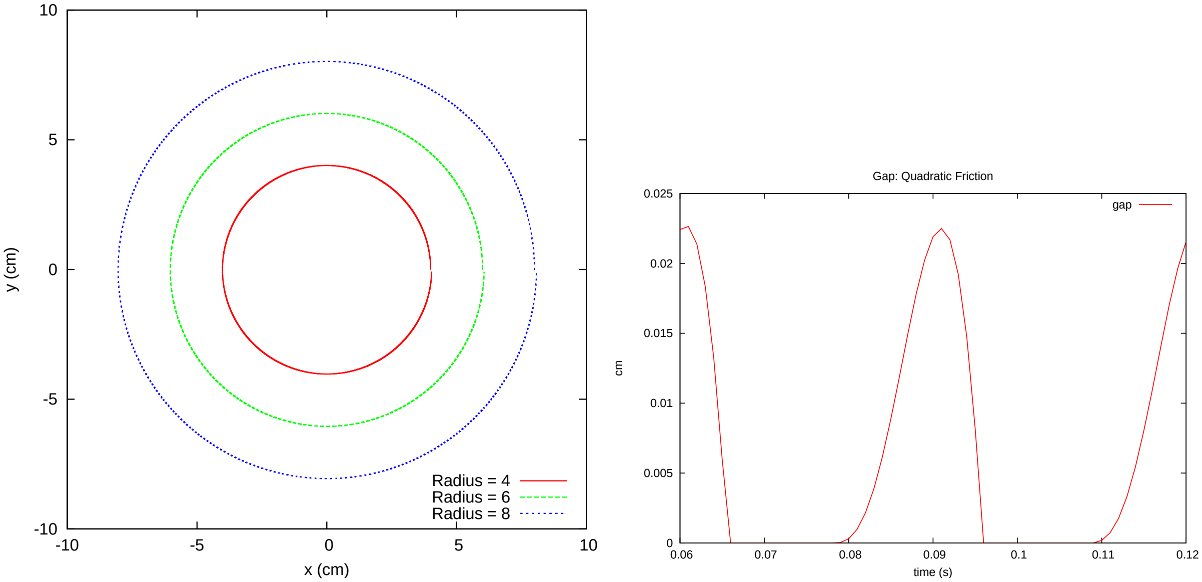
1.03644. This objective value is much less than the previous value found with the restrictions in place and is very close to satisfying the goal.

The corresponding sinusoidal motion of the plate is given by:

$$\alpha_z = 40.8332 \sin(66\pi t) \tag{27}$$

$$\ddot{p}_z = 15.3654 \sin(66\pi t + 6.5374) \tag{28}$$

The results of the simulation are shown in figure 16. With this plate motion, the particle is both sticking and breaking contact (figure 16b) 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 [14] since the part is both separating and sticking during the course of this particular plate motion.



(a) Trajectory

(b) Gap

Figure 16: 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.

5.2 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 [14]. 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:

$$\alpha_x = 3.88479\sin(66\pi t - 1.08465) \tag{29}$$
$$\alpha_y = -4.31188\sin(66\pi t - 2.0938) \tag{30}$$
$$\alpha_z = 0 \tag{31}$$
$$\ddot{p}_x = 155.816\sin(66\pi t + 0.430032) \tag{32}$$
$$\ddot{p}_y = 204.171\sin(66\pi t + 5.77261) \tag{33}$$
$$\ddot{p}_z = 0 \tag{34}$$

Figure 17 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.

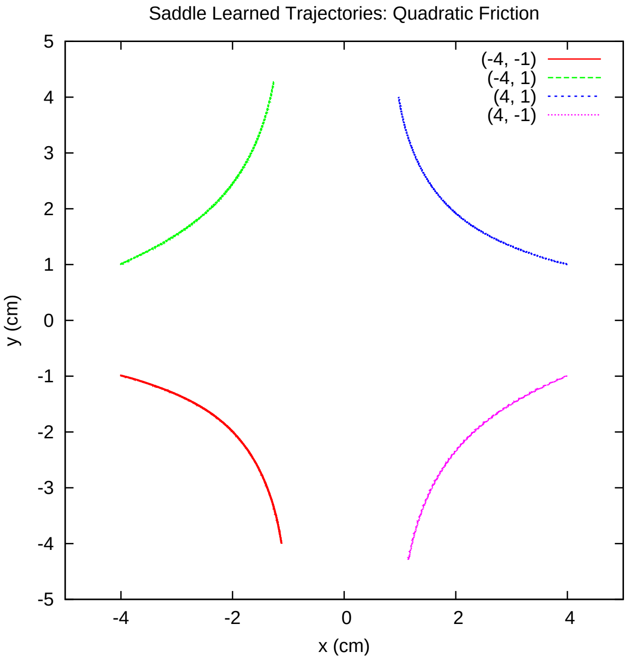


Figure 17: Results of the learned saddle trajectory for various initial positions and a step size of .0001.

6 Conclusion and Future Work

In this paper, we discussed simulation results for rigid bodies in intermittent frictional contact with a rigid plate undergoing a prescribed spatial trajectory, as studied theoretically and experimentally by Vose *et al.* [13, 14, 15]. The results were obtained with a complementarity-based time-stepping method whose subproblems were formulated as complementarity problems. 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 the final sections of the paper, we discussed how to generate new plate motions from using simulation results. 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 [14], 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.

Acknowledgements

We thank Tom Vose and Kevin Lynch for their timely feedback and insightful discussions.

References

[1] 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.

[2] 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 ICRA*, 2007.

[3] 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.

[4] Shinjiro Sueda, Andrew Kaufman, and Dinesh K. Pai. Musculotendon simulation for hand animation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 27(3), 2008.

[5] 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.

[6] 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.

[7] R. Smith. Open dynamics engine, 2008. http://www.ode.org.

[8] D.E. Stewart and J.C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *Int'l Jrnl. of Numerical Methods in Engineering*, 39:2673–2691, 1996.

[9] J. Trinkle, J. 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.

[10] 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.

[11] D.E. Stewart. Convergence of a timestepping scheme for rigidbody dynamics and resolution of painlev's problem. *Archive for Rational Mechanics and Analysis*, 145(3):215–260, 1998.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

[17] Jong-Shi Pang and Francisco Facchinei. *Finite-Dimensional Variational Inequalities and complementarity Problems (I).* Springer Verlag, New York, 2003.

[18] 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*, pages 174 – 181, March 2005.

[19] Bruce Donald, Patrick Xavier, John Canny, and John Reif. Kinodynamic motion planning. *Journal of the ACM*, 40(5):1048–1066, 1993.

[20] P. Song, J.C. Trinkle, V. Kumar, and J.-S. Pang. Design of part feeding and assembly processes with dynamics. In *icra*, apr 2004.

[21] Z.Q. Luo, J.S. Pang, and D. Ralph. *Mathematical Programs With Equilibrium Constraints.* Cambridge University Press, Cambridge, England, 1996.

APPENDIX

A Computation of Plate's Orientation

The sinusoidal plate controllers in [14] 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

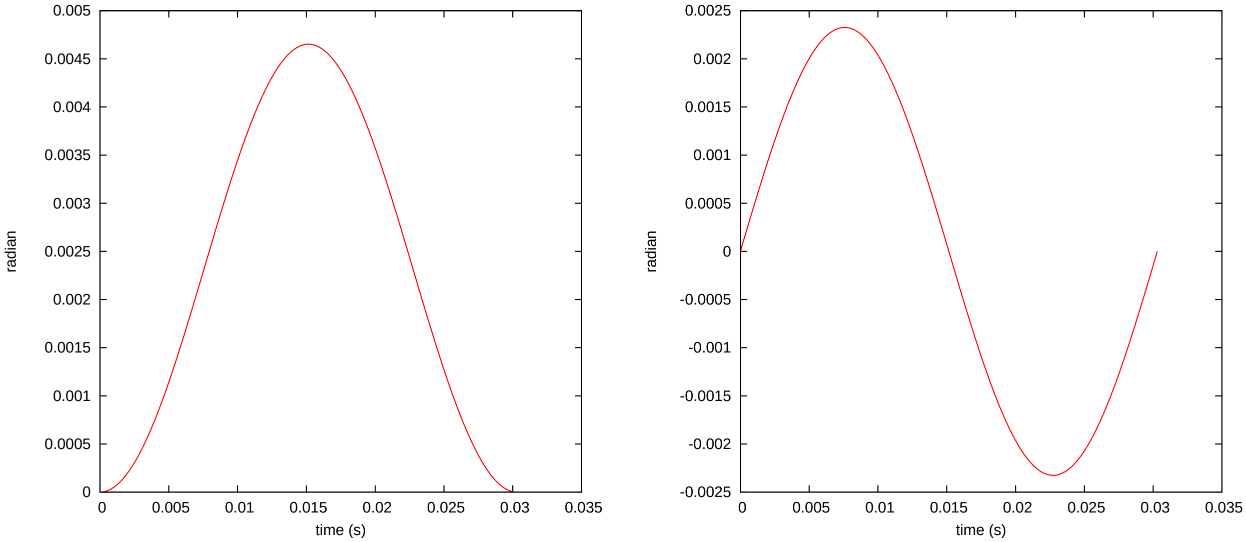
$$\boldsymbol{\varepsilon}(t) = \left[\cos\left(\frac{\theta}{2}\right)\ \sin\left(\frac{\theta}{2}\right)\mathbf{k}\right]^T \tag{35}$$

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 was used to solve the ODE system defined in equation (2). 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 [14] 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 18 shows the resulting rotation angles about the x and y axes if we solve equation (2) with $\boldsymbol{\varepsilon}(0) = [1\ 0\ 0\ 0]$, the identity unit quaternion. The rotation about the y-axis (figure 18b) is symmetric about zero, but the rotation about the x-axis (figure 18a) is not.

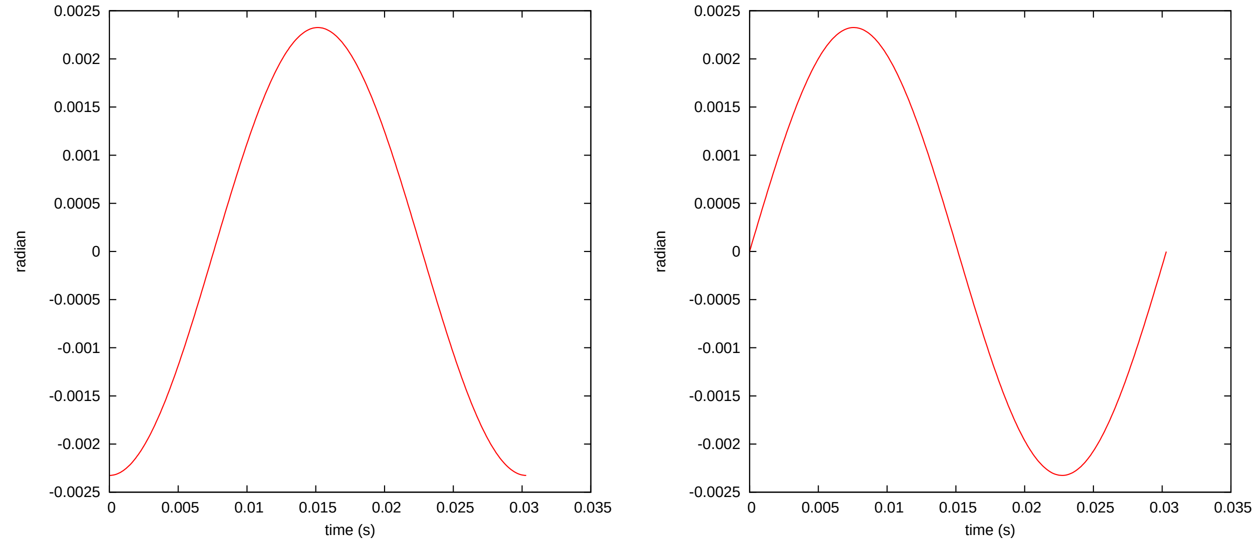


(a) Rotation About X-Axis

(b) Rotation About Y-Axis

Figure 18: 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.

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 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 $[0.999999,\ -0.00116298,\ 3.04478 \times 10^{-09},\ -4.24925 \times 10^{-06}]$. Figure 19 shows the resulting rotation angles about the x and y axes if we solve equation (2) with this computed initial unit quaternion.



(a) Rotation About X-Axis

(b) Rotation About Y-Axis

Figure 19: The x and y angles of rotation for one period of the centrifuge plate motion, when starting with the computed initial unit quaternion.