Simulation 1: 1/8th Car

EME 271

Brandon Reddish

System Design

The one eighth car model can be represented by a mass, spring and damper system. The diagram in Figure 1 shows this simplified model.

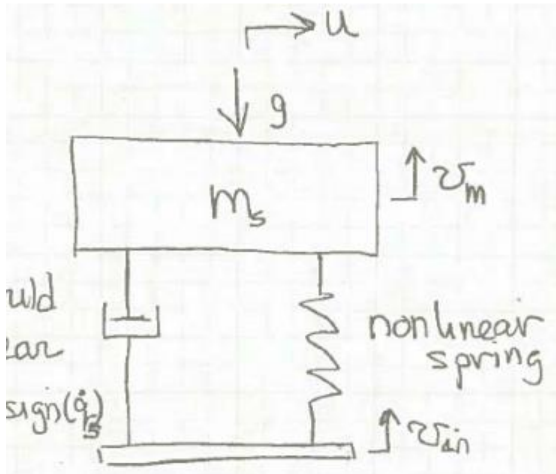

Figure 1 – One Eighth Car Model Diagram

Figure 2 shows the Bond Graph representation of the system.

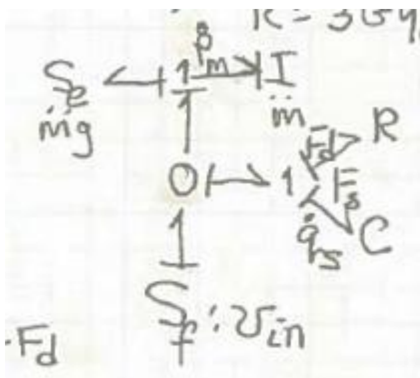

Figure 2 – Bond Graph of System

Equations of Motion

The equations of motion begin with the input values for vin. This value represents the surface of the road way pushing against the tire that is connected to the suspension. In this scenario the system was tested by simulating a bump in the road defined by equation 1.

$$Y_{in} = A * 0.5 * (1 - \cos(2 * \pi\ (x/\lambda))) \tag{1}$$

For the purposes of modeling the system, the rate of change of Y with respect to X is needed as an input into the dynamic system. Taking the derivative of equation 1 with respect to X produced equation 2.

$$\frac{dY_{in}}{dX} = A * 0.5 * \left(\frac{2\pi}{\lambda}\right) sin(2 * \pi \ \frac{x}{\lambda}) \tag{2}$$

With this equation, and the assumption that the X velocity is constant, dy_in/dx can be determined by multiplying the value time by the speed to get the x location and ultimately, the rate of change of Y. Equation 3 shows the use of the forward velocity to get the vertical velocity.

$$v_{in} = U \ \frac{dY_{in}}{dx} \tag{3}$$

With the input known, the next step is to define the equations for the spring and damper. For the spring, the force acting on the mass is modeled by equation 4 for a nonlinear system.

$$F_s = Gq_s^3 \tag{4}$$

Because the car rests on the spring at equilibrium, the value for G can be represented with equation 5.

$$G = \frac{m_s g}{q_e^3} \tag{5}$$

In order to model the spring as a linear system, the value for the spring constant can be determined by using equation 6.

$$k_{eq} = 3Gq_s^2 \tag{6}$$

As a result, the linear equation depends only on the spring constant derived in equation 6 times the spring displacement.

$$F_s = k_{eq}q_s \tag{7}$$

Using the spring constant from the spring, the following equations were used to characterize the behavior of the linear damper in the system.

$$f_s = \frac{1}{2\pi}\sqrt{\left\{\frac{k_{eq}}{m_s}\right\}} \tag{8}$$

$$b_{eq} = 2\,\zeta(2\pi f_s)m_s \tag{9}$$

Similar to a spring constant, the damping factor can be multiplied by the relative velocity of the input plate and the mass to generate the resulting force. Equations 10 and 11 show the final form of the damper equation.

$$F_d = b_{eq}v_{rel} \tag{10}$$

$$v_{rel} = v_{in} - v_{in} \tag{11}$$

For a nonlinear model of the damping in this system, equation 12 was used to add the friction force as a nonlinear term that is only depends on the sign of the motion.

$$F_d = 0.5\, m_s g\; sign(\dot{q}_s) \tag{12}$$

The parameters used for the simulation are shown in table 1.

Table 1 – System Parameters

Parameter	**Value**
Mass	3000/2.2 kg
Spring state at equilibrium	0.25 meters
Bump length	1 meter
Bump amplitude	0.5 meters
Forward speed	10 mph
Damping ratio	0.3

With the system components defined, the state equations can be calculated. Equations 13 and 14 describe the two state variables of the system.

$$\dot{p_m} = -m_s g + F_s + F_d \tag{13}$$

$$\dot{q}_s = v_{in} - \frac{p_m}{m_s} \tag{14}$$

Simulation Results

The results of the simulation are shown in Figure 3 and Figure 4. Each figure has a plot for the three combination of the linear and nonlinear spring and damper.

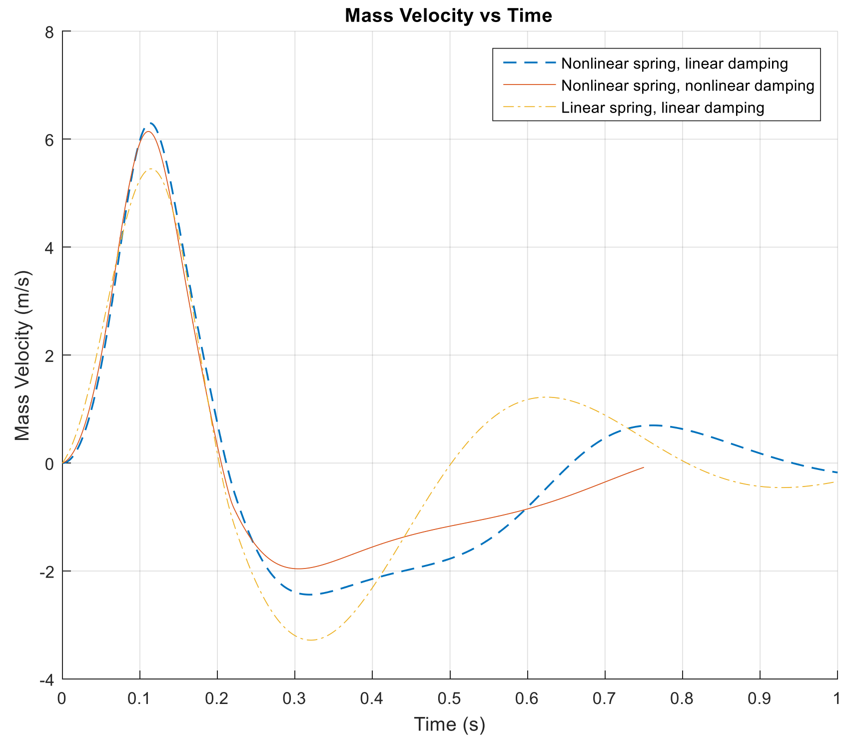

Figure 3 – Mass Velocity vs Time, A=0.5m U=10mph

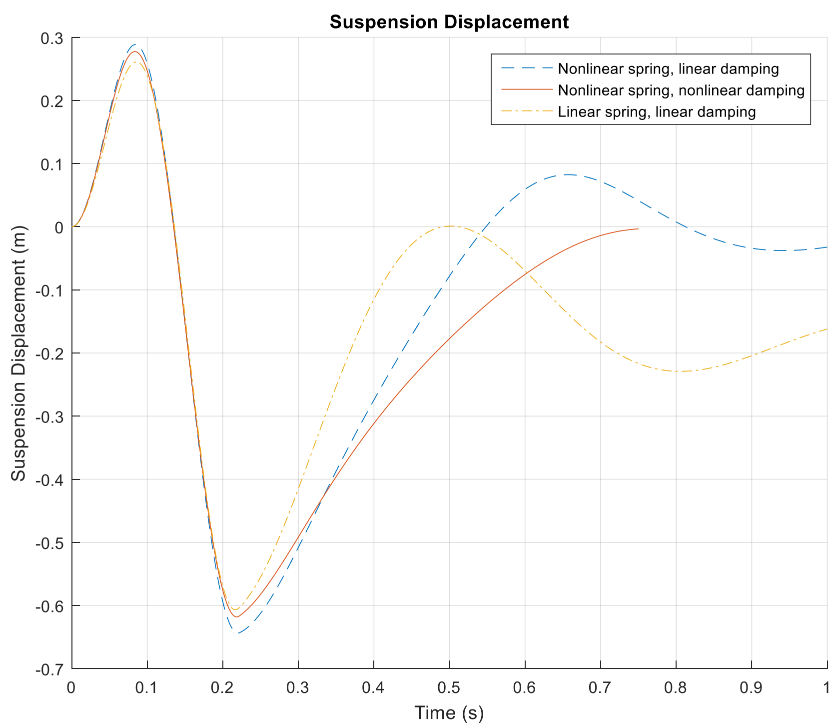

Figure 4 – Suspension Displacement, A=0.5m U=10mph

The biggest difference between the linear and nonlinear systems is the shape of the curves for both suspension displacement and mass velocity. The linear model has a periodic shape that maintains a smooth curve through out the simulation. The nonlinear results show more than a diminishing oscillation. The nonlinear curve does not reach as negative of a velocity as the linear solution and takes more time to return to a positive velocity upward. The simulation with a nonlinear spring and a linear damper looks most similar to the the purely nonlinear system which suggests that the unique behavior of the nonlinear system comes mostly from the properties of the spring rather than the damper.

The nonlinear damping adds the additional friction force to the equation and as a result the amplitude of the result for the spring displacement and mass velocity are slightly less than the linear model of the damper.

Appendix A – Code

The following code was used to generate a solution to the mechanical system. Another script was used to make the plot, but it was not included because it is not relevant to the set up and finding of the lab.

EME 271 Simulation Number 1

Eighth Car Over Bump

```
function main()
```

Set Parameters

```
global mass roadLen g amp vel G beq qe
% Given values
mass  = 3000/2.2; % mass (kg)
qe = 0.25; % spring compression at equilibrium (m)
roadLen = 1; % road length (m)
g = 9.8; % gravitational constant (m/s^2)
dampRatio = 0.3; % damping ratio (non-dimentional)
amp = 0.5; % amplitude (m)
vel = 10*0.45; % car velocity (m/s)
```

Spring constants

```
G = (mass*g)/(qe^3); % non-linear spring const
keq = 3 * G * qe^2; % linear const
```

Damper constants

```
fs = (1/(2*pi))*sqrt(keq/mass); % frequency
beq = 2*dampRatio*(2*pi*fs)*mass; % damping constant
```

Set boundary conditions and run parameters

```
tfinal = 1; % time end of the simulation (sec)
tspan = linspace(0,tfinal,2000);
initials = [0.25 0]; % The weight of the car causes 0.25m of initial disp
```

ode45

```
[t,x] = ode45(@Equations,tspan,initials);
```

```
end
```

Equation Function

```
function [dx, oth] = Equations(t,x)
```

```
% Global vars
global mass roadLen g amp vel G beq qe
```

Unpack

```
qs = x(1); % Spring state
pm = x(2); % mass momentum
```

Calculate dy/dx from time and velocity

```
dist = vel * t; % distance
if dist < roadLen
    dyindx = amp*0.5 *2*(pi/roadLen)*sin(2*pi*(dist/roadLen)); % rate of change of input vertical velocity
    vin = vel * dyindx; % vertical velocity input (m/s)
else
    vin = 0;
end
vm = pm/mass; %vertical velocity of mass
dqs = vin-vm; % relative velocity
```

Spring Calculation

Both linear and nonlinear equations are shown below for the spring and damper. I commented out whichever one I wasn't using, but kept it there to easily switch back and forth between methods.

```
springForce = 3*G*(qe^2)*qs; % Linear spring
% springForce = G*qs^3; % Non-linear spring
```

Damper Calc

```
damperForce = beq*dqs;% Linear damper
% damperForce = 0.5*mass*g*sign(dqs) + beq*dqs; % Non-linear damp
```

Calculate rate of change

dqs = vin - (pm/mass); %relative velocity of car mass to wheel

```
dpm = -mass*g + springForce + damperForce; % rate of change of mass momentum
dx = [dqs;dpm];
oth = damperForce;
```

```
end
```