Alexander Chew
999766439
Andrew Choi
999796522
Navea Dasz
998884752

EME 171 Lab #3: Motorcross Pitch-Heave Model

**Bond Graph**

A bond graph of the system was drawn to visually represent the power flow and velocities as shown below in Figure 1:

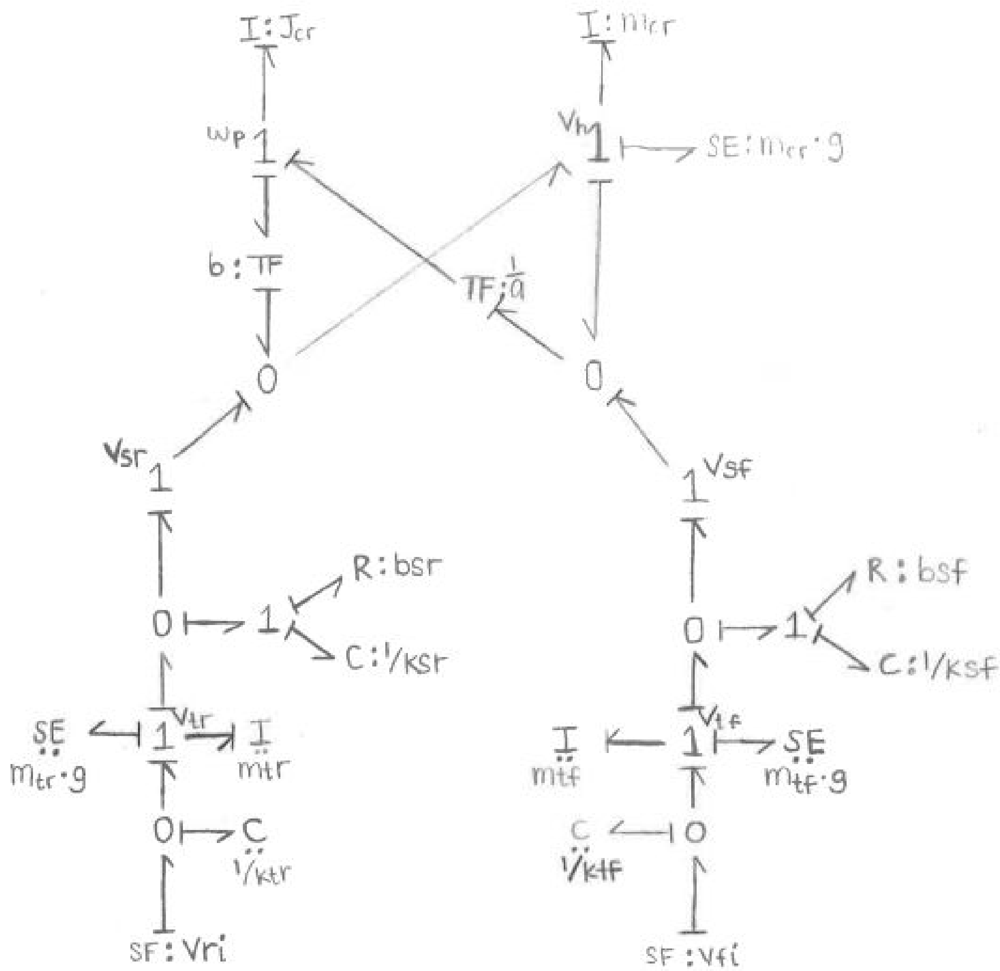


**Figure 1:** Bond graph of the system

**Initial Conditions**

The initial conditions were determined from the equations of motion and using statics. The initial momentums are all equal to zero because the system is in equilibrium. The initial displacements were developed using statics, finding the force supported by the front and rear suspension, then applying the forces to the equation for displacement of a spring under a force to yield the equations below:

$$\mathrm{fssd} = \frac{\mathrm{mcr} * \mathrm{b} * \mathrm{g}}{\mathrm{ksf} * \mathrm{Lwb}}$$

$$\mathrm{rssd} = \frac{\mathrm{mcr} * \mathrm{a} * \mathrm{g}}{\mathrm{ksr} * \mathrm{Lwb}}$$

$$\mathrm{ftd} = \frac{\mathrm{mcr} * \mathrm{b} * \mathrm{g}}{\mathrm{ktf} * \mathrm{Lwb}} + \frac{\mathrm{mtf} * \mathrm{g}}{\mathrm{ktf}}$$

$$\mathrm{rtd} = \frac{\mathrm{mcr} * \mathrm{a} * \mathrm{g}}{\mathrm{ktr} * \mathrm{Lwb}} + \frac{\mathrm{mtr} * \mathrm{g}}{\mathrm{ktr}}$$

**Time Control Parameters**

The time control parameters were determined by approximating the range of natural frequencies of the system. Three models, the heave model, pitch model, and wheel model, were used to symbolize the various vibration periods and are shown in Figure 2 below.

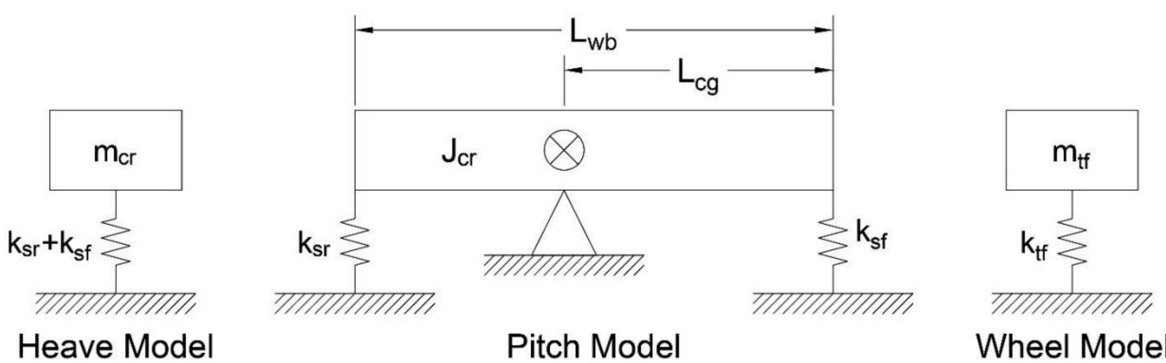


**Figure 2:** The three models used to determine time control parameters: heave model, pitch model, and wheel model

The equation for vibration periods is the inverse of the natural frequency, with the radial frequency being the square root of the ratio between stiffness and inertia:

$$T = \frac{1}{f_n}, f_n = \frac{\omega_n}{2\pi}, \omega_n = \sqrt{\frac{\text{stiffness}}{\text{inertia}}}$$

*Solving for Heave Model*

For the heave model, sum up the rear and front suspension stiffnesses and divide by the mass of the cycle and rider. Square root this value then to obtain the natural frequency.

$$\omega_{nH} = \sqrt{\frac{k_{sr} + k_{sf}}{m_{cr}}} = \sqrt{\frac{6500}{300}} = 4.655\,\frac{rad}{s}$$

$$f_{nH} = \frac{4.655}{2\pi} = 0.7409\text{ Hz}$$

$$T_{nH} = {f_{nH}}^{-1} = 1.35s$$

*Solving for Pitch Model*

For the pitch model, the approach is a little less straightforward. To obtain the angular stiffness we must sum up the torque felt with respect to the fulcrum. The small angle theorem can be used as the angular displacement is quite small. These displacements multiplied by their respective stiffnesses resemble the forces that are creating the torque. Angular stiffness is then a ratio of torque and angular displacement which can be simplified as shown below. Finally, divide this stiffness by the polar moment of inertia and square root to obtain the natural frequency of the pitch model.

$$T = k_{sr}b^2\theta + k_{sf}a^2\theta$$

$$k_{angular} = \frac{T}{\theta} = k_{sr}b^2 + k_{sf}a^2$$

$$k_{angular} = k_{sr}(L_{wb} - L_{cg})^2 + k_{sf}{L_{cg}}^2$$

$$k_{angular} = 3500(1.6 - 0.9)^2 + 3000 * 0.9^2 = 4145$$

$$\omega_{nP} = \sqrt{\frac{k_{angular}}{J_{cr}}} = \sqrt{\frac{k_{angular}}{m_{cr} * {r_{gy}}^2}}$$

$$\omega_{nP} = \sqrt{\frac{4145}{75}} = 7.434\text{ rad/s}$$

$$f_{nP} = \frac{7.434}{2\pi} = 1.1831\text{ Hz}$$

$$T_{nP} = {f_{nP}}^{-1} = 0.8452s$$

*Solving for Wheel Model*

Finally, for the wheel model, simply divide the front tire stiffness by the mass of the front tire and square root.

$$\omega_{nW} = \sqrt{\frac{k_{tf}}{m_{tf}}} = \sqrt{\frac{30000}{15}} = 44.72\,\frac{\text{rad}}{\text{s}}$$

$$f_{nW} = \frac{44.72}{2\pi} = 7.1174\ \text{Hz}$$

$$T_{nW} = {f_{nW}}^{-1} = 0.1405\text{s}$$

Once we have all the vibration periods solved for, we can set the proper time control parameters. The maximum step size is to be at most one-tenth of the shortest vibration period or one-tenth the time it takes to go over one half of one bump. Going over half of one bump takes 0.25s → 0.025s. Dividing the smallest vibration period by ten results in 0.01405s. As this is smaller, this becomes our maximum step size.

The finish time is to be set to three times the longest vibration period after the rear tire reaches the end of the second bump. Our simulation sets the time at which this happens to 0.92s so the finish time will be 4.97s.

With our max step size and finish time known, we can then insert our time control parameters into MatLab as linspace(0, 4.97, 354).

**Maximum Bump Height**

The maximum bump height at which the cycle can go over the bumps without bottoming out either the front or rear suspension is calculated using the linear relationship between the suspension deflection and bump height:

$$\frac{A_1}{A_{max}} = \frac{\delta_1}{\delta_{max}} \quad \rightarrow \quad A_{max} = \frac{A_1 * \delta_{max}}{\delta_1}$$

where the max suspension deflection $\delta max$ is 0.1m, $A_1$ is an estimated bump height, and $\delta_1$ is the max suspension deflection at that bump height. $\delta_1$ is determined using the matlab max() command to find the max deflection of both the front and rear, and the A and $\delta$ values are determined through simulation. $A_{max}$ is determined to be 0.161m.

## Plots

*Standard Configuration*

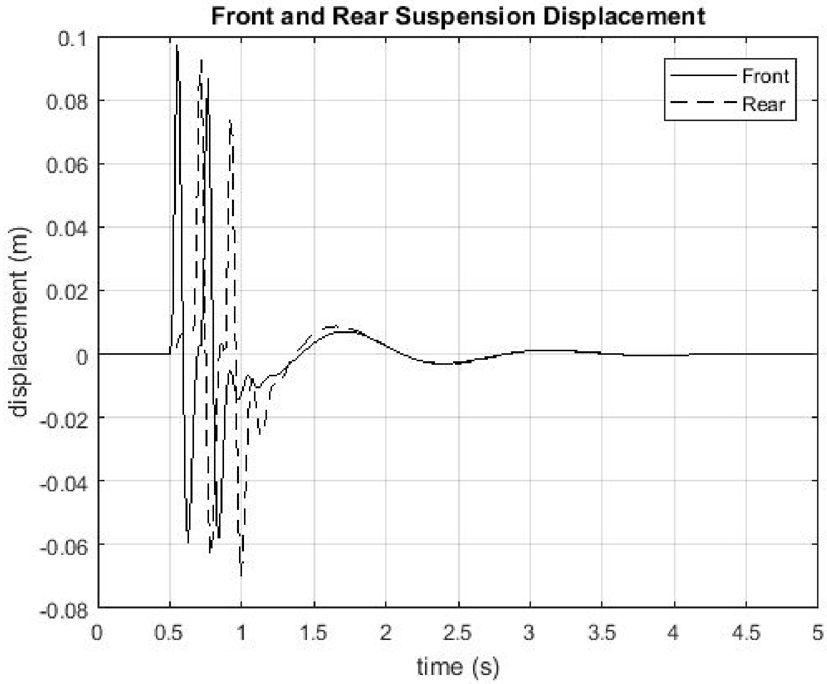


**Figure 3:** Front and Rear Suspension Displacement for the standard CG configuration ($L_{CG}$=0.9m)

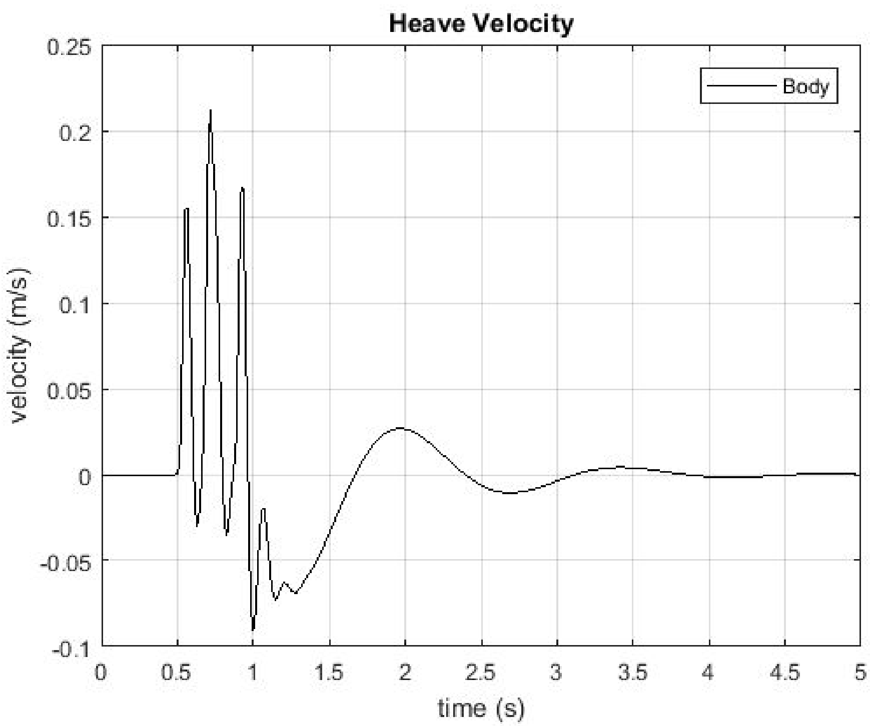


**Figure 4:** Heave velocity for the standard CG configuration ($L_{CG}$=0.9m)

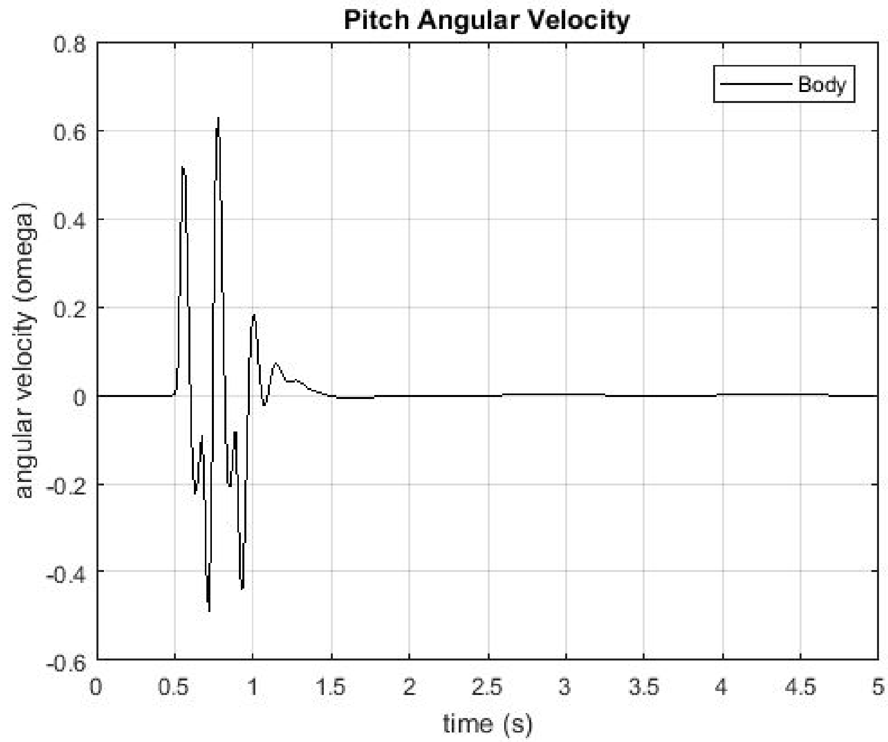


**Figure 5:** Pitch angular velocity for standard CG configuration ($L_{CG}$=0.9m)

*Forward Configuration*

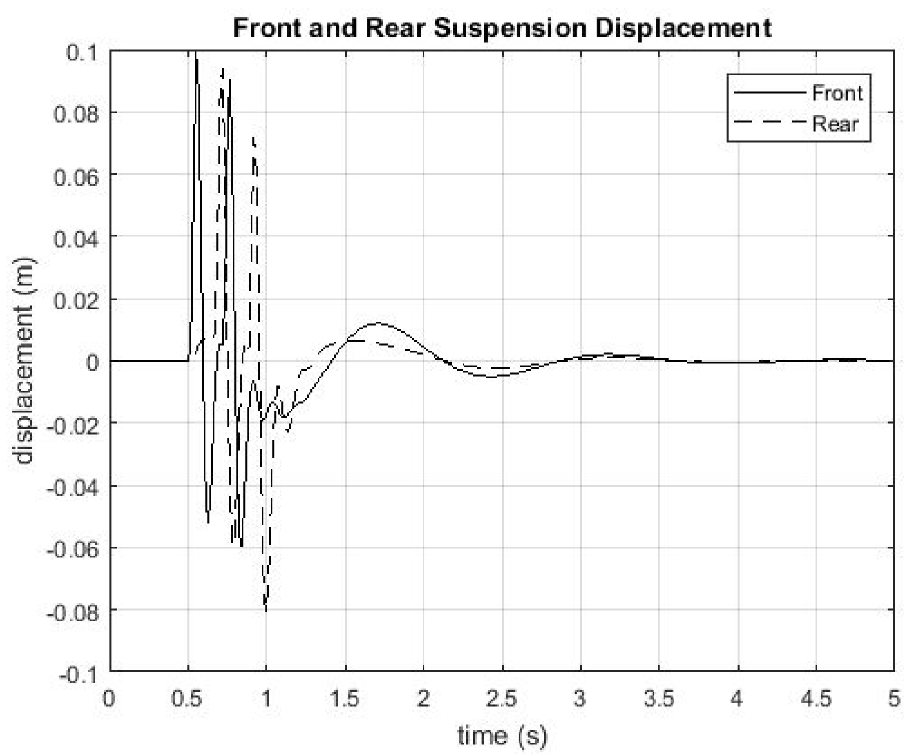


**Figure 6:** Front and rear suspension displacement for forward configuration ($L_{CG}$=0.7m)

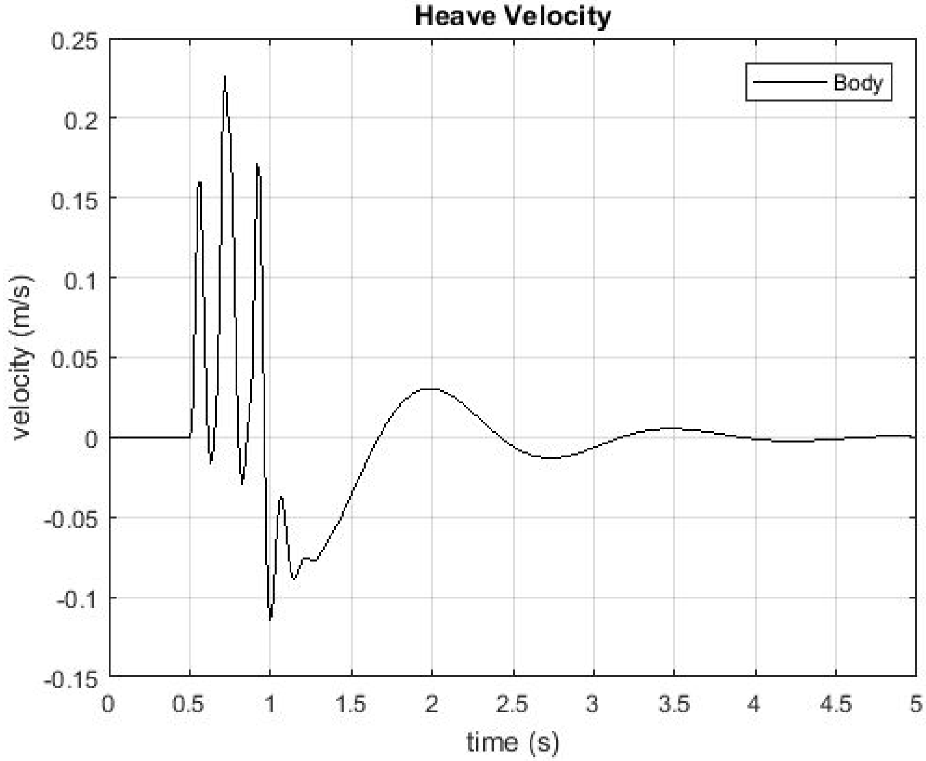


**Figure 7:** Heave velocity for forward configuration ($L_{CG}$=0.7m)

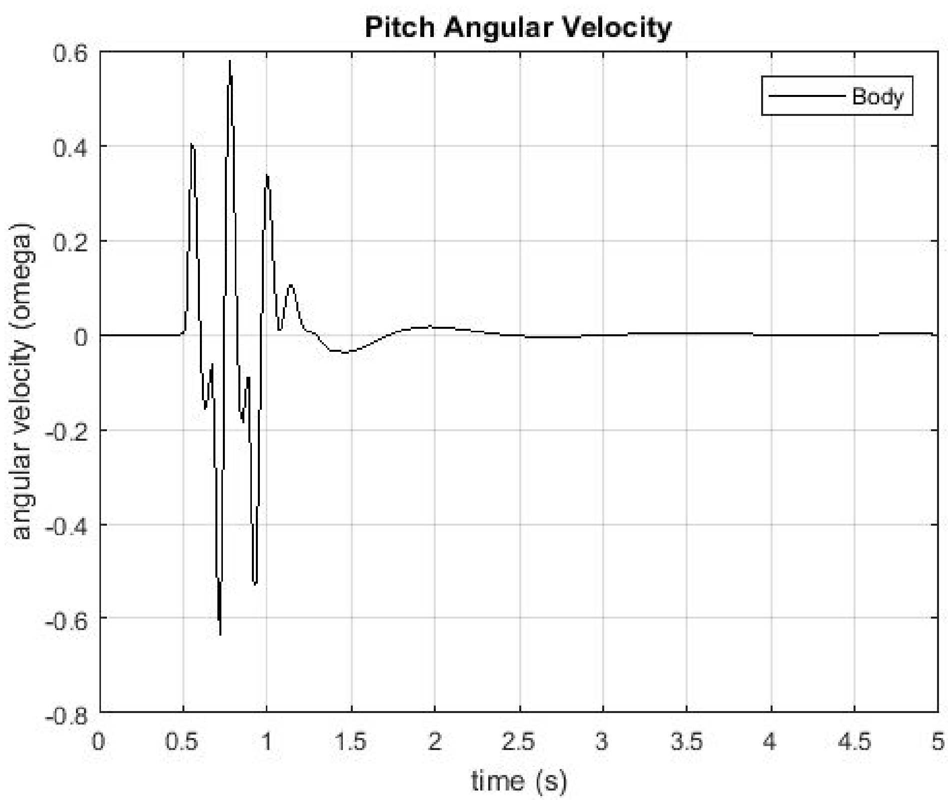


**Figure 8:** Pitch angular velocity for forward configuration ($L_{CG}$=0.7m)

**Questions**

1. The natural frequencies of the system are the heave, pitch, and wheel natural frequencies, and these were calculated to be 4.655 rad/s, 7.434 rad/s, and 44.72 rad/s, respectively. From these calculated natural frequencies, the vibration periods were determined to be 1.35s, 0.8452s, and 0.1405s for the heave, pitch, and wheel, respectively.

   The sample time and simulation length are determined from the vibration period. Using the vibration period calculated from the wheel frequency, the maximum step size is found to be 0.01405 seconds. The finish time is three of the longest vibration periods after the rear tire reaches the end of the second bump, so using the period calculated from the heave model, the finish time is 4.97 seconds.

2. According to the power flow on the bond graph, the deflections of the suspension are positive in compression. This is because we made flow positive going upwards to coincide with our flow sources. Due to gravity, this positive flow direction results in compression of the springs and damper making compression positive in our suspension. If we were to have chosen the down direction as our positive flow direction, then tension would become positive.

3. The shift in center of mass towards the front in the forward configuration made for a slightly greater maximum suspension deflection in the front suspension as well as some differences in pitch angular velocity, however the graphs show the system was not changed drastically by the change in center of mass.

4. By combining the kinematic equations, $v_{sf} = v_h + a\omega_p$ and $v_{sr} = v_h - b\omega_p$, the relationship between the front and rear velocities can be shown for the heave velocity and pitch angular velocity. Solving for both gives:

$$V_h = \frac{b * v_{sf}}{b + a} + \frac{a * v_{sr}}{b + a}$$

$$\omega_p = \frac{v_{sf} - v_{sr}}{b + a}$$

   There is a positive relationship between the front and rear velocities in the heave velocity equation, which explains why the spikes in the plot are in the same direction. In the pitch angular velocity equation, it shows that there is a negative relationship between the front and rear velocities, which explains the opposing directions of the spikes in that plot.

*Bonus Question*

There's compression in a tire due to the weight of the rider and bike against a stagnant surface, in this case this is either the road or bump. For there to be tension in the wheel, the rider and bikes acceleration would need to be moving up, and the tire would also need to be pulled down to the road, as if it were clamped or stuck to the road. This doesn't often happen in real life, although it is possible that the tire could be stuck to the road if it were caught in a rock or some other obstruction.  The model would need to change by adding a effort source to the tires.

Lab3_master.m

```
global vc Lcg mcr rgy ksf ksr bsf bsr mtf mtr ktf ktr Lwb L g Jcr A
vc = 10;
Lcg = 0.7;
mcr = 300;
rgy = 0.5;
ksf = 3000;
ksr = 3500;
bsf = 400;
bsr = 500;
mtf = 15;
mtr = 20;
ktf = 30000;
ktr = 40000;
Lwb = 1.6;
A = 0.161;
L = 0.5;
g = 9.81;

Jcr = mcr*(rgy^2);

a = Lcg; %defition of distance a
b = Lwb-Lcg; % definition of distance b

fssd = mcr*b*g/(ksf*Lwb); % Initial front suspension displacement
rssd = mcr*a*g/(ksr*Lwb); % Initial rear suspension displacement
ftd = mcr*b*g/(ktf*Lwb) + mtf*g/ktf; % Initial front tire deflection
rtd = mcr*a*g/(ktr*Lwb) + mtr*g/ktr; % Initial rear tire deflection

initial = [0 0 fssd rssd 0 0 ftd rtd];
%s(1) = Pitch angular momentum
%s(2) = Verical momentum of cycle and rider
%s(3) = Front suspension spring displacement
%s(4) = Rear suspension spring displacement
%s(5) = Momentum of front tire mass
%s(6) = Momentum of rear tire mass
%s(7) = Front tire deflection
%s(8) = Rear tire deflection
```

```
tspan = linspace(0,4.97,326);

[t, s] = ode45(@lab3_eqns,tspan,initial);
ext = zeros(length(t),2);
ds = zeros(length(t),8);

for i = 1:length(t)
   [ds(i,:), ext(i,:)] = lab3_eqns(t(i), s(i,:));
end

susf = max(s(:,3)-fssd); % max front suspension displacement
susr = max(s(:,4)-rssd); % Max rear suspension displacement

figure('Name','displacements','NumberTitle','off','Color','white')
plot(t,s(:,3)-fssd,'k',t,s(:,4)-rssd,'--k'), grid on
title('Front and Rear Suspension Displacement')
legend('Front','Rear')
ylabel('displacement (m)')
xlabel('time (s)')

figure('Name','Heave Velocity','NumberTitle','off','Color','white')
plot(t,s(:,2)/mcr,'k'), grid on
title('Heave Velocity')
legend('Body')
ylabel('velocity (m/s)')
xlabel('time (s)')

figure('Name','pitch angular velocity','NumberTitle','off','Color','white')
plot(t,s(:,1)/Jcr,'k'), grid on
title('Pitch Angular Velocity')
legend('Body')
ylabel('angular velocity (omega)')
xlabel('time (s)')
% figure('Name','dforce','NumberTitle','off')
% plot(t,ext(:,1),'k'), grid on
% title('Suspension Damper Force')
% ylabel('force (N)')
% xlabel('time (s)')
```

## Lab3_eqns.m

```
function [ds, ext] = lab3_eqns(t,s)
global vc Lcg mcr rgy ksf ksr bsf bsr mtf mtr ktf ktr Lwb L g Jcr A

pj = s(1); % Pitch angular momentum
pcr = s(2); % Vertical momentum of cycle and rider
qsf = s(3); % Front suspension spring displacement
```

```
qsr = s(4); % Rear suspension spring displacement
ptf = s(5); % Momentum of front tire mass
ptr = s(6); % Momentum of rear tire mass
qtf = s(7); % Front tire deflection
qtr = s(8); % Rear tire deflection

a = Lcg;
b = Lwb - Lcg;

T1 = 0.5; %start of front wheel on first bump
T2 = L/(2*vc) + T1; %apex of fron wheel on first bump
T3 = L/vc + T1; % end of front wheel on first bump
T4 = T3 + Lwb/vc; % start of front wheel on second bump
T5 = T4 + L/(2*vc); % apex of front wheel on second bump
T6 = T4 + L/vc; % end of front wheel on second bump
T7 = T1 + Lwb/vc; % start of rear wheel on first bump
T8 = T7 + L/(2*vc); % apex of rear wheel on first bump
T9 = T7 + L/vc; % end of rear wheel on first bump
T10 = T9 + Lwb/vc; % start of rear wheel on second bump
T11 = T10 + L/(2*vc); % apex of rear wheel on second bump
T12 = T10 + L/vc;% end of rear wheel on second bump

Vup = 2*A*vc/L;
Vdown = - Vup;

if t < T1 % defining time parameters of front input velocity
  vfi = 0;
elseif t >= T1 && t <= T2
  vfi = Vup;
elseif t > T2 && t <= T3
  vfi = Vdown;
elseif t > T3 && t <= T4
  vfi = 0;
elseif t > T4 && t <= T5
  vfi = Vup;
elseif t > T5 && t <= T6
  vfi = Vdown;
else
  vfi = 0;
end

if t < T7  % defining time parameters for rear input velocity
  vri = 0;
```

```
elseif t >= T7 && t <= T8
    vri = Vup;
elseif t > T8 && t <= T9
    vri = Vdown;
elseif t > T9 && t <= T10
    vri = 0;
elseif t > T10 && t <= T11
    vri = Vup;
elseif t > T11 && t <= T12
    vri = Vdown;
else
    vri = 0;
end

pj_dot =a*(qsf*ksf+bsf*(ptf/mtf-pcr/mcr-a*pj/Jcr))-b*(qsr*ksr+bsr*(ptr/mtr-pcr/mcr+b*pj/Jcr));
pcr_dot = -mcr*g+qsf*ksf+bsf*(ptf/mtf-pcr/mcr-a*pj/Jcr)+qsr*ksr+bsr*(ptr/mtr-pcr/mcr+b*pj/Jcr);
qsf_dot = ptf/mtf-pcr/mcr-a*pj/Jcr;
qsr_dot = ptr/mtr-pcr/mcr+b*pj/Jcr;
ptf_dot = qtf*ktf-mtf*g-qsf*ksf-bsf*(ptf/mtf-pcr/mcr-a*pj/Jcr);
ptr_dot = qtr*ktr-mtr*g-qsr*ksr-bsr*(ptr/mtr-pcr/mcr+b*pj/Jcr);
qtf_dot = vfi - ptf/mtf;
qtr_dot = vri - ptr/mtr;

ext(1) = vfi;
ext(2) = vri;

ds = [pj_dot; pcr_dot; qsf_dot; qsr_dot; ptf_dot; ptr_dot; qtf_dot; qtr_dot];
```