PRECISION MOTION CONTROL AT LOW VELOCITIES IN THE PRESENCE OF FRICTION

A Senior Thesis

By

Erika Ooten

1997-98 University Undergraduate Research Fellow

Texas A&M University

Group: Engineering III

PRECISION MOTION CONTROL AT LOW VELOCITIES IN THE PRESENCE OF FRICTION

A Senior Thesis

By

Erika Ooten

1997-98 University Undergraduate Research Fellow

Texas A&M University

Group: ENGINEERING III

PRECISION MOTION CONTROL AT LOW VELOCITIES.IN THE PRESCENCE OF FRICTION

By

Erika A. Ooten

Submitted to the
Office of Honors Programs and Academic Scholarships
Texas A&M
University in partial fulfillment of the requirements for

1997-98 UNIVERSITY UNDERGRADUATE RESEARCH FELLOWS PROGRAM

April 16, 1998

Approved as to style and content by:

R. Langari

Reza Langari (Faculty Advisor)
Department of Mechanical Engineering

Susanna Finnell

Susanna Finnell, Executive Director
Honors Programs and Academic Scholarships

Fellows Group: Engineering III

Abstract

Precision Motion Control at Low Velocities in the Presence of Friction

Erika A. Ooten, University Undergraduate Fellow, 1997-1998, Texas A&M University, Department of Mechancial Engineering

Reza Langari: advisor

The force of friction is almost always present when there is relative motion between two objects. There is currently no model of friction that describes this force at all times. In particular slip stick friction particular is extremely important since it determines the lower bounds for any piece of machinery that must operate at velocities close to zero. At a velocity near zero, most models of friction characterize the friction force as discontinuous. I propose that the friction force is not discontinuous at zero. I have tried to generate a friction model that is continuous around a zero velocity. In order to generate the friction model, I used two computer programs: Working Model and Matlab . The software I used is able to represent simple models of friction, but when the model becomes more complex, the computer modeling programs are often at their limits. Nonetheless, it is shown that a model that captures the microscopic structural features of the contact zone exhibits the experimentally observed phenomena with relative fidelity. In particular it is shown that low velocity stick-slip oscillations are potentially enhanced by regularity of the asperities on the contacting surfaces while random appearances of these asperities reduces or potentially eliminates these oscillations. This in in my view an important result that can have applications in the design of machinery and in particular in precision motion applications.

Precision Motion Control at Low Velocities in the Presence of Friction

Leanardo Da Vinci defined friction to be independent of the contact area, and its direction always opposes motion [4]. However, today, there are many different models of friction which are used. The force of friction is present in almost every system where there is relative motion between two objects. In some situations, the force of friction can be ignored when compared to the other forces that drive the system. In other situations, the force of friction is the driving force in the system. For these cases, it is important to use a friction model that adequately describes the experimentally collected data. One of these cases where the friction force is one of the most important factors in the system is in the control of motion at or near zero velocity. In order to precisely control the motion of an object at low velocities, the friction model must reflect what has been determined experimentally to be the friction force[1].

There are many different types of friction forces that are present during motion. The type of friction generally depends on the relative velocity of the object.

At the most basic level, friction is characterized by two different kinds of friction forces: static and kinetic friction. *Static* friction is defined as the force necessary to initiate motion from rest [4]. *Kinetic* friction is defined friction force that is present in the system once motion has begun. Equation 1 describes both the kinetic and static friction forces.

$$F_{Friction} = \mu_{*} N \quad (1)$$

The coefficient of friction is characterized by μ_{*} and it can be equal to one of two constants. One constant is used to define static friction, the other constant is used to define kinetic friction. The magnitude of the coefficient of friction is dependent on the type of materials that are described. For example, a smooth surfaced material should have a lower coefficient of friction than that of a rougher material. This is experimentally verified; more force is necessary to initiate motion of the rough material than is needed to initiate motion on the smoother material. In most cases, the value of the static friction coefficient is larger than the value of the kinetic friction coefficient for the same material. Equation 1 is adequate for estimating the friction force in simple systems, and in situations where the motion is steady and the magnitude of the velocity is not near zero.

A different type of friction force is characterized by a viscous friction force. In viscous friction, a component of the total friction force is proportional to the velocity. Viscous friction is used to describe the friction force when motion just begins. As the velocity of an object goes to zero, the viscous friction force component also goes to zero. It is important to note that the viscous effect on the friction force is related to the lubrication. An increase in the viscosity of a lubricant between two moving plates will increase the viscous component of friction.

An important time period for any friction model is during the time interval when no motion is present, but an outside force is applied to an object. The friction force characterized during the period of no motion is called Dahl friction [4]. Dahl friction is a phenomenon that arises due to the elastic deformation of the bonding sites between two materials. Any material surface is composed of microscopic peaks, or asperities, and valleys (See figure 1).

Figure 1

Before motion can take place, the peaks on both or one of the surfaces must deform in order to allow the other surface to pass over it. The elastic deformation of a material's surface before motion is known as the Dahl effect. It is possible to model the elastic deformation of a material before motion has occurred as a linear spring. When no external force is applied to either of the two objects at rest, the Dahl friction is equal to zero. When a load is applied, the Dahl friction force begins to increase linearly. It in creases until the applied force deforms the material enough such that motion can occur (See figure 1). The amount of force that is applied at the exact time when motion occurs is knows as the *breakaway force* (1)

When two surfaces are in contact, the actual contact areas vary according to the surface roughness of a material. If the two surfaces are under no external pressures, then the bottom surface has to support the weight of the top surface. The weight of the surface is distributed linearly across the bottom of the top surface, however, the contact areas vary. If the contact area is small enough, the pressure ,which is calculated as force per unit area can be great enough to cause the two surfaces to "weld" together at the contact point. When this occurs, plastic rather than elastic deformation must happen before motion can begin. The welded joint must break apart before motion can take place.

When motion occurs at or near zero velocity, the motion of an object can be charaterized as stick-slip. Stick-slip occurs when a system traveling at speeds near zero periodically come to rest. As the mass slows down, the addition of the total friction force and the unbalanced force causing motion equal zero, and the object "sticks" in place. When the sum of the two forces no longer equals zero, the object begins to slip against the other surface. It is important to know when slip-stick will occur since it will determine the lower performance bounds of a machine [1]. In order for a machine to maintain a constant speed at low velocities, the friction model used in the controller must account for slip-stick motion.

In the past, the force of friction has been characterized mathematically as a discontinuous function The discontinuous model of friction incorporates the static and kinetic models of friction (See figure 2).

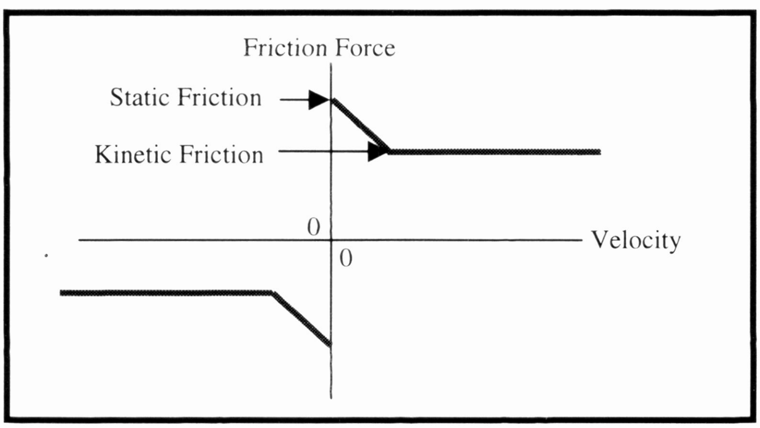


Figure 2

This model can also include the viscous effect of friction (See figure 3).

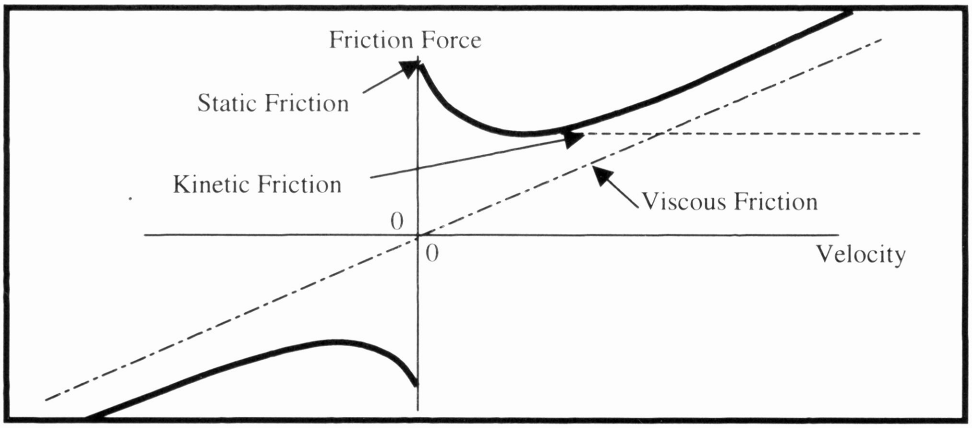


Figure 3

The discontinuity takes place at zero, however, it is proposed that the force of friction is not discontinuous, but actually continuous. This can be accomplished with the introduction of the Dahl Effect (See figure 4).

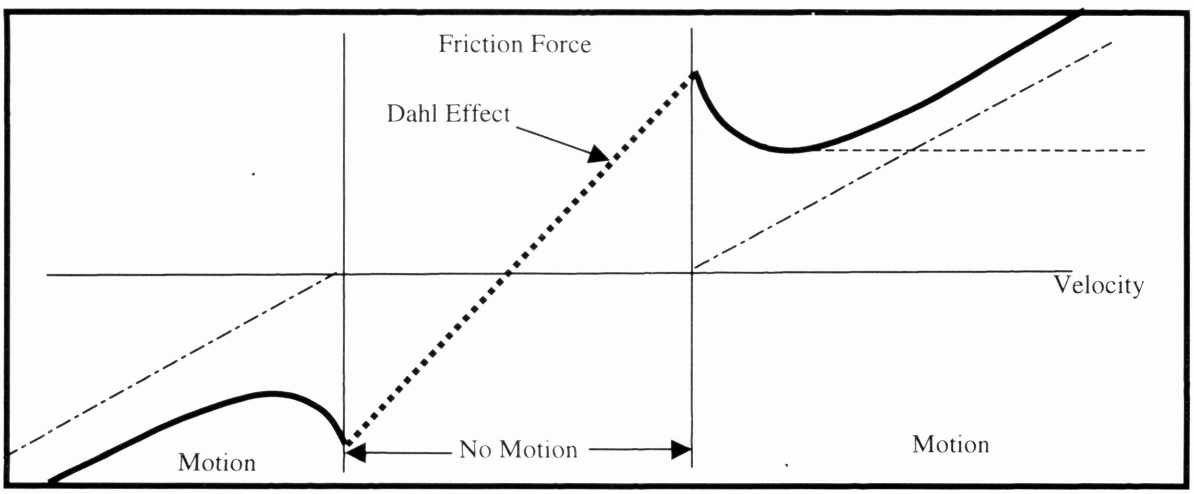


Figure 4

The purpose of the research was to create a model of friction that undergoes the stick-slip phenomena at speeds near zero. The friction model should incorporate the following different types of friction: static friction, kinetic friction, viscous friction and the Dahl friction. If model predictions match previously collected experimental data, the friction model can then be used in a system's controller. In the controller, the friction model would be used in order to estimate what the actual friction force is. The controller is then able to predict what the friction force will be and make changes based on the model. In theory, this will allow the system reduce the impact of slip-slick oscillations in precision motion control systems.

Initially, it was decided that the computer modeling program Working Model would be used in order to generate a real time simulation of what occurs on the microscopic level (Working Model – Knowledge Revolution, Inc., Pala Alto, California). The surface of the two materials would be modeled as a series of mass peaks or asperities. The asperities were modeled as triangular shaped bodies with a finite mass. In order to account for the Dahl effect, the mass peaks were connected to the stationary bottom surface with springs. The springs represent the elastic deformation at the material surface (See figure 5).

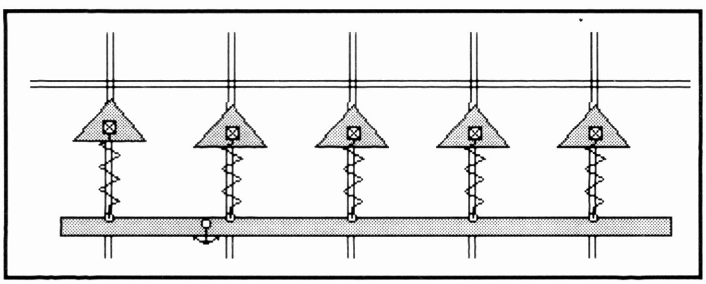


Figure 5

The mass peaks on the top surface were rigidly connected to an upper moving surface (See figure 6).

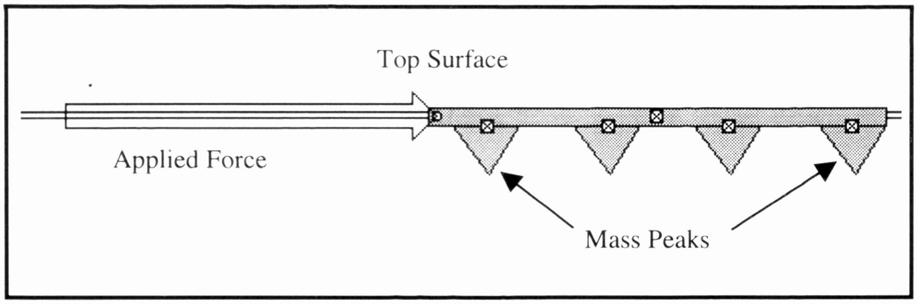


Figure 6

The bottom surface would remain motionless while the top surface moves horizontally relative to the bottom surface. The motion of the mass peaks on the bottom surface was constrained such that the peaks could move in the vertical direction, but not in the horizontal direction. This was accomplished by placing the mass spring system on a vertical track (See figure7).

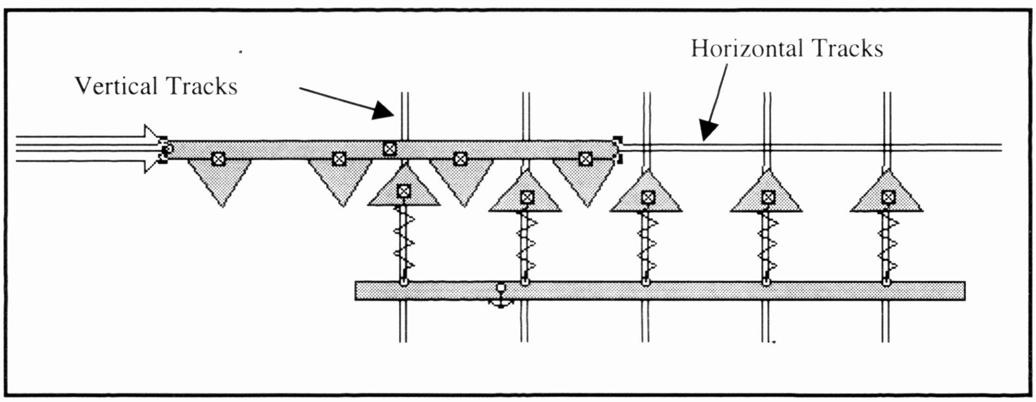


Figure 7

Similarly, the top surface was constrained to move on a horizontal track. After running some simulations, dampers were added to to reduce transient phenomenon due to impact, and to increase the accuracy of the materials elastic deformation (See figure 8).

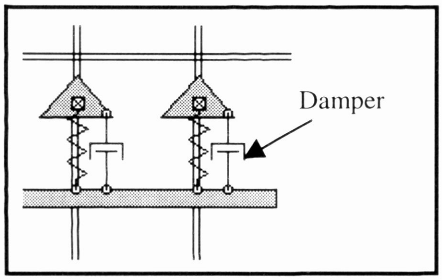


Figure 8

The paramaters that were modified during the simulations include: the heights of the triangular mass peaks, the distance between the asperities, the spring stiffnesses, the force applied to the top surface, and the damping coefficients. The triangle parameter was the first parameter that was changed. The triangles were either full triangles or half triangles. Holding everything else constant, the half triangles required a larger force to pass over a given mass peak on the bottom surface (See figure 9).

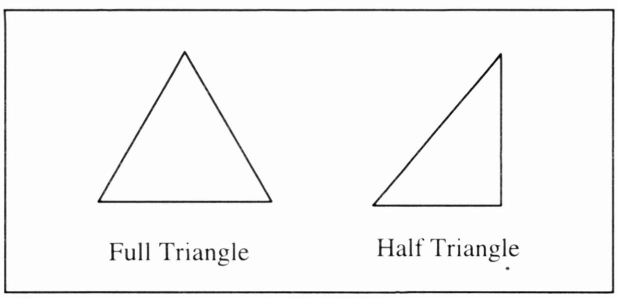


Figure 9

In the half triangles, once the tip of the top triangles passes over the bottom triangle, it is no longer in contact with the bottom triangle. In the full triangles, however, the triangles remain in contanct until the back edge of the top triangle moves past the right edge of the bottom triangle. Consequently, the restoring force in the spring pushes the top surface in the same direction as it is moving. The friction simulation used in Working Model verified that the full triangles would require less force to pass a given mass peak (See

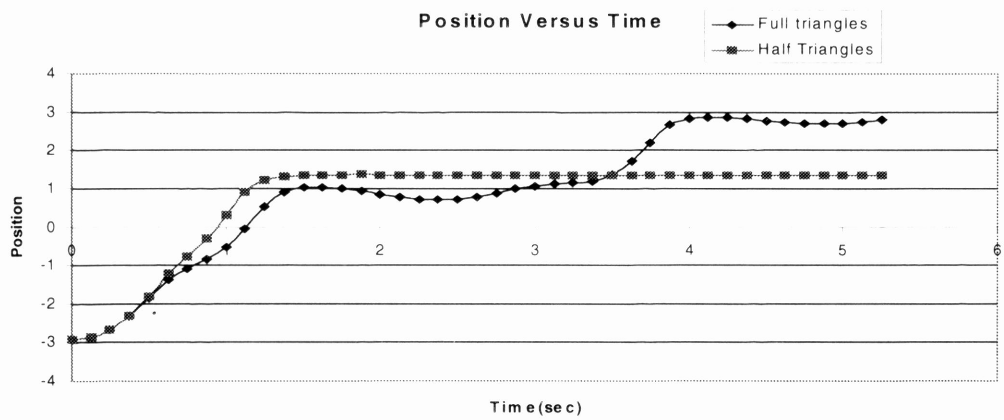


figure 10).

Discussion of Working Model Results

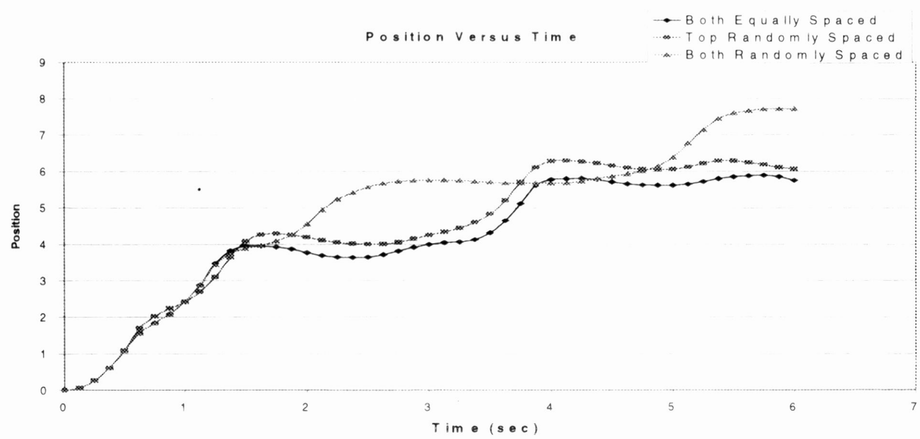
Figure 10

In figure 10, the flat part of the curve represents the point in time when the top surface could not force the mass peaks down. The top surface was stuck against the bottom surface. As seen above, for the same for, the moving surface was able to move farther than the half triangles.

Another parameter that was varied in order to determine the impact it had on the system was the distance between the mass peaks on both the top and bottom surfaces. In the first situation, the mass peaks were equally spaced along the top surface. Similarly, the mass peaks were equally spaced on the bottom surface. For the next simulation, the distances along the top surface were randomly chosen. The distances on the bottom surfaces were equal. For the third simulation, the distances between the mass peaks for both the top and bottom surfaces were chosen at random. As seen in figure 11, the simulations with randomly spaced asperities on both top and bottom surfaces required the smallest force to move the top surface over the bottom surface. This is due to smaller probablility that all of the mass peaks will be in contact with the other surface at the same time. The smaller the number of peaks in contact at any particular time, the less energy required to deform the bottom spring and allow the top surface to pass over (See figure 11).

Figure 11

In order to simulate the resistance to motion that an actual material shows,



dampers were placed in the Working Model simulation. The dampers simulated the materials resistance to deformation. The simulations for both the half triangles and the full triangles were ran. As shown in figure 12 the simulation predicted that the systems with the undamped mass peaks would travel farther than the damped mass peaks holding everything else constant. The reduction in final position is due to the dissapation of energy by the damper. The plateaus as seen in figure 12 indicate that the moving surface was unable to pass all of the mass peaks on the bottom surface. However, the undamped system did pass more of the mass peaks than the damped system. During any of the simulations, if the top surface was unable to move past the bottom surface it may have oscillated back and forth stopping each time at the same spot, or it came to a rest

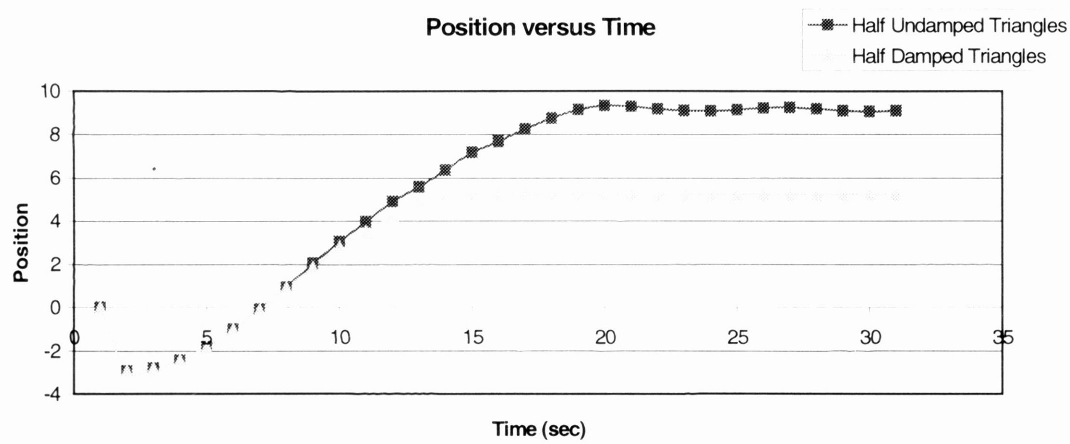


Figure 12

Working Model is a two dimensional kinematic simulation. It is easy to simulate the interaction between forces, masses and springs. Working Model is an excellent tool to produce a visual representation of the dyanamic responses of a system. Although some preliminary results were obtained with the surface friction model, the results were not indicating slip-stick friction. In order for slip-stick friction to take place, the moving mass would have to contact the bottom mass peak, decelerate, stop, begin building up the force pushing on the top surface, deform the bottom peak, and accelerate past the bottom asperity. This sequence of events never took place in the Working Model simulation. Once the simulation encountered a mass peak that it could not push down, it simply stopped. The primary reason it was not able to then go over the mass peak was due the constant force that was applied to moving surface. For slip-stick friction to occur the force acting on the moving surface can not be a constant force. It is difficult to simulate complex and intricate systems. Initially Working Model proved to be an asset since it produces a visual image of the systems behavior. However, we soon began to reach the limits of Working Model. It is not easy to distinguish what integration model Working Model is using to solve the differential equations. In addition, the user has no control over how Working Model sets up the differential equations. Initially, the values for the parameters in the simulations were arbitrary picked so that we could test if the computer program would run the simulations. We then entered in values for the parameters according to table 1 (See Appendix A for calculations).

Table 1

Parameter	**Calculated Value used in Working Model**
Density of Material, ρ	8.00 kg/m^3
Base of Asperity, b	1 μm
Cross Sectional Area, A	4E-12 m^2
Volume of Asperity, V	3.6E-17 m^3
Modulus of Elasticity, E	1.86E10 N/m^2
Spring Stiffness, K_{spring}	37,200 N/m
Mass of Asperities, M	2.88E-16 kg

These values were calculated in order to simulate a steel surface. However, when the values were entered and the simulation was started, Working Model was unable to calculate any values. We hypothesize that the large difference in the magnitude of the spring stiffness (37,200 N/m) and the cross-sectional area (4E-12 m^2) exceed the intentions of the program. Consequently we decided to use Matlab in order to further model the friction forces acting on the two surfaces.

In order to ensure that slip-stick friction could take place, the model of the moving surface was changed to the following (See figure 13).

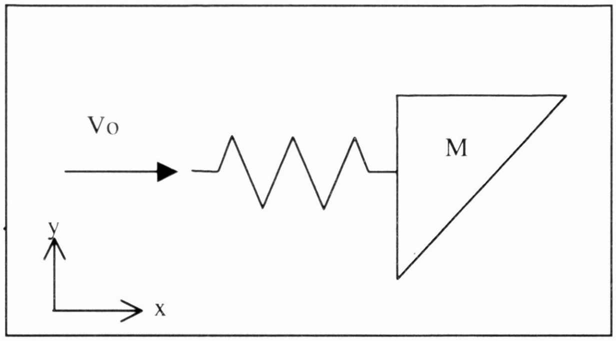


Figure 13

The sliding surface for Matlab was modeled as a single peak with mass, M, and a spring attached to the side. The free end of the spring is given a constant velocity Vo, and the mass spring system is constrained to move only in the x direction. Attaching a spring with one end at a constant velocity will supply the variable force needed for slip-stick friction. The bottom surface for Matlab simulation was modeled as the following (See figure 14).

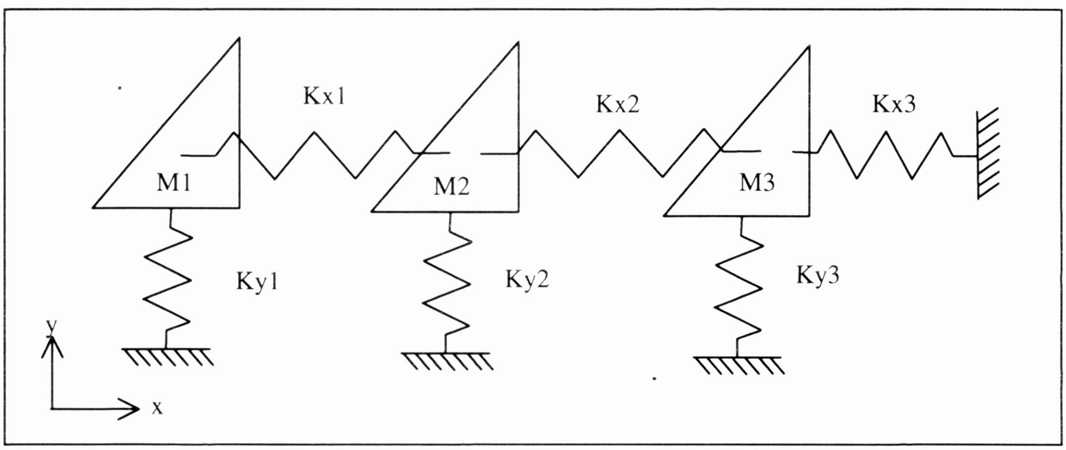


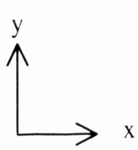
Figure 14

The mass peaks for the Matlab simulation are similar to the Working Model simulation in that they are individual peaks with finite masses that are attached to ground via springs. Instead of constraining the mass peaks to only translation in the y direction, the mass peaks can translate in the x direction. The mass peaks can not rotate. The springs connecting the mass peaks together simulate the connecting material on a real surface. In order to elastically deform the material in x direction, the material in the y direction must also deform. In order to simulate the path of the top surface, its motion was broken into a series of cases. Each case has a set of differential equations that governs the motion of the entire system (both top and bottom surfaces). The cases for the motion of the system are as follows (See table 2).

Table 2

Case #	Description
1	No contact between the two surfaces..
2	Contact with first mass peak.
3	No contact.
4	Contact with second mass peak.
5	No contact
6	Contact with third mass peak
7	No contact.

During cases 1, and 7 the equations of motion are defined by the top surface. During cases 2, 4 and 6 the equations of motion are defined by the contact between the top surface and the bottom surface. For cases 3 and 5, the of motion are governed by both the top surface, however, the excitation of the system because on the bottom surface is not exculded since it can translate in the x direction and alter the distance the top surface must travel before contacting the next peak. The free body diagrams for the systems are as follows (See figure 15). See Appendix B for complete system diagram and equations of motion used in the Matlab simulation.



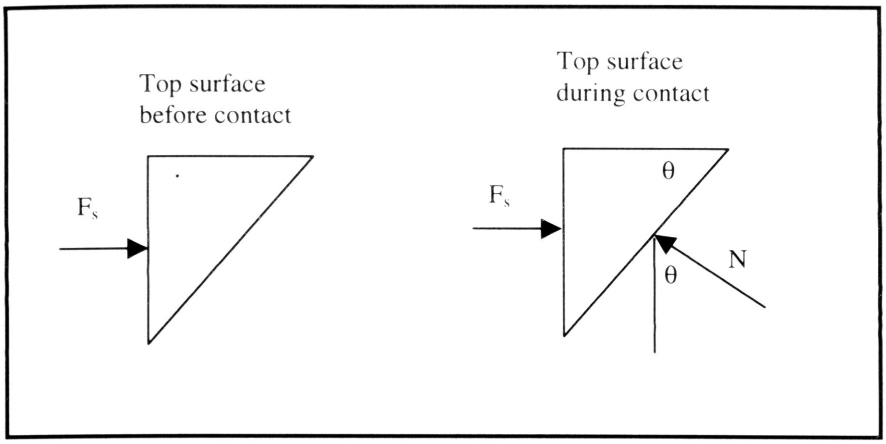

Top surface
before contact
F_s
Top surface
during contact
θ
F_s
N
θ

Figure 15

Before Contact:

$$F_s = Mass \cdot \ddot{x} \quad (2)$$

$$F_s = K_{spring}\left(V_o - \dot{x}\right) \quad (3)$$

where *Mass*, K_{spring}, and V_o are constants.

During Contact:

$$F_s - N\sin(\theta) = Mass \cdot \ddot{x} \quad (4)$$

where *N* is the contact force between the two surfaces.

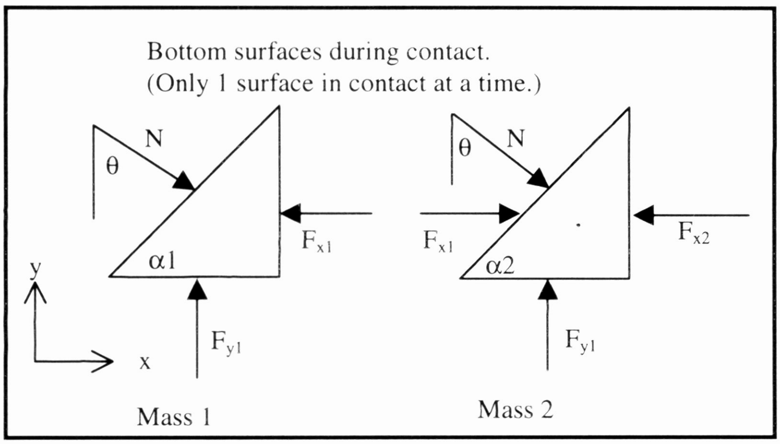


Figure 16

Equations of Motion for Mass 1:

$$N\sin(\theta) - F_{x1} = Mass \cdot \ddot{x} \quad (5)$$

$$-N\cos\theta + F_{y1} = Mass \cdot \ddot{y} \quad (6)$$

$$K_{x1}(x_1 - x_2) = F_{x1} \quad (7)$$

$$K_{y1}(y) = F_{y1} \quad (8)$$

See the figure in Appendix B for coordinate systems used.

Equations of Motion for Mass 2.

Only one bottom surface is in contact with the top surface at any given time.

$$F_{x1} + N\sin(6) - F_{x2} = Mass \cdot \ddot{x} \quad (9)$$

$$-N\cos(\theta) + F_{y2} = Mass \cdot \ddot{y} \quad (10)$$

$$K_{x2}(x_2 - x_3) = F_{x2} \quad (11)$$

$$N\sin(6) + F_{x1} - F_{x2} = Mass \cdot \ddot{x} \quad (12)$$

$$K_{y2} \cdot y = F_{y2} \quad (13)$$

Equations of Motion for Mass 3

$$F_{x2} + N\sin(6) - F_{x3} = Mass \cdot \ddot{x} \quad (14)$$

$$-N\cos(\theta) + F_{y3} = Mass \cdot \ddot{y} \, (15)$$

$$K_{x3}(x3) = F_{x3} \, (16)$$

$$N\sin(6) + F_{x2} - F_{x3} = Mass \cdot \ddot{x} \quad (17)$$

$$K_{y3} \cdot y = F_{y3} \quad (18)$$

where x_1 is defined by the coordinate system and x2, and x3 are relative to x1 and x2 respectively.

Once the general equations of motion were determined for the system, they were entered in to a Matlab m-file (See Appendix B). Since Matlab can not solve second order differntial equations, the equations were changed to the following format (See equations 19-23).

$$q(1) = x_{main} \, (19)$$

$$q(2) = d/dt(x_{main}) \, (20)$$

$$d/dt[q(1)]=q(2) \, (21)$$

where d/dt[x] is the derivative x with respect to t.

$$d/dt[q(1)]=d2/dt2[xmain] \quad (22)$$

$$d/dt[q(1)]=(Fmain-Nsin(theta))/Mmain \quad (23)$$

The equations are placed into a 1 x 22 matrix (See Appendix B). In order to run the simulation, I used a series of "if" statements in order to differentiate between contact and no contact. If there was contact between the top and bottom surfaces, the equations of motion were different than if there was no contact. In addition, the equations of motion are also dependant on which of the bottom mass peaks is in contact with the moving surface. Again, "if" statements were used to determine which mass peak was in contact with the top surface. Figure 17 shows the results one of the Matlab simulations.

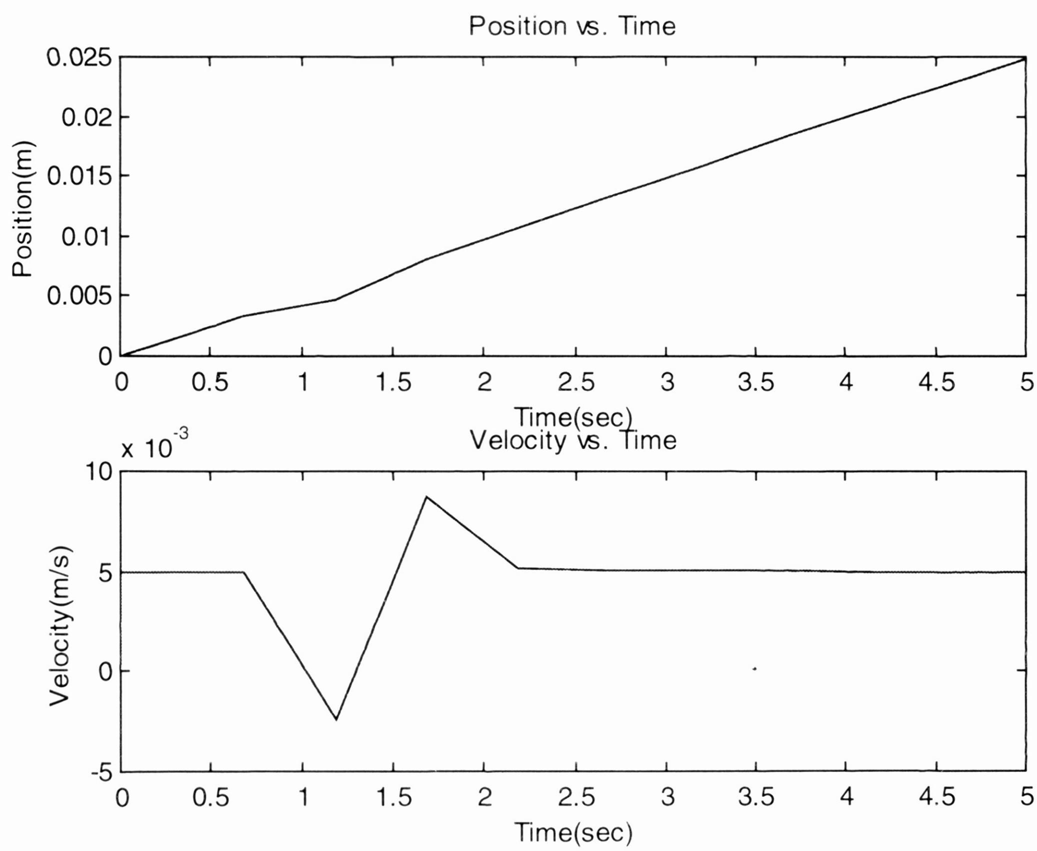


Figure 17

It is important to note that the parameters used in the Matlab simulation are those shown in table 1. Matlab is able to calculate the response of the system even though some of the parameters are very large and others are small. In the figures shown above, the top surface moves in the x-direction, contacts and passes over the first peak on the bottom. The surface then continues to move in the x-direction. The sharp spike in the velocity curve indicates the time when the two surfaces collide.

Conclusions

Modeling the friction force at or near zero velocity is an important task. If the friction model is able to capture the microscopic feature of the contact zone, then it can be confidently used in control systems. Many different industries would benefit greatly from a friction model that is accurate for velocities near zero. The military could use the improved motion control in their line of sight indicators. The medical profession could use the motion control system in order to guide surgical lasers during surgery with greater precision. The manufacturing industry could use the technology to improve machining.

Works Cited

1. Armstrong-Helouvry, B., "Stick Slip and Control in Low-Speed Motion". *IEEE Transactions on Automatic Control*, Vol 38, No. 10, October 1993.

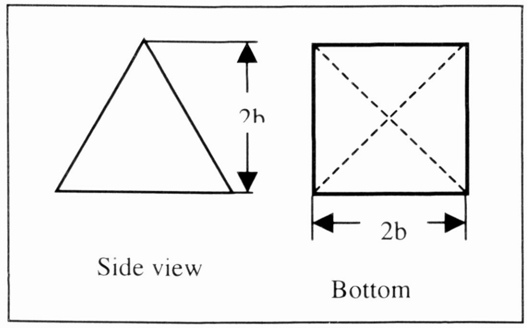
2. Armstrong-Helouvry,B, *Control of Machines with Friction*. Boston, Kluwer Academic Publishers, 1991.

3. Krim, J., "Friction at the Atomic Scale". *Scientific American*, October 1996, pg 74-80.

4. Surampudi, B., "Low Speed Joint Motion Control in Presence of Stic-Slip" Ph.D. Thesis. TEXAS A+M UNIVERSITY, AUG. 1995

Appendix A

Parameter Calculations for Working Model

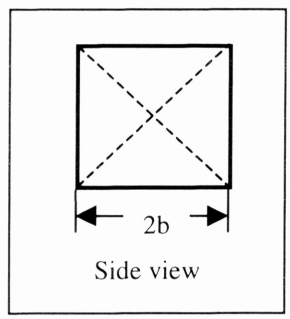


Cross Sectional Area at the Base:

Assumed it is a four sided pyramid.
$A=4b^2$
$B=1\mu m$
$A=4E\text{-}12 m^2$

Modulus of Elasticity:

E of steel is approximately 30E6 psi
$E=1.86E10\ N/m^2$



Volume of Pyramid:

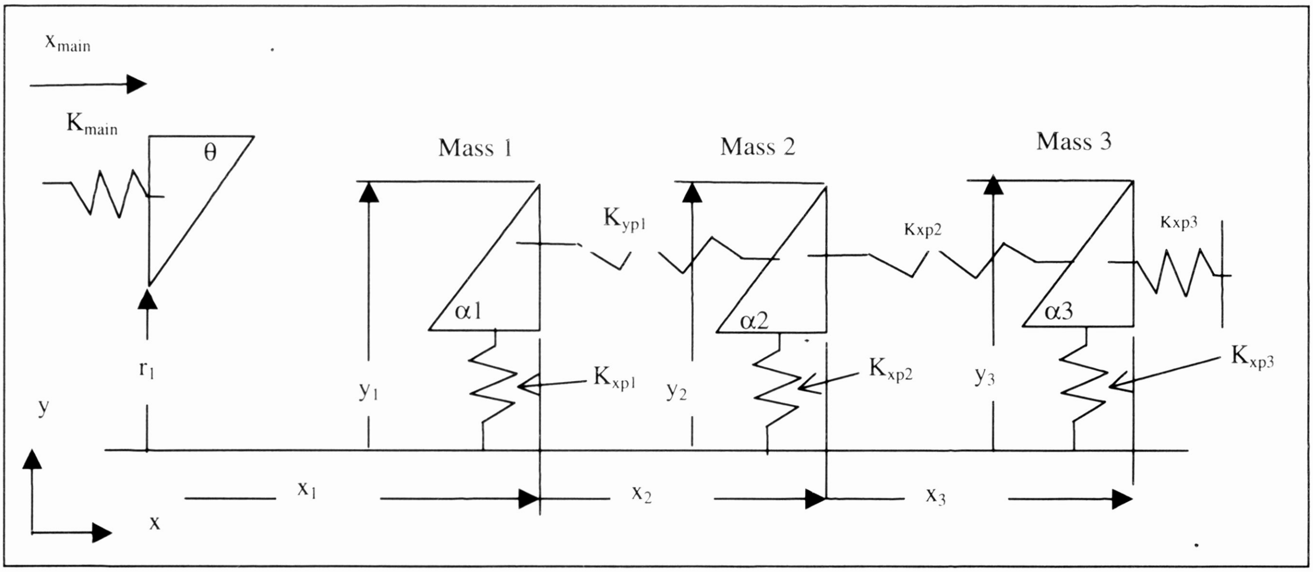
Assume 6 pyramids in a cube
Volume of Cube $=\text{base}^3$
Volume of Pyramid=1/6*Volume of Cube
$b=1\ \mu m$
$V=3.6E\text{-}17\ m^3$

Mass of Pyramid:

$M=\rho*V$
$\rho=8\ kg/m^3$
$M=2.88E\text{-}16$ kg

Apendix B – Matlab Simulation

Matlab System Diagram:



Equations used in Matlab

$q(1) = x_{main}$

$q(2) = d/dt[x_{main}]$

$q(3) = F_{main}$

$q(4) = x_{p1}$

$q(5) = d/dt[x_{p1}]$

$q(6) = F_{xp1}$

$q(7) = y_{p1}$

$q(8) = d/dt[y_{p1}]$

$q(9) = F_{yp1}$

$q(10) = N$

$q(11) = x_{p2}$

$q(12) = d/dt[x_{p2}]$

$q(13) = F_{xp2}$

$q(14) = y_{p2}$

$q(15) =d/dt[yp2]$

$q(16) = x_{p3}$

$q(17) = d/dt[x_{pe}]$

$q(18) = F_{xp3}$

$q(19) = y_{p3}$

$q(20)=d/dt[y_{p3}]$

$q(21)=F_{yp2}$
$q(22)=F_{yp3}$

```
%This function calculates the equations and the variables
function qdot=mainspring(t,q,flag,told,Nold,Vconstant);

deltat=t(1)-told;

Mmain=2.88e-16; %Mass of the moving surface
Kmain=37200; %Spring constant of the moving surface
Mp1=2.88e-16; %Mass of peak one
Mp2=2.88e-16; %Mass of peak two
Mp3=2.88e-16; %Mass of peak three

undeformed=1e-6; %Undeformed length of spring on moving surface
Vconstant %Constant velocity
alpha1=45*2*3.14159/360; %Internal angles of the triangles
alpha2=45*2*3.14159/360;
alpha3=45*2*3.14159/360;
theta=45*2*3.14159/360;

r1=1e-6; %Height of moving tip
y1=1.5e-6; %Height of triangle one at the tip
y2=1.5e-6; %Height of triangle two at the tip
y3=1.5e-6; %Height of triangle three at the tip
y=[y1 y2 y3]; %Matrix of triangle heights
r=[r1 r1 r1];
x=[q(1) q(1)+q(11) q(1)+q(11)+q(16)];
xc(1)=(y-r)*[1;0;0]*(1/tan(alpha1)); %xc equals contact distances
xc(2)=(y-r)*[0;1;0]*(1/tan(alpha2));
xc(3)=(y-r)*[0;0;1]*(1/tan(alpha3));
Contact=x-xc;

Kxp1=37200; %Spring Constants
Kxp2=37200;
Kxp3=37200;
Kyp1=37200;
Kyp2=37200;
Kyp3=37200;

if q(1)<Contact(1);
   qdot(1)=q(2);
   qdot(2)=q(3)/Mmain;
   qdot(3)=Kmain*(Vconstant-q(2));
   qdot=[qdot(1);qdot(2);qdot(3);0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];
end;

if q(1)>=Contact(1);
   if q(1)<=q(4);
      qdot(1)=q(2);
      qdot(2)=(q(3)-q(10)*sin(theta))/Mmain;
      qdot(3)=Kmain*(Vconstant-q(2));
      qdot(4)=q(5);
      qdot(5)=(q(10)*sin(theta)-q(6))/Mp1;
      qdot(6)=Kxp1*(q(5)-q(12));
      qdot(7)=q(8);
      qdot(8)=(-q(10)*cos(theta)+q(9))/Mp1;
      qdot(9)=Kyp1*q(8);
      qdot(10)=(Nold-q(10))/deltat+q(4)-q(1)-xc(1);
      qdot(11)=q(12);
      qdot(12)=(q(6)-q(13))/Mp2;
      qdot(13)=Kxp2*(q(12)-q(17));
      qdot(14)=q(15);
      qdot(15)=0;
      qdot(16)=q(17);
      qdot(17)=(q(13)-q(18))/Mp3;
      qdot(18)=Kxp3*q(17);
      qdot(19)=q(20);
      qdot(20)=0;
      qdot(21)=0;
      qdot(22)=0;
```

```
      qdot=[qdot(1);qdot(2);qdot(3);qdot(4);qdot(5);qdot(6);qdot(7);qdot(8);qdot(9);qdot(1
0);qdot(11);qdot(12);qdot(13);qdot(14);qdot(15);qdot(16);qdot(17);qdot(18);qdot(19);qdot(2
0);qdot(21);qdot(22)];
   elseif q(1)<Contact(2);
      qdot(1)=q(2);
      qdot(2)=q(3)/Mmain;
      qdot(3)=Kmain*(Vconstant-q(2));
      qdot(4)=q(5);
      qdot(5)=-q(6)/Mp1;
      qdot(6)=Kxp1*(q(5)-q(12));
      qdot(7)=q(8);
      qdot(8)=q(9)/Mp1;
      qdot(9)=Kyp1*q(8);
      qdot(10)=0;
      qdot(11)=q(12);
      qdot(12)=(q(6)-q(13))/Mp2;
      qdot(13)=Kxp2*(q(12)-q(17));
      qdot(14)=q(15);
      qdot(15)=0;
      qdot(16)=q(17);
      qdot(17)=(q(13)-q(18))/Mp3;
      qdot(18)=Kxp3*q(17);
      qdot(19)=q(20);
      qdot(20)=0;
      qdot(21)=0;
      qdot(22)=0;
      qdot=[qdot(1);qdot(2);qdot(3);qdot(4);qdot(5);qdot(6);qdot(7);qdot(8);qdot(9);qdot(1
0);qdot(11);qdot(12);qdot(13);qdot(14);qdot(15);qdot(16);qdot(17);qdot(18);qdot(19);qdot(2
0);qdot(21);qdot(22)];
   end;
end;

if q(1)>=Contact(2);
   if q(1)<=(q(4)+q(11));
      qdot(1)=q(2);
      qdot(2)=(q(3)-q(10)*sin(theta))/Mmain;
      qdot(3)=Kmain*(Vconstant-q(2));
      qdot(4)=q(5);
      qdot(5)=-q(6)/Mp1;
      qdot(6)=Kxp1*(q(5)-q(12));
      qdot(7)=q(8);
      qdot(8)=q(9)/Mp1;
      qdot(9)=Kyp1*q(8);
      qdot(10)=(Nold-q(10))/deltat+q(4)+q(11)-q(1)+xc(2);
      qdot(11)=q(12);
      qdot(12)=(q(6)-q(13)+q(10)*sin(theta))/Mp2;
      qdot(13)=Kxp2*(q(12)-q(17));
      qdot(14)=q(15);
      qdot(15)=(-q(10)*cos(theta)+q(21))/Mp2;
      qdot(16)=q(17);
      qdot(17)=(q(13)-q(18))/Mp3;
      qdot(18)=Kxp3*q(17);
      qdot(19)=q(20);
      qdot(20)=0;
      qdot(21)=Kyp2*q(15);
      qdot(22)=0;
      qdot=[qdot(1);qdot(2);qdot(3);qdot(4);qdot(5);qdot(6);qdot(7);qdot(8);qdot(9);qdot(1
0);qdot(11);qdot(12);qdot(13);qdot(14);qdot(15);qdot(16);qdot(17);qdot(18);qdot(19);qdot(2
0);qdot(21);qdot(22)];
   elseif q(1)<Contact(3);
      qdot(1)=q(2);
      qdot(2)=q(3)/Mmain;
      qdot(3)=Kmain*(Vconstant-q(2));
      qdot(4)=q(5);
      qdot(5)=-q(6)/Mp1;
      qdot(6)=Kxp1*(q(5)-q(12));
      qdot(7)=q(8);
      qdot(8)=q(9)/Mp1;
```

```
      qdot(9)=Kyp1*q(8);
      qdot(10)=0;
      qdot(11)=q(12);
      qdot(12)=(q(6)-q(13))/Mp2;
      qdot(13)=Kxp2*(q(12)-q(17));
      qdot(14)=q(15);
      qdot(15)=q(21)/Mp2;
      qdot(16)=q(17);
      qdot(17)=(q(13)-q(18))/Mp3;
      qdot(18)=Kxp3*q(17);
      qdot(19)=q(20);
      qdot(20)=0;
      qdot(21)=Kyp2*q(15);
      qdot(22)=0;
      qdot=[qdot(1);qdot(2);qdot(3);qdot(4);qdot(5);qdot(6);qdot(7);qdot(8);qdot(9);qdot(1
0);qdot(11);qdot(12);qdot(13);qdot(14);qdot(15);qdot(16);qdot(17);qdot(18);qdot(19);qdot(2
0);qdot(21);qdot(22)];
   end;
end;

if q(1)>=Contact(3);
   if q(1)<=(q(4)+q(11)+q(16));
      qdot(1)=q(2);
      qdot(2)=(q(3)-q(10)*sin(theta))/Mmain;
      qdot(3)=Kmain*(Vconstant-q(2));
      qdot(4)=q(5);
      qdot(5)=-q(6)/Mp1;
      qdot(6)=Kxp1*(q(5)-q(12));
      qdot(7)=q(8);
      qdot(8)=q(9)/Mp1;
      qdot(9)=Kyp1*q(8);
      qdot(10)=(Nold+q(10))/deltat+q(11)+q(16)-q(1)-xc(3);
      qdot(11)=q(12);
      qdot(12)=(q(6)-q(13))/Mp2;
      qdot(13)=Kxp2*(q(12)-q(17));
      qdot(14)=q(15);
      qdot(15)=q(13)/Mp2;
      qdot(16)=q(17);
      qdot(17)=(q(13)-q(18))/Mp3;
      qdot(18)=Kxp3*q(17);
      qdot(19)=q(20);
      qdot(20)=(-q(10)*cos(theta)+q(22))/Mp3;
      qdot(21)=Kyp2*q(15);
      qdot(22)=Kyp3*q(20);
      qdot=[qdot(1);qdot(2);qdot(3);qdot(4);qdot(5);qdot(6);qdot(7);qdot(8);qdot(9);qdot(1
0);qdot(11);qdot(12);qdot(13);qdot(14);qdot(15);qdot(16);qdot(17);qdot(18);qdot(19);qdot(2
0);qdot(21);qdot(22)];
   else
      qdot(1)=q(2);
      qdot(2)=q(3)/Mmain;
      qdot(3)=Kmain*(Vconstant-q(2));
      qdot(4)=q(5);
      qdot(5)=-q(6)/Mp1;
      qdot(6)=Kxp1*(q(5)-q(12));
      qdot(7)=q(8);
      qdot(8)=q(9)/Mp1;
      qdot(9)=Kyp1*q(8);
      qdot(10)=0;
      qdot(11)=q(12);
      qdot(12)=(q(6)-q(12))/Mp2;
      qdot(13)=Kxp2*(q(12)-q(17));
      qdot(14)=q(15);
      qdot(15)=q(13)/Mp2;
      qdot(16)=q(17);
      qdot(17)=(q(13)-q(18))/Mp3;
      qdot(18)=Kxp3*q(17);
      qdot(19)=q(20);
      qdot(20)=q(22)/Mp3;
```

```
%This m-file runs the Matlab simulation
%It calls the m-file Mainspring that contains the differential equations

global Nold
global told
global Vconstant

Vconstant=5e-3;
options=odeset('Reltol',10,'AbsTol',10);

initalcond=[0;0;0;1;0;0;0;0;0;0;1;0;0;0;0;1;0;0;0;0;0;0];
Nold=0;
told=1;
tspan=[0 5];
[t,data]=ode23S('mainspring',tspan,initalcond,options,told,Nold,Vconstant);
plot(t(:,1),data(:,1));
```

```
      qdot(21)=Kyp2*q(15);
      qdot(22)=Kyp3*q(20);
      qdot=[qdot(1);qdot(2);qdot(3);qdot(4);qdot(5);qdot(6);qdot(7);qdot(8);qdot(9);qdot(1
0);qdot(11);qdot(12);qdot(13);qdot(14);qdot(15);qdot(16);qdot(17);qdot(18);qdot(19);qdot(2
0);qdot(21);qdot(22)];
   end;
end;

told=t(1);
Nold=q(10);
```