Simulation 5: Active Control Suspension

Brandon Reddish

February 9, 2020

1 Introduction

This lab examines the effect of an active control system on a model of a car suspension. The system will be modeled as shown in Figure 1.

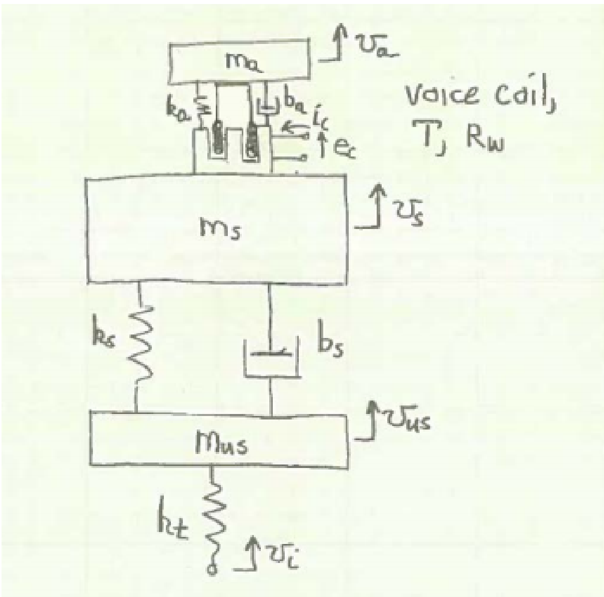

Figure 1: Model of Mass Striking a Barrier

This system will be moving the the horizontal direction at some speed U and the input velocity at the bottom of the spring v_i will be the result of a bumpy road surface. In addition to the sprung and unsprung mass, there will be an additional mass attached to the top of the sprung mass with a spring, damper, and voice coil. The voice coil will be controlled by some current input. The objective is to design a controller that applies a current to the voice coils that will generate a force that will counteract the acceleration of the sprung mass. As a result, the motion of the sprung mass will be smoother than if a normal suspension was implemented.

1.1 Bond Graph Model and State Equations

The bond graph shown in Figure 2 is a representation of the suspension system with the active control system. The voice coil is modeled as a flow source with a winding resistance and a gyrator that relates the current of the flow to the force applied by the voice coil.

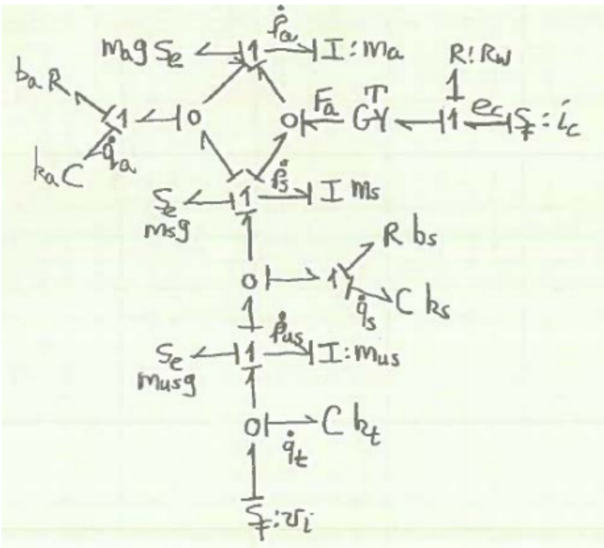

Figure 2: Simplified Model and Bond Graph

This model has six state equations and they are as follows:

$$\dot{q}_t = v_i - \frac{p_{us}}{m_{us}} \tag{1}$$

$$\dot{p}_{us} = q_t k_t - m_s g - \Big((\frac{p_{us}}{m_{us}} - \frac{p_s}{m_s}) b_s + q_s k_s \Big) \tag{2}$$

$$\dot{q}_s = \frac{p_{us}}{m_{us}} - \frac{p_s}{m_s} \tag{3}$$

$$\dot{p}_s = \Big(\frac{p_{us}}{m_{us}} - \frac{p_s}{m_s} \Big) b_s + q + s k_s = m_s g - \Big((\frac{p_s}{m_s} - \frac{p_a}{m_a}) b_a + q_a k_a \Big) - T i_c \tag{4}$$

$$\dot{q}_a = \frac{p_s}{m_s} - \frac{p_a}{m_a} \tag{5}$$

$$\dot{p}_a = (\frac{p_s}{m_s} - \frac{p_a}{m_a}) b_a + q_a k_a - m_a g + T i_c \tag{6}$$

1.2 Input Parameters

The main input to the system will be the vertical velocity of the lower spring caused by the road. This velocity will be calculated by multiplying the forward velocity by the slope of the surface of the road at a given location. The shape of the road will be randomly generated with a mean slope of zero to simulate a road with no incline with bumps. Once the shape of the road is defined, the following equation is used to determine the value of v_i.

$$v_i = U * slope \tag{7}$$

For this simulation, the following parameters will be used to compare the passive suspension with the controlled suspension.

```
g = 9.8                    Gravitational constant
m_tot = 3000./2.2          Total vehicle mass
msmus=5                    Sprung to unsprung mass ratio
m_us = m_tot/(1+msmus)     Unsprung mass
m_s=m_tot-m_us             Sprung mass
w_s=2*pi*1.2               Suspension frequency
k_s=m_s*w_s^2              Suspension stiffness
zeta_s=.7, zeta_c=.7       Damping ratio for passive and active system.
                           These will be varied.
b_s=2*zeta_s*w_s*m_s       Suspension damping constant
b_c=2*zeta_c*w_s*m_s       Effective damping constant for control
w_wh=2*pi*8 Wheel          Hop frequency
k_t=m_us*w_wh^2            Tire stiffness
R_w=.005                   Winding resistance, Ohm
T=5                        Coupling constant, Nm/A
m_a=.02*m_s                Actuator mass, kg
w_a=2*pi*5                 Actuator frequency
k_a=m_a*w_a^2              Actuator stiffness
b_a=2*.1*w_a*m_a           Actuator damping
U=40*.46                   Trial vehicle velocity, m/s
```

1.3 Controller

The controller will be used to counteract the acceleration of the sprung mass. One model for this type of controller can be represented with the following equation:

$$F_a = b_c v_s \tag{8}$$

Because the controller can only control the current, this equation becomes:

$$F_a = T i_c \tag{9}$$

Combining these two equations we can solve for the input of the controller i.

$$i_c = \frac{b_c v_s}{T} \tag{10}$$

2 Running the Simulation

For the first simulation, the value of $z_s = 0.7$ for the passive system. This will be compared with a second scenario where the value of $z_s = 0.1$ and $z_c = 0.7$ with the active control turned on. Figures 3 through 5 show the results of the passive and active control systems.

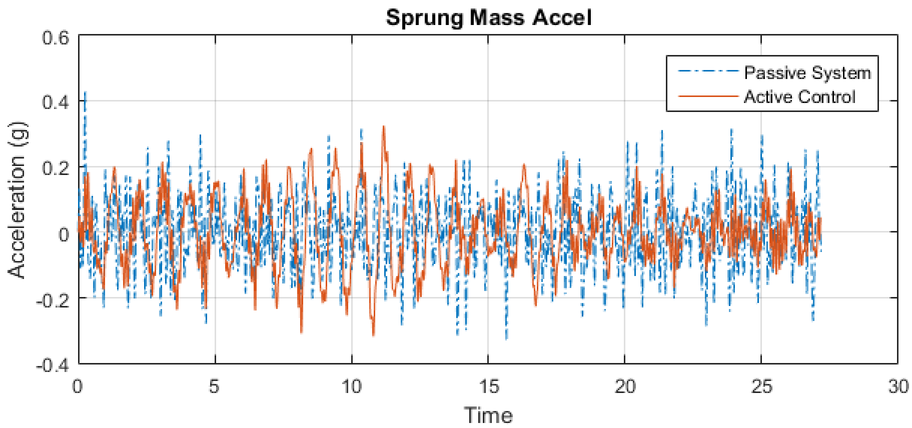

Figure 3: Acceleration of Sprung Mass

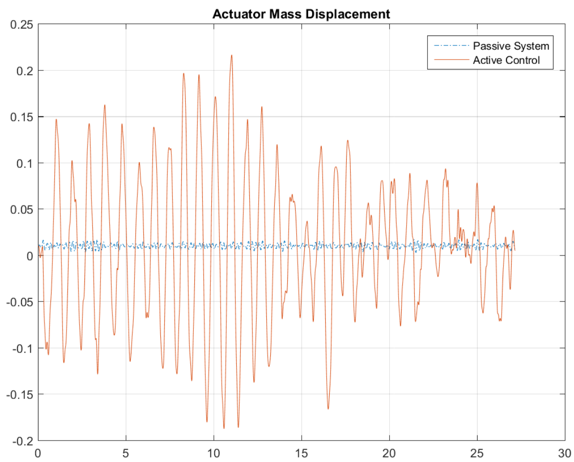

Figure 4: Actuator Mass Displacement

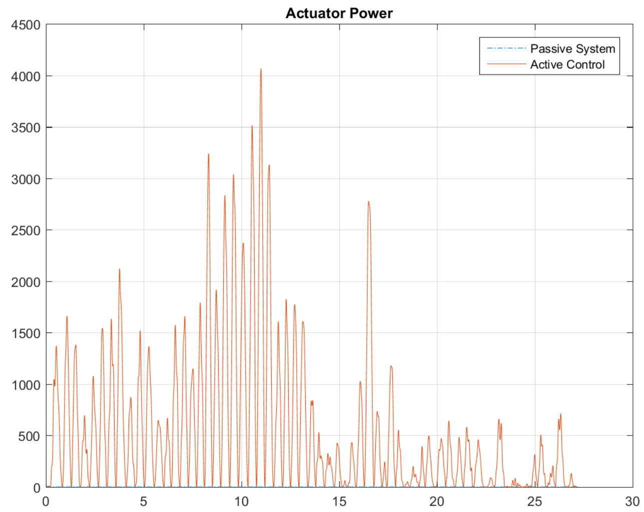

Figure 5: Actuator Power

From looking at the acceleration plot, one can see that the active controller does reduce the amount of large spikes in acceleration. By reducing the magnitude of the highest g loads on the sprung mass, the ride would feel smoother if this was applied to a car suspension. However, there is a cost in displacement and power. The actuator mass has a maximum displacement of over 20 cm which is more than a car designer would like to allocate for this device. Additionally, the power required goes over 3 kW at some locations which too high of a power demand.

To reduce the amount of power required, a manufacturer could spend more money on an actuator with less winding resistance and a greater coupling constant. In an extreme case, the resistance could be lowered to $R_w = 0.001$ and the coupling constant can be increased to $T = 10$.To fix the issue with the actuator mass movement, one could increase the stiffness of the actuator spring and increased the mass of the actuator. In this example, we will increase the stiffness of the spring by 40% and the mass will be 4% of the sprung mass. Figures 6 through 8 show the impact of these results.

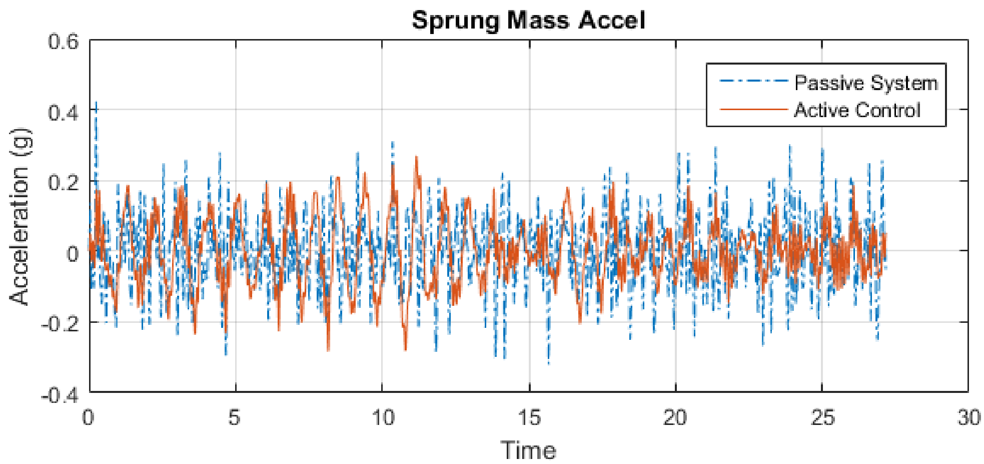

Figure 6: Acceleration of Sprung Mass - Improved

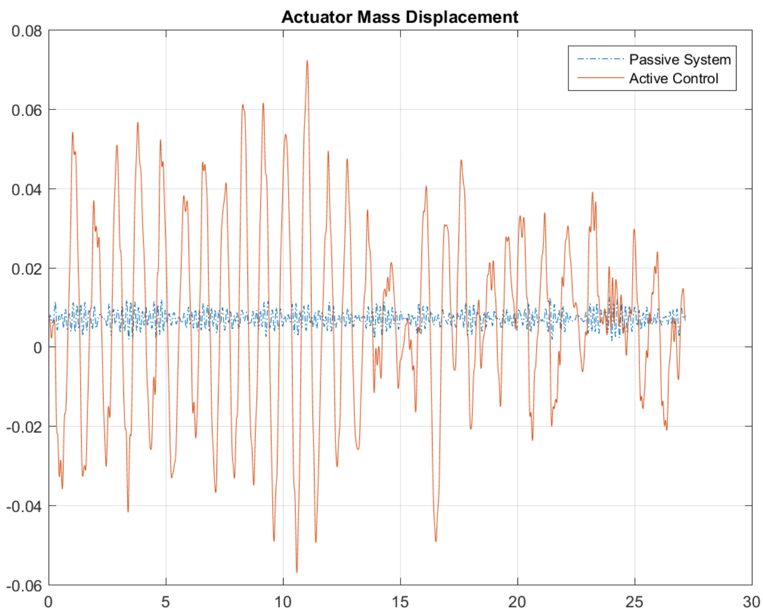

Figure 7: Actuator Mass Displacement - Improved

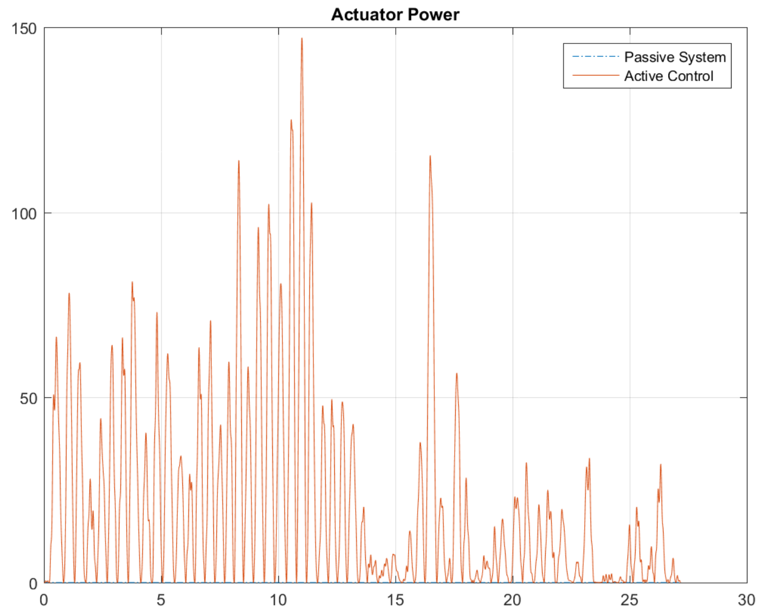

Figure 8: Actuator Power - Improved

By making these changes, the power and displacement were both reduced to values that are practical for implementation on a moving vehicle. Although this control system may make for a smoother ride, the additional complexity may not be worth it. Adding a voice coil and a controller would increase the price of the vehicle and create more complexity in the design of the car. For an average car that stays on paved roads, this system may be overkill. For high end luxury cars or off road vehicles these systems may prove to be profitable. This analysis is preliminary and is built on many assumptions about the model. If a company were to refine this system and bring the maximum acceleration significantly lower, then this may be a marketable feature. However, for the controller designed in this lab, the improved riding conditions do not outweigh any significant increase in cost.

3 MATLAB Code

3.1 Main Script

```
function main()
%% Set parameters
global mus g kt ms bs ks ka T ma ba U X_i slope_i Rw bc flag
close all
%% Parameters and initial conditions
%build a random road input
rng('default');
delta_x=.5; Length=500;% This defines a length of a road in meters
```

```
X_i=0:delta_x:Length;% This establishes a length vector
n_pts=fix(Length/delta_x);
slope_raw=randn(n_pts+1,1);% This generates uniformly distributed random number that we interpret
slope_i=.007*(slope_raw-mean(slope_raw));
%This is the slope vector that has any average value removed.
%This makes the road zero mean slope. The scalin  number at the front makes the passive vehicle
%without control have a sprung mass acceleration that is
%reasonable. I experimented to determine this number.
for flag = [1 2]
    g = 9.8;
    m_tot=3000./2.2;
    msmus=5;
    mus=m_tot/(1+msmus); %unsprung mass
    ms=m_tot-mus; %Sprung mass
    w_s=2*pi*1.2; %Suspension frequency
    ks=ms*w_s^2; %Suspension stiffness
    if flag ==1
        zeta_s=.7;
        zeta_c=.7; %Damping ratio for passive and active
    else
        zeta_s=.1;
        zeta_c=.7; %Damping ratio for passive and active
    end
    bs=2*zeta_s*w_s*ms; %Suspension damping constant
    bc=2*zeta_c*w_s*ms; %Effective damping constant for control
    w_wh=2*pi*8; %Wheel hop frequency
    kt=mus*w_wh^2; %Tire stiffness
    Rw=.001; %Winding resistance, Ohm
    T=10; %Nm/A Coupling constant
    ma=.04*ms; %Actuator mass, kg
    w_a=2*pi*5; %Actuator frequency
    ka=ma*w_a^2*1.4; %Actuator stiffness
    ba=2*.1*w_a*ma; %Actuator damping
    U=40*.46; %m/s Trial vehicle velocity

    %% Initial conditions
    qti =(m_tot+ma)*g/kt;
    qsi = (ma+ms)*g/ks;
    qai = ma*g/ka;

    %% Run setup
    %          [qt pus qs ps qa pa]
    initials = [qti 0 qsi 0 qai 0];
    timeStep = 0.001;
    tfinal = Length/U;
    tspan = 0:timeStep:tfinal;

    %% Run ODE
    [t,x] = ode45(@equations,tspan,initials);
```

```
    %Extract derivatives and other values
    for i = 1:length(t)
        [dx(i,:),oth(i,:)] = equations(t(i),x(i,:));
    end
    %% Plotting
    figure(1)
    plot(t,(dx(:,4)/ms)/g)
    title('Sprung Mass Accel')
    xlabel('Time')
    ylabel('Acceleration (g)')
    daspect([12 1 1])
    grid on
    hold on

    figure(2)
    plot(t,x(:,5))
    title('Actuator Mass Displacement')
    grid on
    hold on

    figure(3)
    plot(t,oth(:,2))
    title('Actuator Power')
    grid on
    hold on
end
figure(1)
hline = findobj(gcf, 'type', 'line');
set(hline(2),'LineStyle','-.')
legend('Passive System','Active Control')
print('accel','-dpng')
figure(2)
hline = findobj(gcf, 'type', 'line');
set(hline(2),'LineStyle','-.')
legend('Passive System','Active Control')
print('disp','-dpng')
figure(3)
hline = findobj(gcf, 'type', 'line');
set(hline(2),'LineStyle','-.')
legend('Passive System','Active Control')
print('power','-dpng')
end
```

3.2 Equations

```
function [dx,oth] = equations(t,x)
% Global variables
global mus g kt ms bs ks ka T ma ba U X_i slope_i Rw bc flag
```

```
% Unpack variables
qt=x(1);
pus=x(2);
qs=x(3);
ps=x(4);
qa=x(5);
pa=x(6);

X=U*t;
slope=interp1(X_i,slope_i,X);
% slope=0;
vi = U*slope;
vs = ps/ms;

if flag ==1
    Fa=0;
    Sf=0;
    P=0;
else
    % Controller on
    Fa = bc*vs;
    Sf = bc*vs/T;
    P = Sf^2 *Rw;
end

dqt = vi - pus/mus;
dpus = qt*kt - mus*g-((pus/mus - ps/ms)*bs+qs*ks);
dqs = pus/mus - ps/ms;
dps = (pus/mus - ps/ms)*bs + (qs*ks) - (ms*g) - ((ps/ms - pa/ma)*ba+qa*ka)-Fa;
dqa = ps/ms - pa/ma;
dpa = (ps/ms - pa/ma)*ba +  qa*ka - ma*g +Fa;


dx =[dqt;dpus;dqs;dps;dqa;dpa];
oth = [vi;P;Fa]; % pressure and volume
end
```