Simulations with Sliding and Intermittent Contact

Matthew P. Kelly
Cornell University
mpk72@cornell.edu

July 29, 2014

Abstract

Systems that have sliding and intermittent contact are said to have hybrid dynamics. These systems have continuous modes (sliding, rotating, flight...) that are connected with discrete transitions (collision, lift-off...). Simulation of hybrid systems can be tricky, and this report will cover a few subtitles that are encountered when simulating hybrid systems that are associated with the contact mechanics of rigid bodies.

1 Simulation Architecture

When simulating mechanical hybrid systems, there are two fundamentally different architectures for the simulator. We will call the first *event-based* simulation and the second *impulse-based* simulation. Event-based simulation is the traditional way to simulate a hybrid system, and it directly uses the finite state machine structure of the hybrid system. Impulse-based simulation is an approximation that is used for systems where the full finite state machine is too complex to explicitly write out. The simulations in this report exclusively use event-based simulation, since their finite state machines are simple and I'm interested in high-accuracy results.

1.1 Event-Based Simulation

In an event-based simulation a full finite state machine must be written out for the system dynamics. In each continuous state a set of differential equations governs the continuous dynamics of the system. The integration algorithm is typically a variable-step integrator that uses root finding to determine the precise instant in time when the state invariants or transition guards switch. This results in an accurate simulation, since both the continuous and discrete dynamics can be calculated to near machine precision.

The major down-side of this type of simulation is that for some mechanical systems, the full finite state machine is too complicated to write down. For example, suppose that you are trying to simulate a system of 10 dice being thrown. Each die can contact the ground at either one, two, or four points[1], and then every die and touch a few other dice as well, in arbitrary locations.

1.2 Impulse-Based Simulation

Impulse-based simulation was developed to deal with systems with arbitrarily complex contact mechanics, such as the dice example from the previous section. It is used in most general dynamics engines, such as Bullet and Open Dynamics Engine (ODE). Some people also call impulse-based simulation *time-stepping* simulation.

The key idea in impulse-based simulation is that each rigid body[2] is handled independently. The dynamics are written out such that all forces are expressed as impulses, given some fixed time-step. At every time step, the impulsive contact forces between objects are solved as part of a large optimization program. This optimization problems solves for the unique solution to these three constraints, applied at every contact pair that is detected[3]. These constraints form a linear complementarity problem (LCP).

$d_n > 0$	Contact seperation
$J_t \leq \mu J_n$	Contact force in friction cone
$d_n J_t = 0$	Contact force when touching

[1] four contact points causes difficulties even for a single die in isolation...

[2] Some algorithms have special algorithms for dealing with linkages, such as a "Featherstone's Algorithm" which is actually a collection of algorithms, published by Roy Featherstone in his book *Rigid Body Dynamics Algorithms* [1].

[3] There is a whole field of research dedicated to determining which points of which objects are in contact, but it is outside the scope of this report.

2 Bouncing Ball

One of the simplest hybrid systems is a bouncing ball. There is one continuous phase of motion: (flight through the air)

$$\dot{x} = v \tag{1}$$
$$\dot{v} = -g \tag{2}$$

and one discrete transition: (collision with the ground)

$$v^+ = -k \cdot v^- \tag{3}$$

Simulation of this system will show that the ball simply bounces up and down, attaining less height on each successive bounce (assuming a realistic value of $0 < k < 1$).

2.1 Zeno's Paradox

It seems that simulating such a simple system would be trivial, but it turns out that you will quickly discover a problem. The problem is that the simple model of a bouncing ball will experience an infinite number of bounces in a finite amount of time. This behavior is known as Zeno's paradox, and it will typically break the simulator, if not handled properly.

Let's look at why this happens. The analytic solution for bounce time, for the ball bouncing equations given above is:

$$T_i = 2\frac{v_i}{g} = 2\frac{(k^i v_0)}{g} \tag{4}$$

Since this is a geometric series, it can readily be shown that the time required for an infinite number of bounces is given by Equation 5. Assuming that the coefficient of restitution (k) is strictly less than one, an infinite number of bounces will occur in a finite time. As a computer attempts to run a simulation up to and through this critical time, it will necessarily run into an error (such as the integration time step going to machine precision, or the event detection missing the collision, or just missing a bounce entirely).

$$\sum_{i=0}^{\infty} T_i = \frac{2v_0}{g}\sum_{i=0}^{\infty} k^i = \left(\frac{2v_0}{g}\right)\left(\frac{k}{1-k}\right) \tag{5}$$

This particular issue of bouncing contact is common in simulations, and most simulation engines treat it as a special case. One common way to do this is to prescribe a minimum escape velocity; if the separation velocity is smaller than this threshold, then the simulator will assume the the objects stay in contact.

2.2 When event detection fails

A bouncing ball on flat ground is not very exciting, but things get more interesting when it is bouncing around on hilly terrain. Since the dynamics are still fairly simple, this is an excellent place to use a variable-step integration algorithm with event detection (as opposed to an impulse-based simulator).

Event detection works by evaluating an event vector at every grid point, and checking the sign of the result. If there is a sign change, then an event occurs and the simulator calls a root finding routine. Interestingly, if the dynamics are simple, but the event function is complicated, this scheme will miss events. This is because the variable-step integrator (Matlab's ode45 in the case of Figure 2.2) is trying to maximize step size while meeting an accuracy constraint. Since the dynamics are simple, it takes huge time steps, which can miss rapid changes in the event function.

In the case of the ball bouncing over hilly terrain, the height of the ball with respect to the ground (the event function) is changing much more rapidly than the absolute height of the ball (the state dynamics). This allows the ball to tunnel through a steep hill, as shown in Figure 2.2. One way to solve this problem is to put a limit on the maximum step size of the integrator, such that it is unlikely to miss a collision.

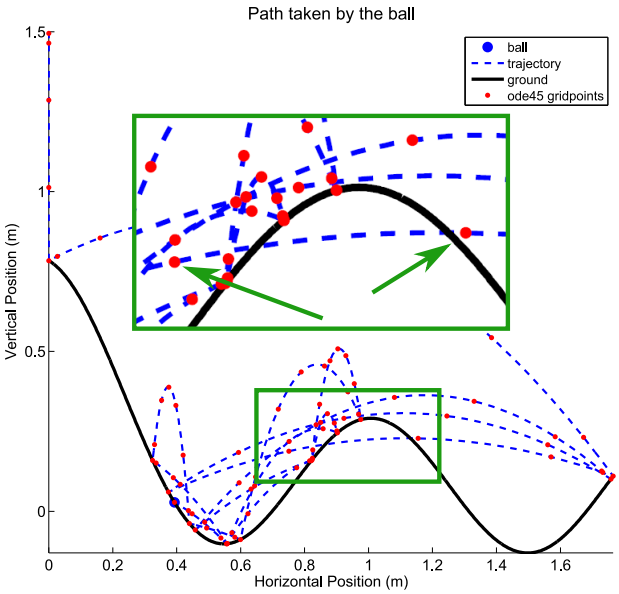

Figure 1: A bouncing ball over hilly terrain. This figure illustrates a special case where matlab's built in event detection fails due to a large step-size in ode45.

3 Sliding is not optional

In many simple simulations, it is tempting to assume a 'no-slip' boundary condition. Under this (bad) assumption, the contact point can either be pinned or free, and the transitions between modes are based on the normal force and separation distance. A simulation using the no-slip boundary condition will either use negative contact forces or allow the colliding object to penetrate, depending on the type of error. The solution to this problem is to assume some sort of contact model that allows for sliding.

3.1 The Toppling Pencil

In the 1980's, Tad McGeer published a paper [4] which showed that a simulation of a pencil toppling from rest produce non-physical results if you make this 'no-slip' assumption. His simulations showed the pencil slowly falling over, and then at some critical angle, the pencil would begin to accelerate through the floor. Usually one would assume that this was a bug in the code, but in this case the code was correct, and the problem was due to the no-slip assumption.

As the pencil topples from rest, there is some critical angle where the normal force goes to zero. In general, the tangential force on the contact is non-zero at this point, and in reality the pencil would slide in response to this force. Since McGeer neglected sliding, his simulator set the contact mode to free. Instead of the contact point remaining above the ground, it instead began to accelerate through the ground. Why?

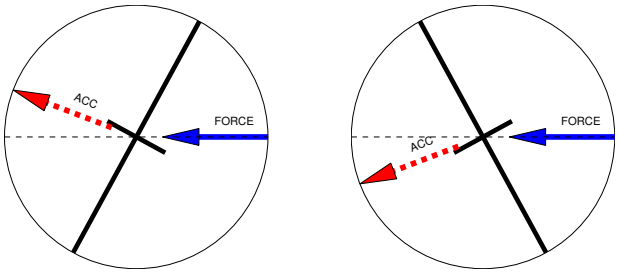

Figure 2: Acceleration of the contact point of a thin rod. The solid blue arrow shows an applied force, and the red dashed arrow shows the resulting acceleration.

Let's consider the instantaneous acceleration of the contact point when the pin-constraint is removed. Before the constraint is removed, the contact point is stationary, but has a non-zero external force applied to it. This implies that there are some inertial forces that are attempting to accelerate this point, and that these forces precisely balance the external force. When the constraint is removed, so are these external forces, and thus the contact point accelerates due to these inertial forces.

One way to understand which direction the contact point accelerates is by understanding the mass matrix of the contact point. This matrix describes how a point on a rigid body moves in response to a force applied at that point. In general, the direction of acceleration is not aligned with the direction of the force. Figure 2 shows a visualization of the mass matrix, where the length of the dark line shows the effective mass in that direction. The left side of the figure shows the case observed in the toppling pencil example: The change in contact forces is directed to the right, and the pencil is tipped to the right. The resulting inertial effects act to swing the contact point into the ground.

3.2 No-Slip $\neq$ Infinite Friction

One reason that people give for making the 'no-slip' assumption is that it is "the limiting case as the coefficient of friction goes to infinity". This is completely false, as is demonstrated by the following example.

Consider a simple example of sliding: a stick being dragged with constant velocity as shown in Figure

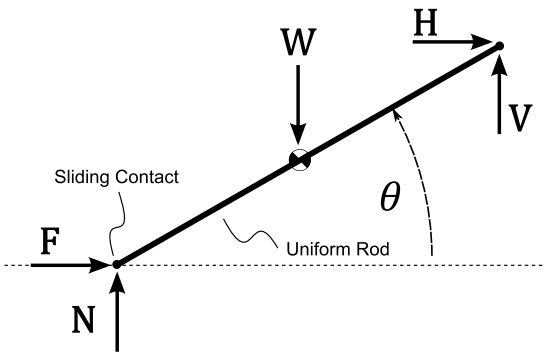

Figure 3: A stick that is being dragged along the ground with constant velocity.

3. Assume that the contact forces are governed by Coulomb friction with coefficient μ. Since the velocity of the stick is constant, we can treat this as a statics problem. It can be shown that horizontal force (H) required to maintain constant speed is given by Equation 6

$$H(\theta) = \frac{W\mu\cos(\theta)}{2\cos(\theta) + \mu\sin(\theta)} \tag{6}$$

Taking the limit as $\mu \to \infty$ shows that the horizontal force remains finite. It can also be shown that the normal force (N) at the contact goes to zero as $\mu \to \infty$. These trends are shown numerically in Figure 4

$$\lim_{\mu\to\infty} H(\theta) = \frac{W}{\tan(\theta)} \tag{7}$$

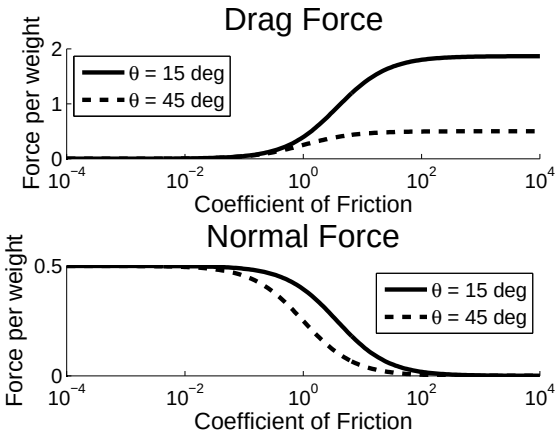

Figure 4: Contact force limits as a function of μ for a stick dragging with constant velocity.

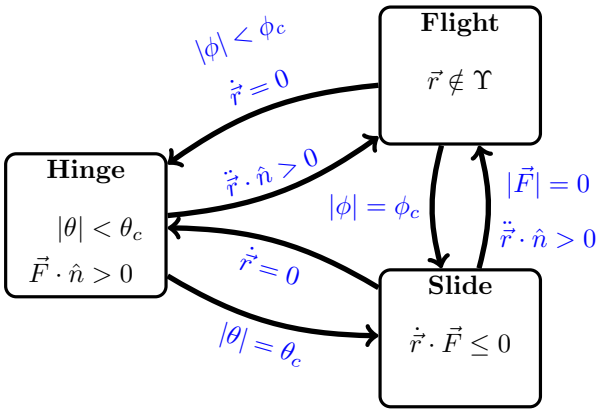

Figure 5: Finite State Machine for generalized rigid body contact

3.3 Contact Finite State Machine

The 'correct' way to deal with contacts is to allow for three different contact modes at each contact: free, sliding, and pinned. Each of these contact modes can be expressed as a constraint at that contact point:

- **free -** The normal force is zero
- **sliding -** The tangential force is such that the contact point travels along the surface
- **pinned -** The tangential force cannot do positive work on the system.

In addition to these constraints, there are also a set of transition conditions that link each of these continuous modes. Figure 5 shows an example of a finite state machine constructed from these states and transitions. Notice that each state has different equations of motion. This is fine for a single contact, but things get very complicated if you have multiple contacts. Consider robot model with two contact points - now the finite state machine has nine states instead of three.

4 Toppling Stick

This first part of this section is an overview of how to create a finite-state-machine-based simulator for a two-dimensional rigid stick. The remainder of the section discusses some simulation results that investigate the conditions necessary for slipping as the pen-

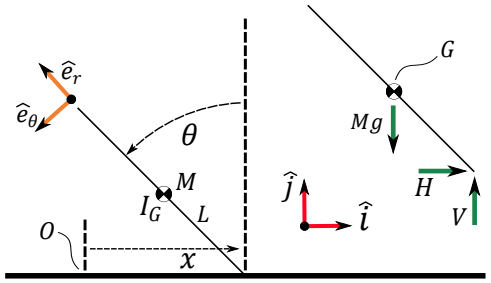

Figure 6: Free body diagram for both phases of motion.

cil topples from rest.

4.1 Peinlevé Paradox

Peinlevé's Paradox occurs when a rigid body is sliding along a surface while tipped backwards. Under these conditions it is possible for a jamming motion to occur where the contact forces cause the body to rotate upwards, which in turn increases the contact force. Under these special conditions the contact forces and accelerations to become infinite in a finite amount of time. This phenomena is well studied and is thoroughly discussed in references: [5] [2] [3]. It is this effect that allows dashed lines to be easily and rapidly drawn on a chalk board. I mention it here as an interesting side note, as remainder of the section are not related to this paradox.

4.2 Model for Continuous Dynamics

This section discusses the general model that is shared across all contact modes. The hinge and slide modes will be discussed in the following sections. The stick is modeled as a rigid body with:

M = mass
I_G = moment of inertia about center of mass (CoM)
L = distance between contact point and CoM

Forces acting on the stick:

Mg = body force due to gravity
H = horizontal contact force
V = vertical contact force

The stick has two degrees of freedom (neglecting flight phase):

θ = angle of the stick
x = horizontal position of contact point

The equations of motion are derived using two frames of reference - an inertial frame and a body frame:

$\hat{\imath}$ = positive horizontal direction
$\hat{\jmath}$ = positive vertical direction
$\hat{k} \equiv \hat{\imath} \times \hat{\jmath}$
$\hat{e}_r$ = direction along stick, pointing away from contact
$\hat{e}_\theta$ = direction transverse to $\hat{e}_r$

Define rotating reference frame and derivatives:

$$\begin{cases} \hat{e}_r &= -\sin(\theta)\hat{\imath} + \cos(\theta)\hat{\jmath} \\ \hat{e}_\theta &= -\cos(\theta)\hat{\imath} - \sin(\theta)\hat{\jmath} \end{cases}$$
$$\begin{cases} \dot{\hat{e}}_r &= \dot{\theta}\hat{e}_\theta \\ \dot{\hat{e}}_\theta &= -\dot{\theta}\hat{e}_r \end{cases}$$
$$\begin{cases} \ddot{\hat{e}}_r &= \ddot{\theta}\hat{e}_\theta - \dot{\theta}^2\hat{e}_r \\ \ddot{\hat{e}}_\theta &= -\ddot{\theta}\hat{e}_r - \dot{\theta}^2\hat{e}_\theta \end{cases}$$

4.3 Hinge Phase Derivation

In the Hinge phase of motion the contact point is fixed at the origin. The system is constrained to rotate about this point, thus giving the system one degree of freedom. The following constraints are used to determine when this phase is no longer valid. They correspond to transitions to falling, flight, and sliding respectively.

$-\pi/2 < \theta < \pi/2$ stick above ground
$V \geq 0$ positive normal contact force
$-\mu V \leq H \leq \mu V$ coulomb friction

Define position vector from the contact point to the CoM:

$$\vec{r} = L\hat{e}_r$$
$$\dot{\vec{r}} = L\dot{\hat{e}}_r$$
$$\ddot{\vec{r}} = L\ddot{\hat{e}}_r$$

Conservation of angular momentum (rate) about the contact point:

$$\sum M_{/o} = \dot{\vec{H}}_{/o}$$
$$\vec{r} \times (-Mg\hat{j}) = I_G\ddot{\theta}\hat{k} + \vec{r} \times (M\ddot{\vec{r}})$$

Conservation of linear momentum (rate):

$$\sum F = \dot{\vec{L}}$$
$$H\hat{i} + V\hat{j} - Mg\hat{j} = M\ddot{\vec{r}}$$

I used Matlab's symbolic toolbox to directly solve these equations for the angular acceleration and contact forces. These symbolic expressions, along with expressions for the kinematics and energy, were then automatically written to functions to be used by the simulation.

4.4 Slide Phase Derivation

In the Slide phase of motion, the contact points is constrained to move along horizontal ($\hat{i}$) axis and the contact forces obey coulomb friction. The following constraints are monitored during simulation. A transition is triggered when they are no longer satisfied.

$-\pi/2 < \theta < \pi/2$	stick above the ground
$V \geq 0$	positive normal contact force
$\lvert\dot{x}\rvert \leq 0$	direction cannot change
$\lvert H\rvert = \mu V$	Coulomb friction

Define position vector from the origin to the CoM:

$$\vec{r} = x\hat{i} + L\hat{e}_r$$
$$\dot{\vec{r}} = \dot{x}\hat{i} + L\dot{\hat{e}}_r$$
$$\ddot{\vec{r}} = \ddot{x}\hat{i} + L\ddot{\hat{e}}_r$$

Conservation of angular momentum (rate) about the origin:

$$\sum M_{/o} = \dot{\vec{H}}_{/o}$$
$$\vec{r} \times (-Mg\hat{j}) + (x\hat{i}) \times (V\hat{j}) = I_G\ddot{\theta}\hat{k} + \vec{r} \times (M\ddot{\vec{r}})$$

Conservation of linear momentum (rate):

$$\sum F = \dot{\vec{L}}$$
$$H\hat{i} + V\hat{j} - Mg\hat{j} = M\ddot{\vec{r}}$$

I used Matlab's symbolic toolbox to directly solve these equations for the angular acceleration and contact forces. These symbolic expressions, along with expressions for the kinematics and energy, were then automatically written to functions to be used by the simulation.

4.5 Simulation

I wrote a simulation in matlab that was based on a finite state machine (FSM) architecture. Each continuous phase of motion was integrated using ODE45 in Matlab, with built-in event detection to check that the constraints were satisfied. At the start of the simulation, it determines which phase of motion the system is in, and then integrates the corresponding equations of motion until a constraint is violated. At this point, the system runs the FSM to determine which phase of motion to transition to. Flight phase was included for completeness, but is never used in these experiments.

4.6 Experiment - Pencil Toppling from Rest

The goal of the experiment is to better understand what happens when the contact forces on an object go to zero. In this case, every experiment starts with a stick toppling from rest. The coefficient of friction at the contact and moment of inertia of the stick a both adjusted over a wide range of values. For every trial the slip distance and critical angle before slipping are recorded.

The parameters in the problem were scaled to be dimensionless: $\{M = 1, g = 1, L = 1\}$. Additionally, the moment of inertia (I_G) is scaled such that $I_G = 0$ corresponds to a point mass at the CoM and $I_G = 1$ corresponds to half the mass at each end of a stick with length $= 2L$. As a point of reference, $I_G = 1/3$ corresponds to a slender rod.

In order for the stick to fall over in simulation, it must be given an initial perturbation. To keep results consistent, the initial energy of the system is held constant across all trials. This prescribes a fixed

relationship between the initial angle and rate:

$$E(\theta,\omega) \equiv MgL\cos(\theta) + \frac{1}{2}(ML^2 + I_G)\omega^2 \quad (8)$$

$$E(\theta_0,\omega_0) \equiv E(0,0) = MgL \quad (9)$$

Solving for initial angular rate yields:

$$\omega_0(\theta_0) = \sqrt{\frac{2MgL(1-\cos(\theta_0))}{ML^2 + I_G}} \quad (10)$$

It turns out that the results of the experiment are sensitive to these initial conditions. For example, if you assume that the system has a small initial angle but no initial velocity, then you will get slightly different results.

4.7 Toppling Point Mass with Infinite Friction

With infinite friction at the contact, the stick will start to slip when the vertical component of the contact forces goes to zero. Assuming the initial conditions from Equation 10, the vertical reaction force is given by:

$$\begin{aligned} V(\theta) =& \frac{Mg(I_G - 2mL^2\cos(\theta) + 3ML^2\cos^2(\theta))}{I_G + ML^2}\bigg|_{I_G\to 0} \\ =& Mg\,cos(\theta)\,(3\cos(\theta) - 2) \end{aligned}$$

Solving for the non-trivial root of this equation yields the critical value for θ at which slipping occurs:

$$\theta_c = \arccos(2/3) \approx 48.189685^\circ \quad (11)$$

A little bit of simple algebra yields a nice expression for the angular velocity at this critical angle:

$$\omega_c = \sqrt{\frac{2g}{3L}} \quad (12)$$

While slipping with infinite friction the vertical component of the contact force is zero by definition. The horizontal component is given by:

$$H = 2I_G\left(\frac{g - L\omega^2\cos(\theta)}{L^2\sin(2\theta)}\right)\bigg|_{I_G\to 0} = 0 \quad (13)$$

Since the only non-zero force acting on the stick (in this special case) is its own weight, the center of mass (CoM) will follow a parabolic trajectory. The initial position and velocity of the CoM are given by:

$$\vec{r}_c = \langle -L\sin(\theta_c), L\cos(\theta_c)\rangle \quad (14)$$

$$\dot{\vec{r}}_c = \langle -L\omega_c\cos(\theta_c), -L\omega_c\sin(\theta_c)\rangle \quad (15)$$

The final distance (d^*) reached by the CoM at the time of impact (t^*) can be found by solving the following system:

$$0 = \Big(L\cos(\theta_c)\Big) + \Big(-L\omega_c\sin(\theta_c)\Big)t^* + \frac{1}{2}\Big(-g\Big)(t^*)^2 \quad (16)$$

$$d^* = \Big(-L\sin(\theta_c)\Big) + \Big(-L\omega_c\cos(\theta_c)\Big)t^* \quad (17)$$

Since the CoM is falling/sliding in the negative direction, we need to add L to get the distance that the contact point slipped:

$$d_{slip} = d^* + L \quad (18)$$

After doing quite a bit of algebra and a letting $g = 1$ and $L = 1$ it can be shown that the slip distance is:

$$d_{slip} = \left(1 - \frac{4}{27}\sqrt{23} - \frac{5}{27}\sqrt{5}\right) \approx -0.124580 \quad (19)$$

4.8 Results - Pencil Toppling from Rest - General Case

For intermediate values of μ there is a regime in which the stick slips in *both* directions. In these cases, the stick first slides backwards, then rotates about a fixed point, and then slides forwards until ultimately striking the ground.

Sticks with low I_G slide for *all* values of μ. This includes the limit as the coefficient of friction goes to infinity.

Sticks with large I_G have a critical value of μ for which they do not slide.

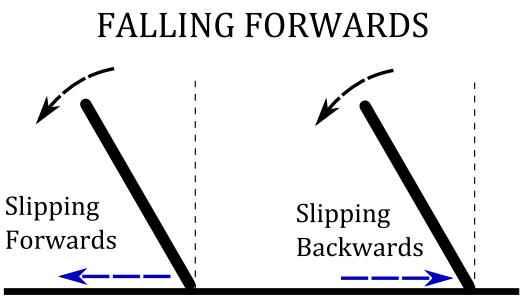

Figure 7: Shows sign conventions for slipping forwards and backwards

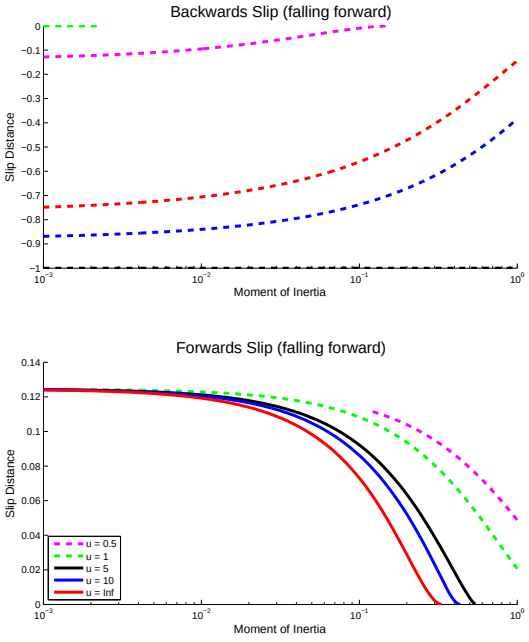

Figure 8: Positive and negative slip as a function of coefficient of friction and moment of inertia. Notice that the stick slides backwards with low friction and forwards for high friction. The slip distance is smaller for large sticks with a large moment of inertia.

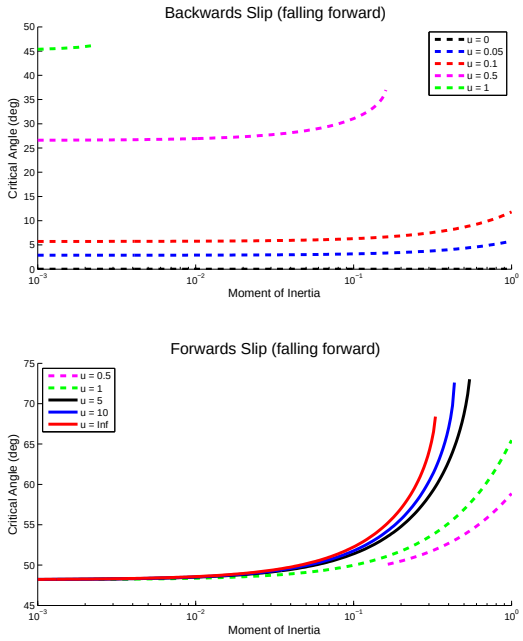

Figure 9: Critical angle before slipping as a function of coefficient of friction and moment of inertia. Notice that as the moment of inertia becomes small (like a point mass) the critical slip angle asymptotically approaches that of a ball bearing rolling off of a sphere.

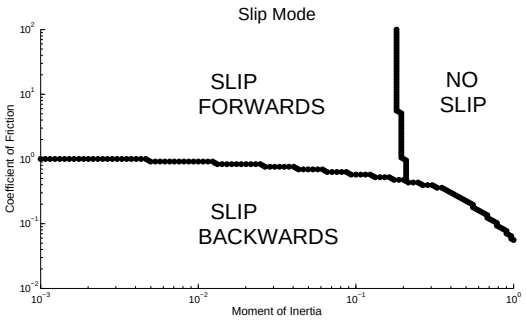

Figure 10: Shows which direction the stick will slip first. Notice that there is a small region where no sliding occurs, but that most of the region is dominated by either forwards or backwards sliding. Notice that for all reasonable values of friction ($\mu < 1$) the stick slips backwards first.

References

[1] Roy Featherstone. *Rigid Body Dynamics Algorithms*. 2007.

[2] S. S. Grigoryan. The solution to the Painleve paradox for dry friction. *Doklady Physics*, 46(7):499–503, July 2001.

[3] R.I. Leine, B. Brogliato, and H. Nijmeijer. Periodic motion and bifurcations induced by the Painlevé paradox. *European Journal of Mechanics - A/Solids*, 21(5):869–896, January 2002.

[4] Tad McGeer. Wobbling, toppling, and forces of contact. *American Journal of Physics*, 57(12):1089, 1989.

[5] Yizhar Or and Elon Rimon. Investigation of Painlev e s Paradox and Dynamic Jamming During Mechanism Sliding Motion. pages 1–20.