Advanced Modeling:
(MAE 271)

Lab#4:
Quarter Car Model

Due Date:
November 22nd, 2019

Course/Section:
MAE 271-001

Term:
Fall 2019

Instructor:
Donald L. Margolis

Author(s):
Ryan Liang
Early C. Cobb III
Rayne Milner

Introduction:

The purpose of this lab was to test a model of automotive vibration control using a voice coil actuator implemented in a quarter car model. The tire of the car is modeled by a spring with direct input from the ground. The suspension is modeled by a parallel spring and damper system. Similar to a realistic application of the skyhook suspension, the vibration control system is modeled by a voice coil actuator with its own set of spring and damper. The impact felt by the passenger is represented by the acceleration of the sprung mass, m_a. The goal of this simulation is to test the required input to dampen the vibrations felt by the passenger using current feed from the actuators.

Schematic Diagram of System:

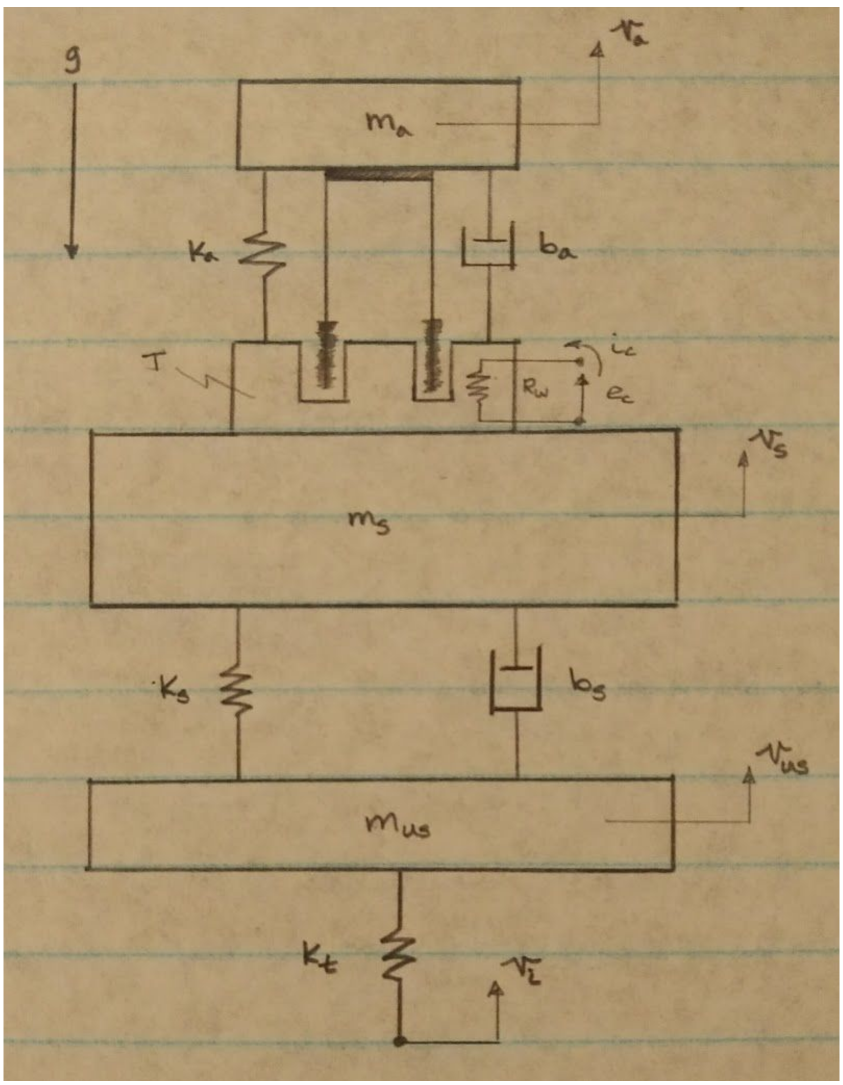


Figure 1 - Schematic Diagram of System

The diagram above represents the quarter car model. Once again, the spring k_t represents the tire's capacitance, while mass m_{us} represents the unsprung mass or the tire. Meanwhile, k_s and b_s represents the sprung mass and sprung damper; along with m_s, the sprung mass, these three components make up the basic parallel suspension system. Since it is not possible to implement a skyhook suspension, since there is nothing we can fix a damper to, we need to implement an actuator at the top of the sprung mass to replicate the effects of a skyhook damper. The actuator we're using is a voice coil that has a coupling constant T, winding resistance R_w, and current input i_c. The actuator mass, m_a, is connected by a spring and damper, k_a and b_a. Gravity is assumed to be perpendicular to the ground, and the acceleration of the sprung mass is of interest as in this model. Thus, we can assume that a_s is the acceleration being felt by a passenger sitting in this car. Note that this is not necessarily true in a real car, where there may be other components altering the impact felt by the passenger, but they will not be modeled in this simulation.

Bond Graph of System:

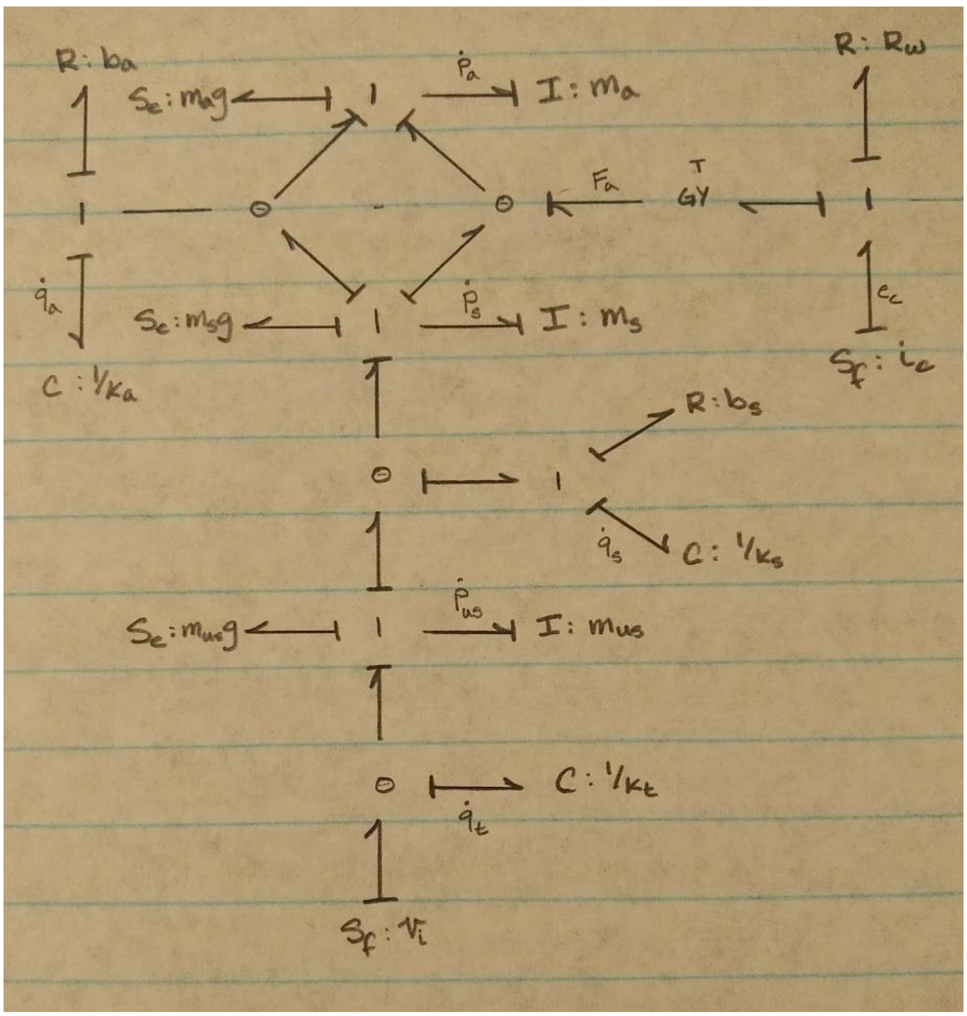
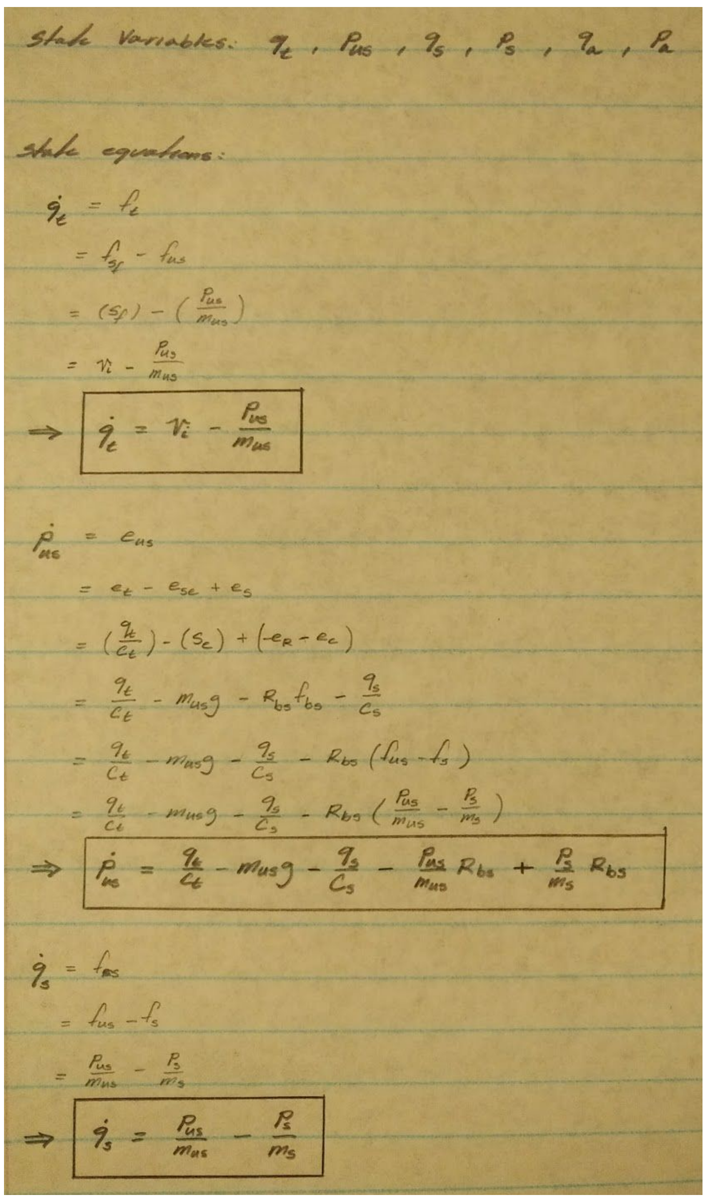


Figure 2 - Bond Graph Diagram of System

The bond graph above represents the quarter car suspension system with a voice coil actuator. The impact from the ground is modeled by the flow input. The energy is initially stored in the capacitance element k_t, and the element m_{us}. The common effort between V_{us} and V_s captures the relative velocity of the suspension system that is represented by the storage element of k_s and dissipation element of b_s. The voice coil is a gyrating component that transforms electrical flow into mechanical energy with a coupling constant of T. The common effort junctions between mass m_s and m_a represents the parallel nature of the spring, damper, and voice coil. All three mass components are subjected to the force of gravity represented by the three effort sources. Using this bond graph, we were able to extract 6 state variables, p_s, p_{us}, p_a, q_s, q_{us}, and q_a.

State Equations From System:



State Variables: q_t, p_{us}, q_s, p_s, q_a, p_a

State equations:

$$\dot{q}_t = f_t = f_{sf} - f_{us} = (S_f) - \left(\frac{p_{us}}{m_{us}}\right) = V_i - \frac{p_{us}}{m_{us}}$$

$$\Rightarrow \boxed{\dot{q}_t = V_i - \frac{p_{us}}{m_{us}}}$$

$$\dot{p}_{us} = e_{us} = e_t - e_{se} + e_s = \left(\frac{q_t}{C_t}\right) - (S_e) + (-e_R - e_C) = \frac{q_t}{C_t} - m_{us}g - R_{bs}f_{bs} - \frac{q_s}{C_s} = \frac{q_t}{C_t} - m_{us}g - \frac{q_s}{C_s} - R_{bs}(f_{us} - f_s) = \frac{q_t}{C_t} - m_{us}g - \frac{q_s}{C_s} - R_{bs}\left(\frac{p_{us}}{m_{us}} - \frac{p_s}{m_s}\right)$$

$$\Rightarrow \boxed{\dot{p}_{us} = \frac{q_t}{C_t} - m_{us}g - \frac{q_s}{C_s} - \frac{p_{us}}{m_{us}}R_{bs} + \frac{p_s}{m_s}R_{bs}}$$

$$\dot{q}_s = f_{bs} = f_{us} - f_s = \frac{p_{us}}{m_{us}} - \frac{p_s}{m_s}$$

$$\Rightarrow \boxed{\dot{q}_s = \frac{p_{us}}{m_{us}} - \frac{p_s}{m_s}}$$

Figure 2a - State Equation Derivation Part 1

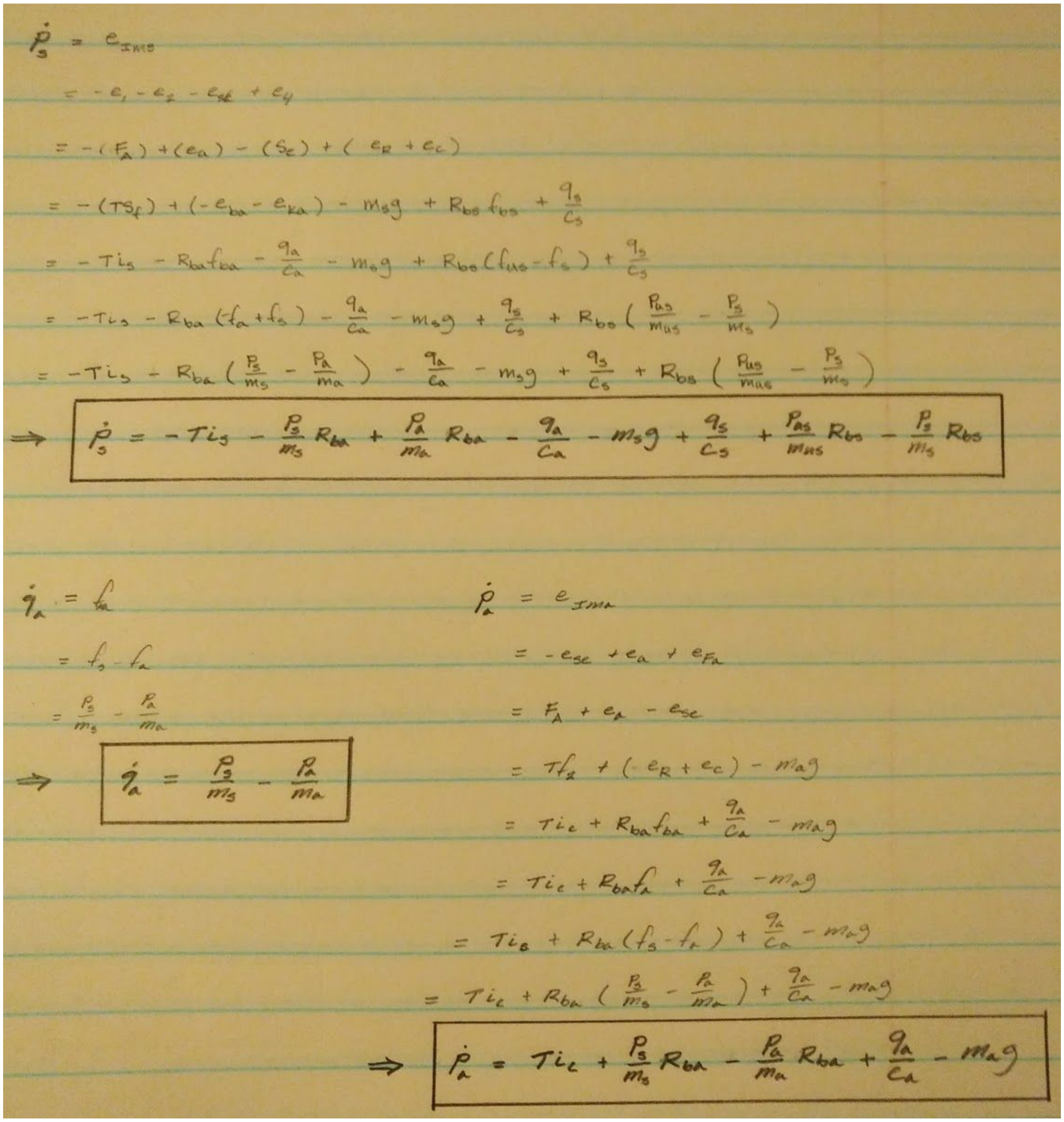


Figure 2b - State Equation Derivation Part 2

Figures 2a and 2b, shown above, display the derivation of the state equations that were derived from the bond graph in Figure 2.

<u>MATLAB Code of System:</u>

For MATLAB code used to model the system, see the appendix attached to the end of this report.

Results & Discussion:

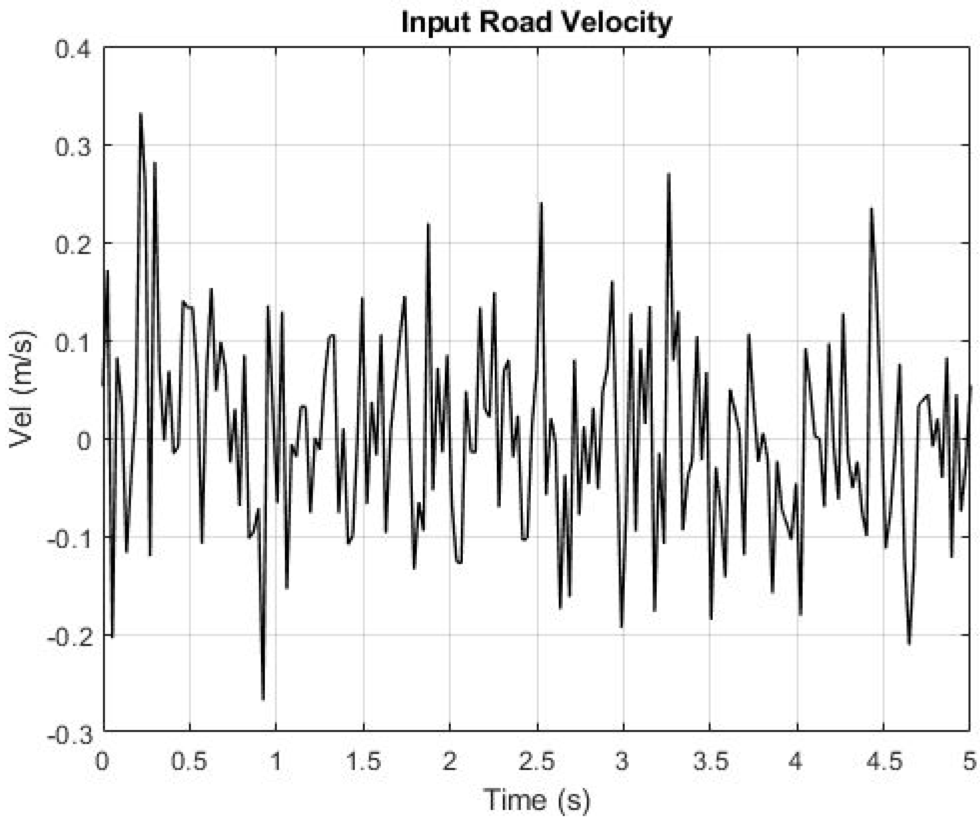


Figure 3 - Input Velocity Graph

The figure shown above displays the random signal generated as the input velocity used in the system.

Nominal Case - No Control Input

With no control input the acceleration of the mass reaches 3 m/s^2 almost ⅓ g. This is quite large, active control may be necessary.

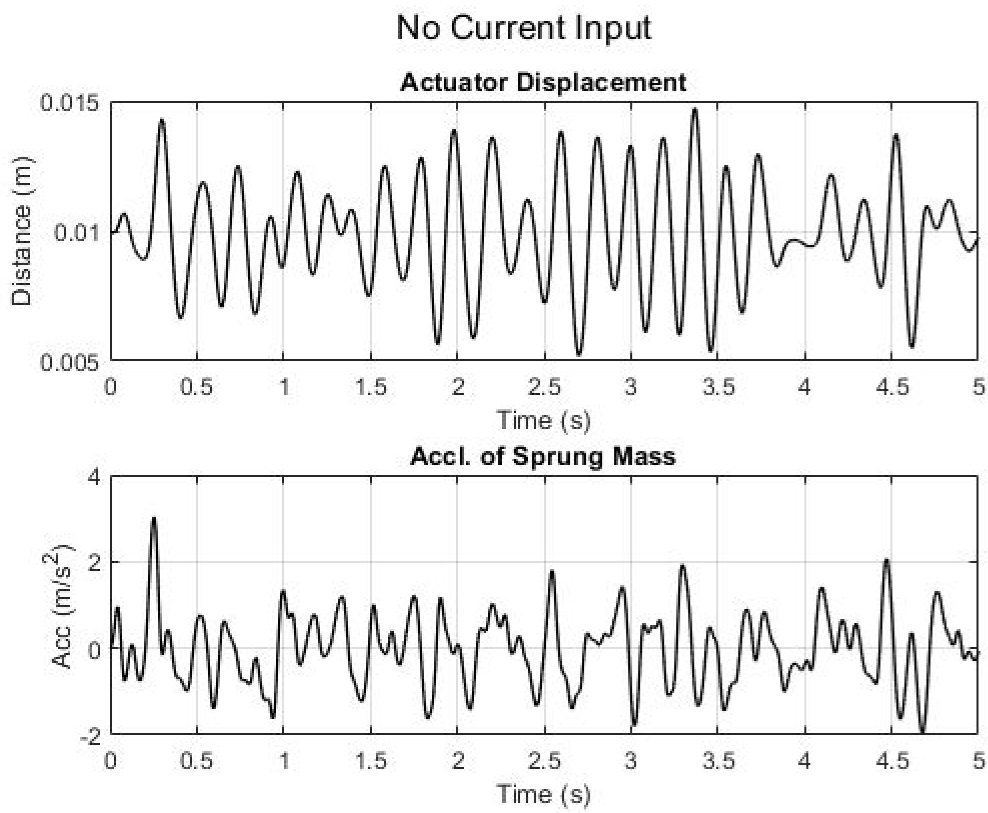


Figure 4 - Actuator Displacement and Mass Velocity w/ No Control Input

Based on the simulation, when the actuator is not active, the sprung mass simply reacts to the input velocity with minimal dampting from the parallel suspension. However, one can see that it is very minimal that it barely dampens the displacement and acceleration from the input. In fact, the quick changing displacement and acceleration represents the violent nature of the oscillation induced by the input velocity. Furthermore, the displacement of the actuator is is roughly between 0.005 to 0.015m, which is not always within our acceptable range. Note that the acceleration of the sprung mass peaked around 3 m/s^2, while fluctuating around -1 to 1 m/s^2.

Nominal Case - Active Control

Active control is enabled with control law given by: $i_c = b_c v_s / T$. This reduces the acceleration significantly but the system draws more than 1kW power in this case. Additionally, the displacement of the actuator reaches 10cm, 2cm above the acceptable range.

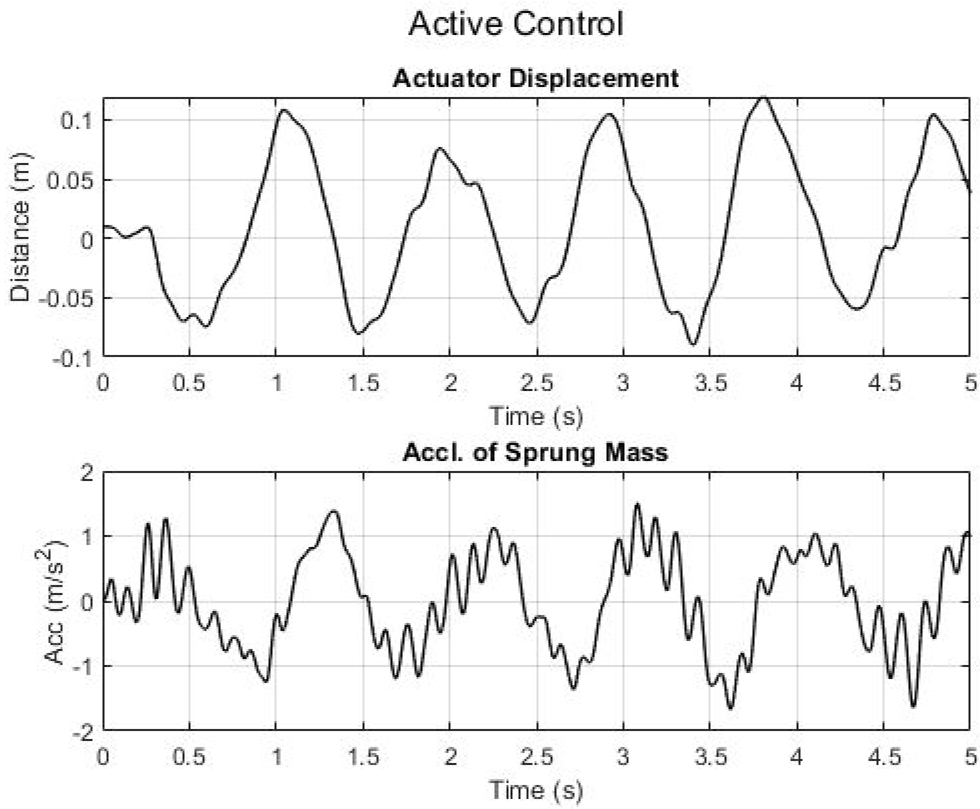


Figure 5 - Actuator Displacement and Mass Velocity for Nominal Case w/ Control

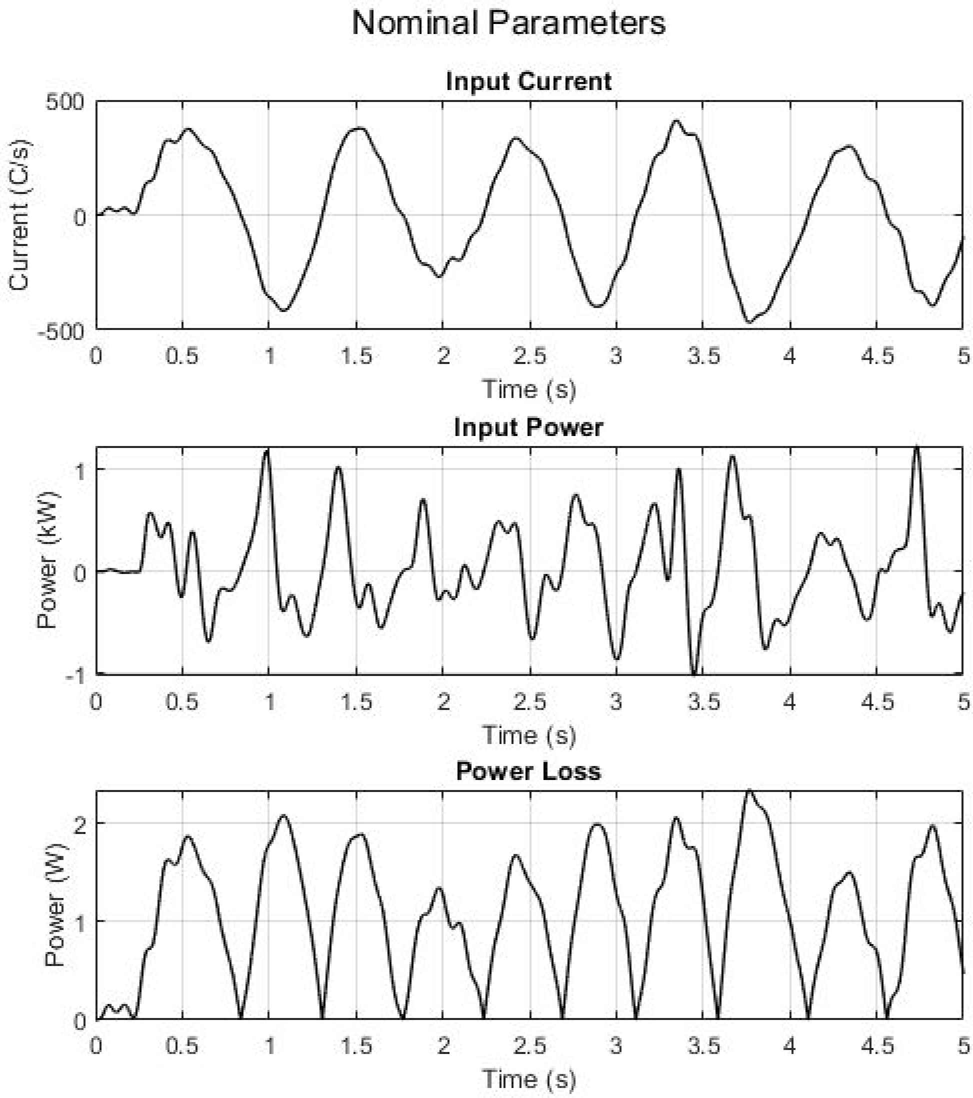


Figure 6 - Input Current, Power and Resistive Power Loss for Nominal Case w/ Control

The difference between simulation with active voice coil actuator compared to zero control case is very clear from figures 4 and 5. The oscillatory behavior of the system's response is significantly tamed with larger time discrepancy between each cycle. The displacement and acceleration takes roughly 1.75s before completing each cycle. Based on this, it is clear that with power input around 1kW as seen in figure 6, the system is capable of significantly reduce the rocky nature of the vibrations felt by a potential passenger. The displacement of the actuator is around -0.09 to 0.12m, which is still outside of our acceptable range. However, the amplitude of acceleration remained fluctuating around -1.2 to 1.2 m/s^2, which is higher than the average fluctuation without active control. This unfortunate fact means that the vibrations are still apparent and will be felt by the passenger. From figure 6, the current input induces the input power as seen above. The power loss of the consequence of the winding resistance of the voice coil, which is modeled above.

Actuator Mass

Changing the mass of the actuator has a significant effect on the total displacement of the actuator and the power input.

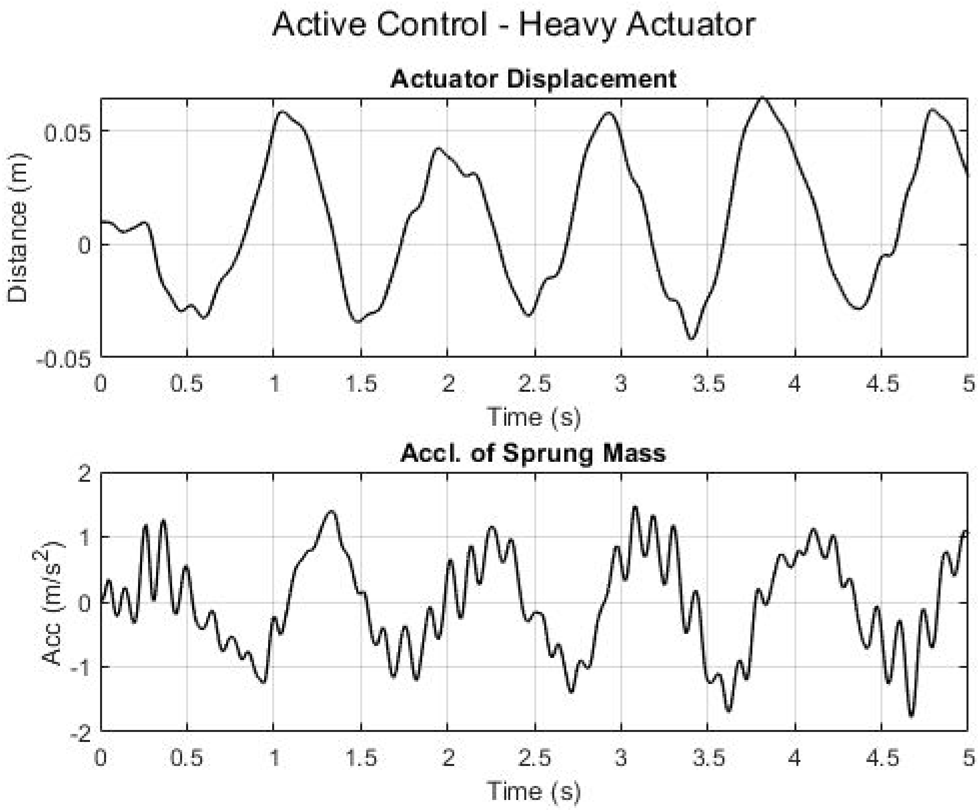


Figure 7 - Actuator Displacement and Mass Velocity for $m_a = 0.04*m_s$

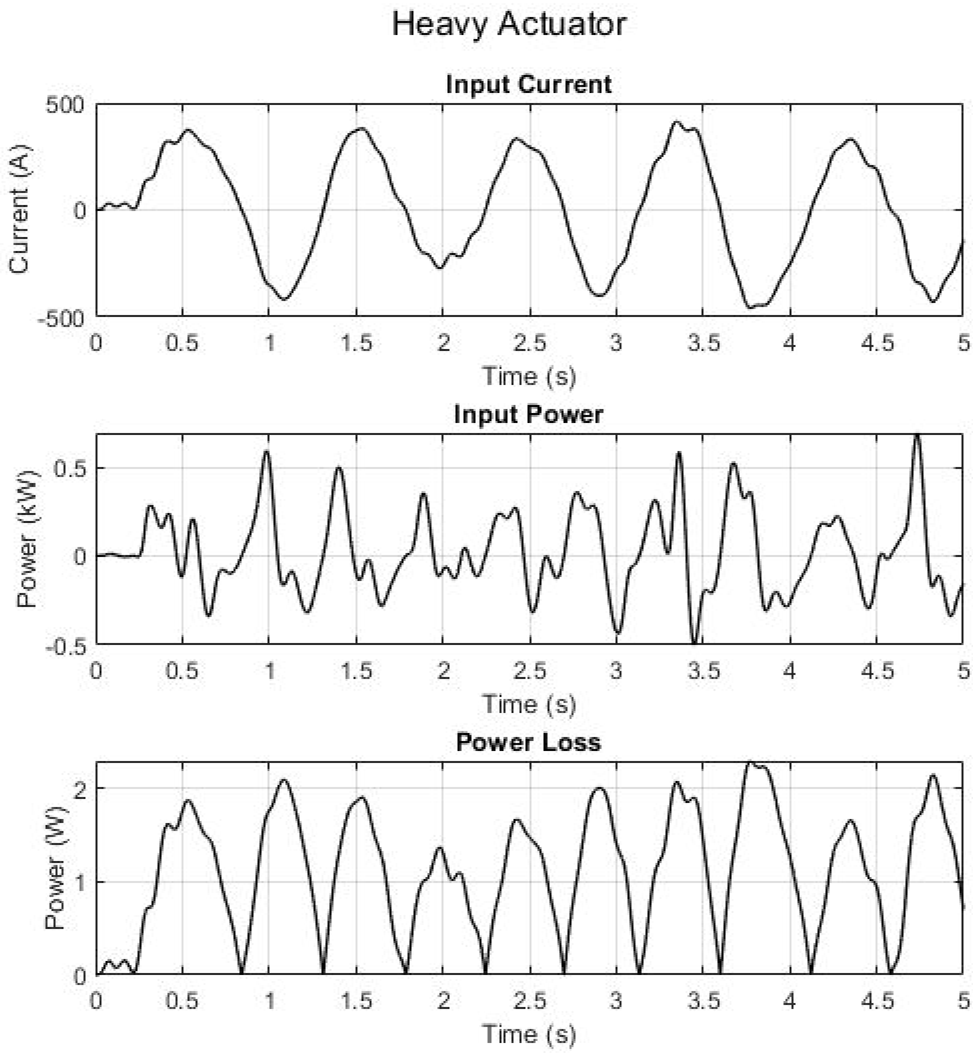


Figure 8 - Input Current, Power and Resistive Power Loss for $m_a = 0.04 * m_s$

By using a heavier actuator, the displacement of the sprung mass is reduced significantly, while the power input is also reduced to half of our original range. Even though the mass of the actuator is only 4% of the sprung mass, the displacement cycles is relatively maintained, while the amplitude fell from 0.1m to 0.05m. Unfortunately, the acceleration of the sprung mass remains relatively similar to the nominal case, which means it did not help the vibration control of the sprung mass. Comparing the power consumption of the heavier actuator case to the nominal case, we can see that the power input required is reduced by half even though the current input remains the same, which is a good sign. The power loss of the system remains relatively consistent to the nominal case.

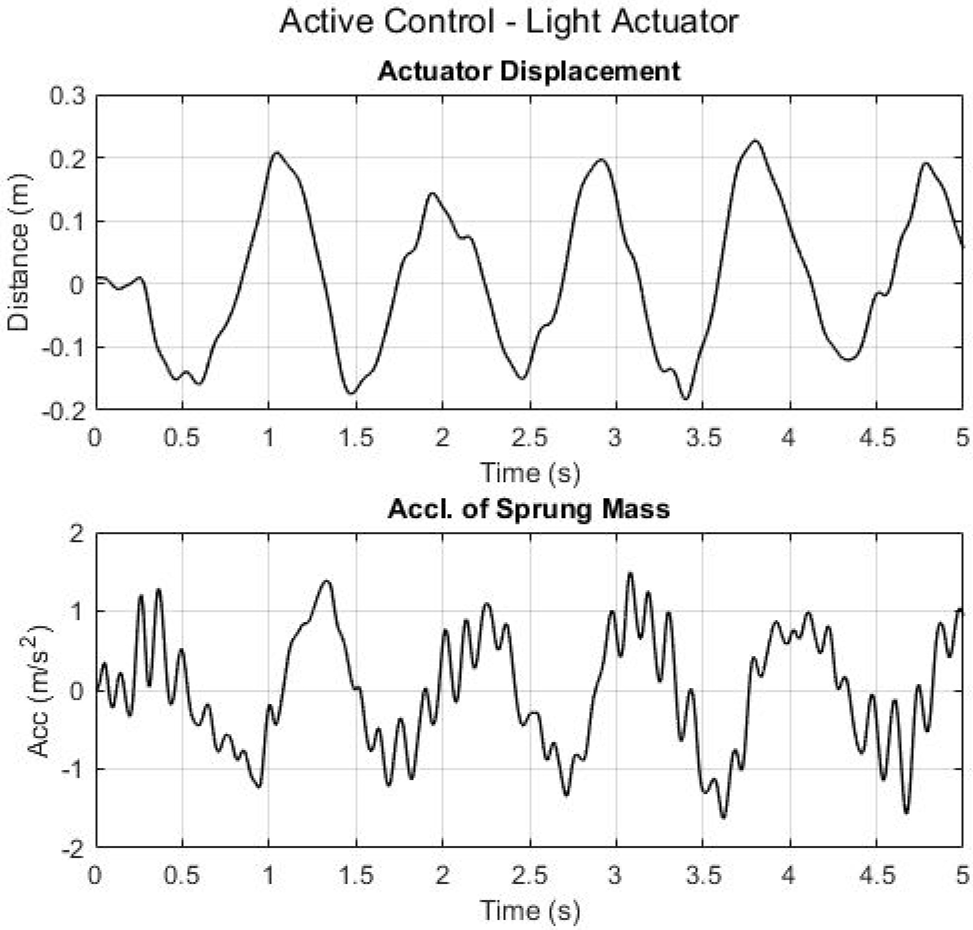


Figure 9 - Actuator Displacement and Mass Velocity for $m_a = 0.01*m_s$

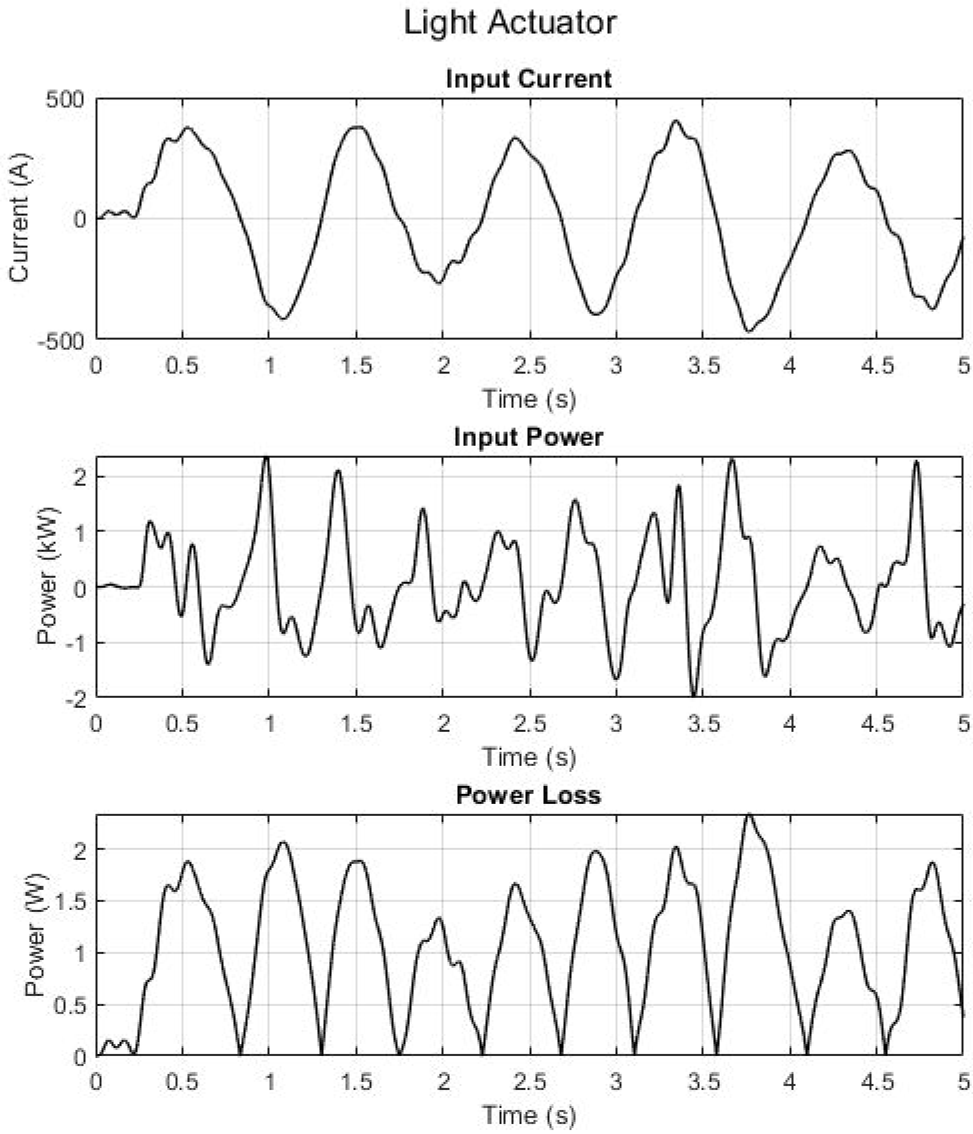


Figure 10 - Input Current, Power and Resistive Power Loss for $m_a = 0.01*m_s$

The light actuator case is 1% of the mass of the sprung mass. To no surprise, by inverting expectations of the results from the heavier case, we did in fact observe higher displacement of the actuator and higher power requirements. In fact, most of the amplitudes appeared to have doubled except for sprung mass acceleration and power loss. Both displacement and input power are out of range, meaning the lighter actuator case is not viable at all.

Spring Stiffness

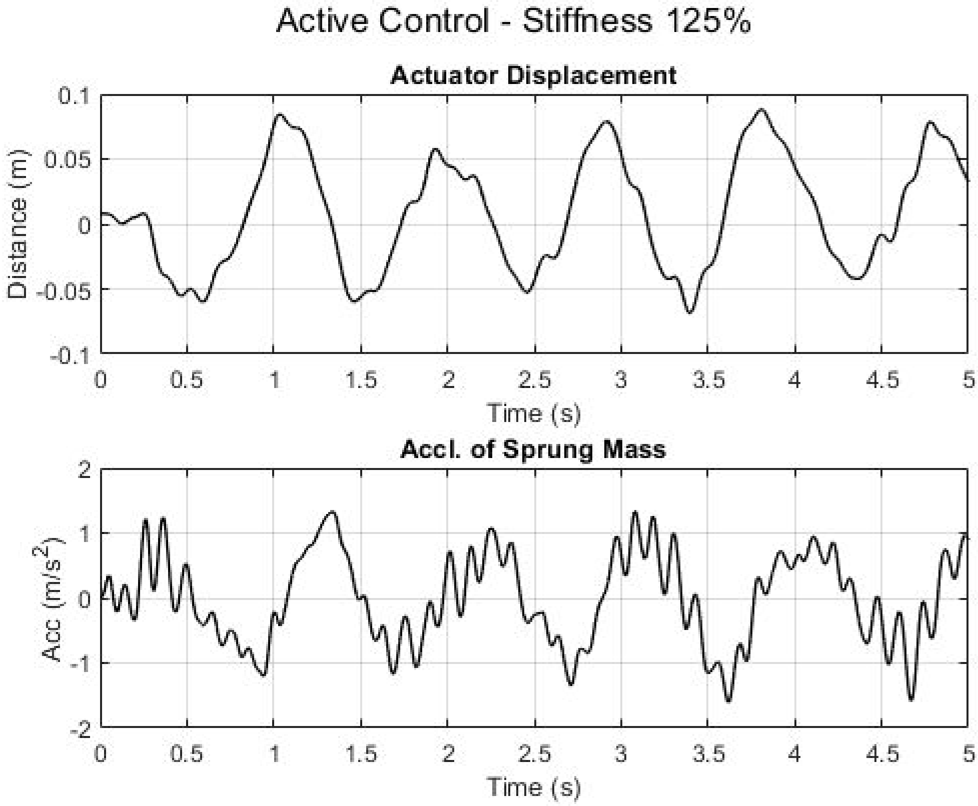


Figure 11 - Actuator Displacement and Mass Velocity for $k_a = 1.25*k_a$

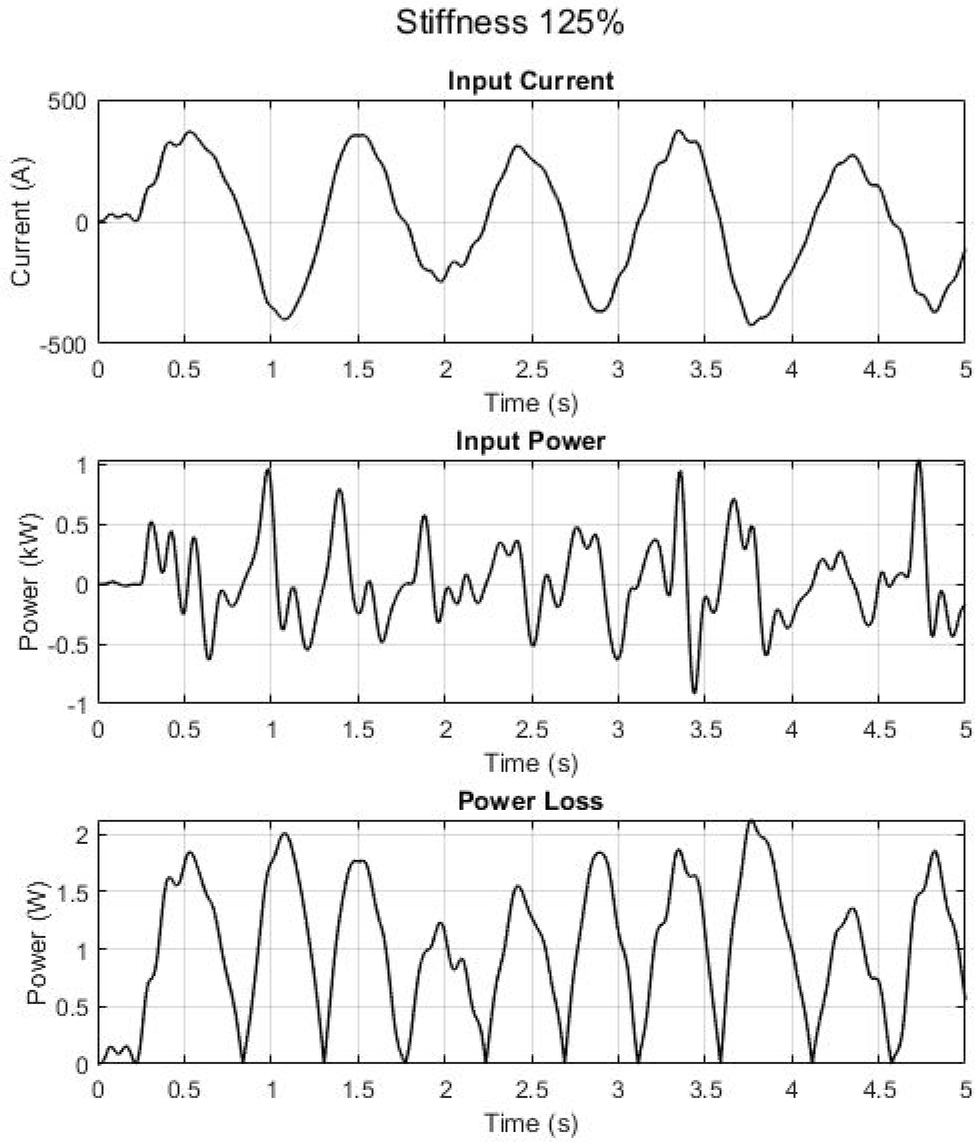


Figure 12 - Input Current, Power and Resistive Power Loss for $k_a = 1.25*k_a$

The stiffness of the actuator spring is increased by 25% from its original stiffness. This case reacted unsurprisingly, in which the actuator displacement had decreased below the acceptable range. This is expected as a stronger spring can store more energy. The acceleration of the sprung mass was unchanged compared to the nominal case. Although the input current remained the same, the power required was reduced very slightly such that it is under 1kW. The power loss has also reduced by a very minimal amount. Based on these behaviors it is recommended to strengthen the spring.

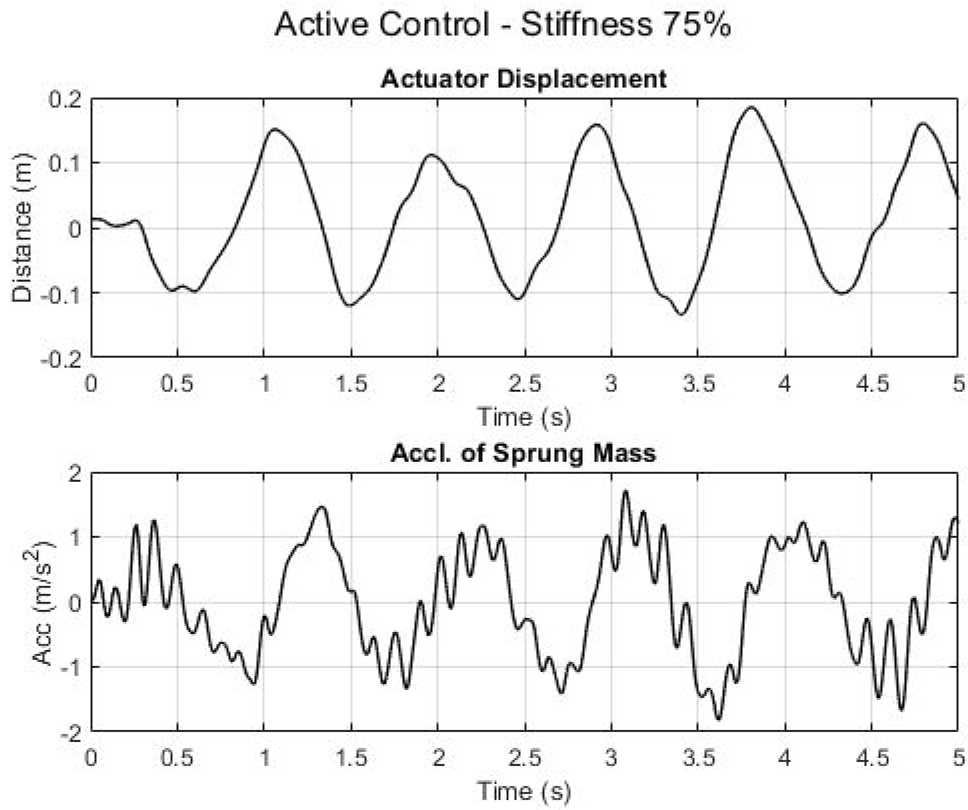


Figure 13 - Actuator Displacement and Mass Velocity for $k_a = 0.75*k_a$

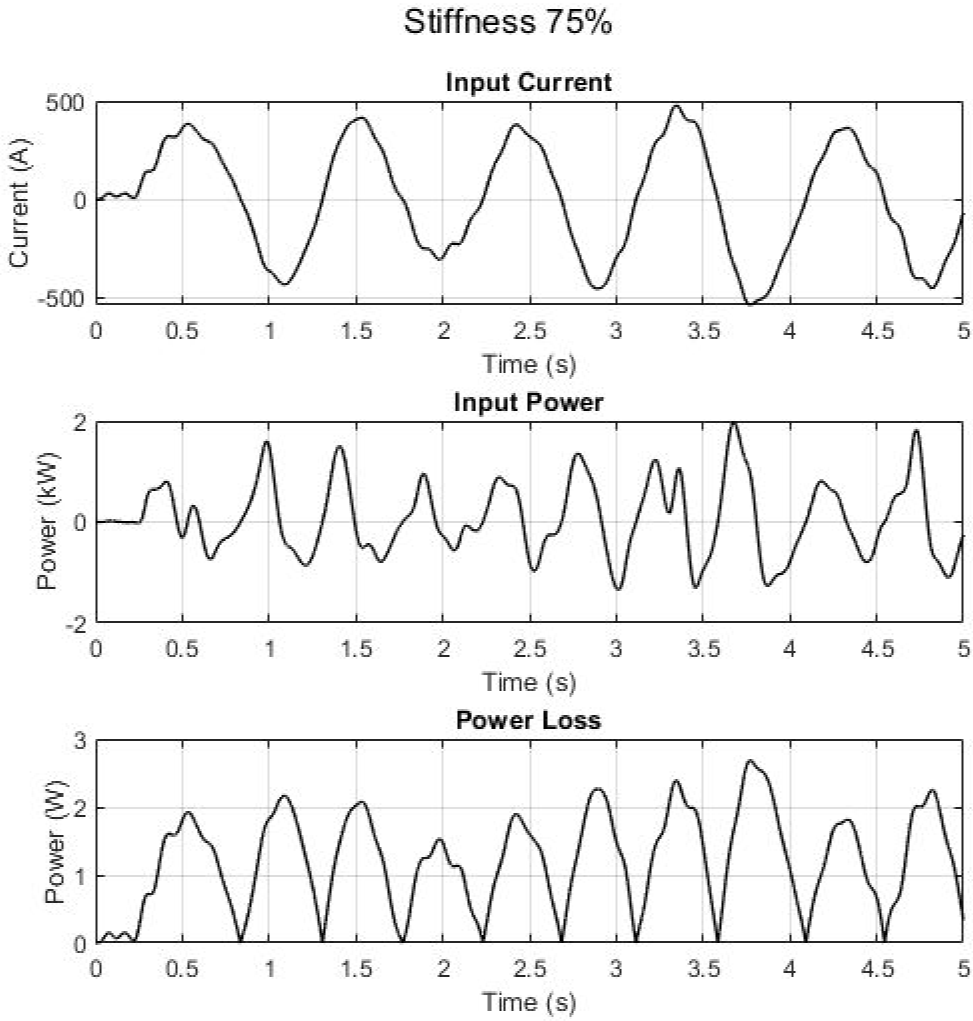


Figure 14 - Input Current, Power and Resistive Power Loss for $k_a = 0.75*k_a$

The stiffness of the actuator spring is decreased by 25% from its original stiffness. This case also reacted unsurprisingly, in which the actuator displacement had increased below the acceptable range. This is expected since a weaker spring stores less energy. The acceleration of the sprung mass was unchanged compared to the nominal case. Although the input current remained the same, the power required was increased to almost double, which is over 1kW. The power loss in this case was also increased slightly. Based on these behaviors, it is not recommended to weaken the spring.

Winding Resistance

Winding resistance only has an effect on the power loss in the resistor. We see that the power loss in the resistor has a maximum of 4W for the nominal case. This is insignificant compared to the total power requirements of the system.

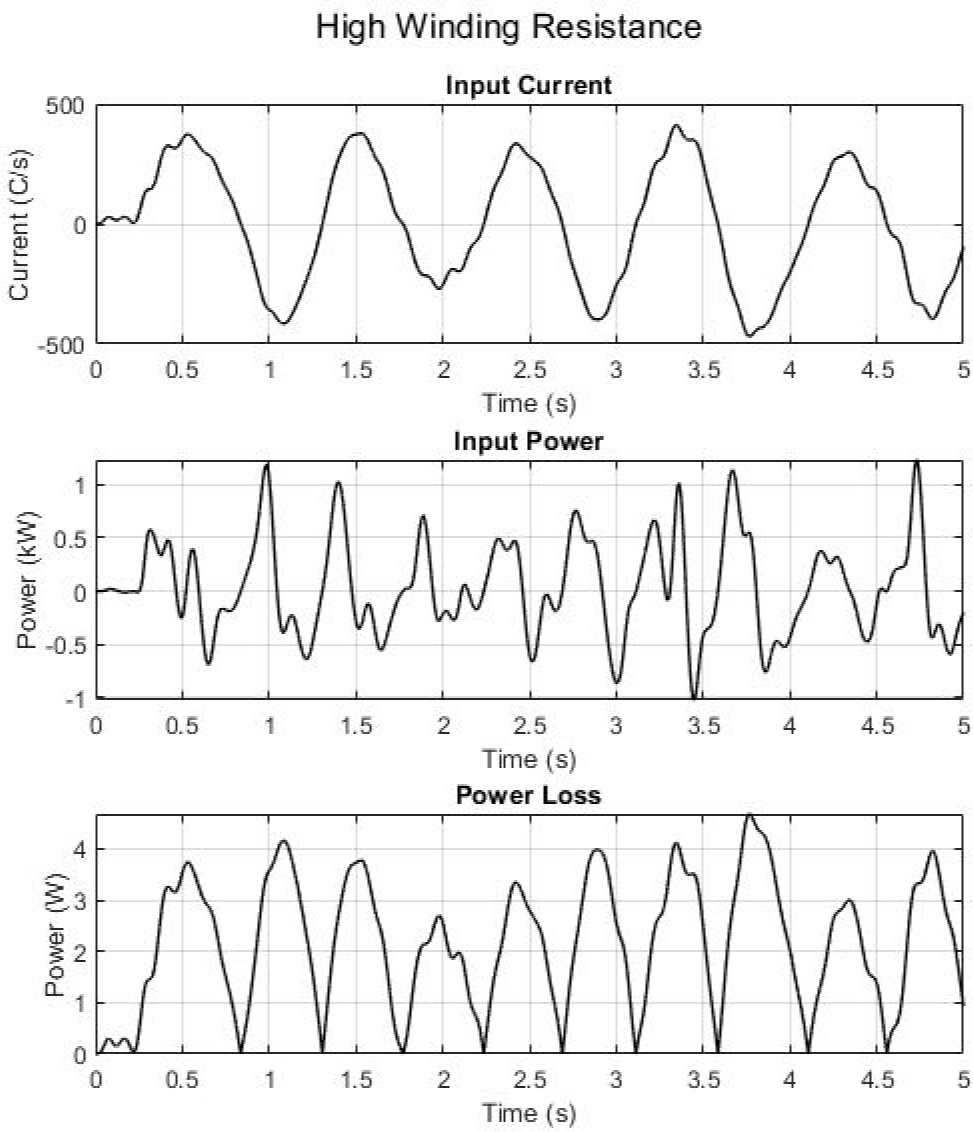


Figure 15 - Input Current, Power and Resistive Power Loss for $R_w = 0.01$□

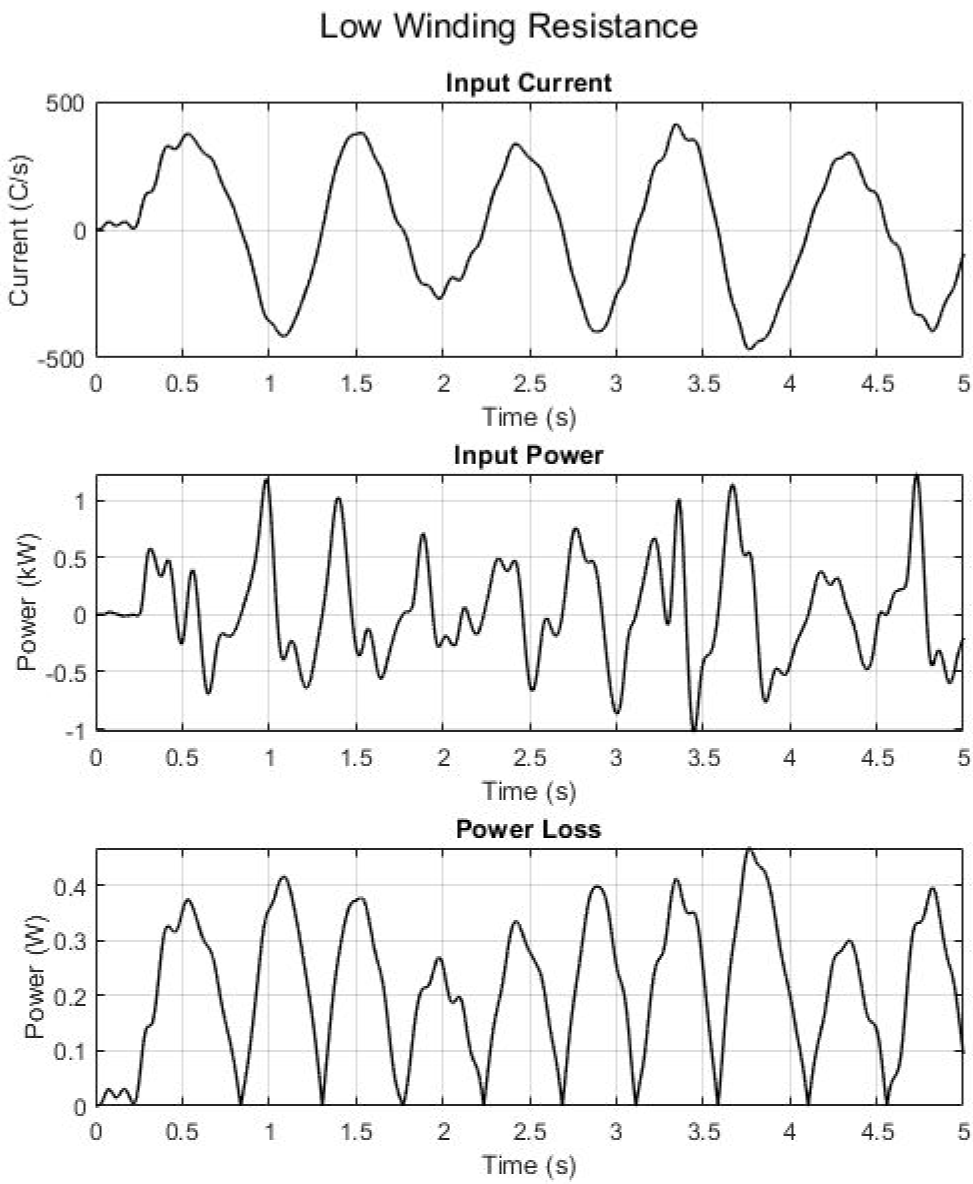


Figure 16 - Input Current, Power and Resistive Power Loss for R_w = 0.001Ω

The difference between 0.01 and 0.001Ω in power loss is by one whole order of magnitude. It was unsurprising that higher winding resistance would increase the power loss, and vice versa with a lower winding resistance. The doubling of the original winding resistance from 0.005 to 0.01Ω had double the effect on power loss, while cutting the resistance in half at 0.001Ω the power loss was 10 times less than the 0.01Ω case.

Motor Coupling

Changing the strength of the motor coupling only has an effect on the amount of current required to actuate the mass. This also increases the power loss.

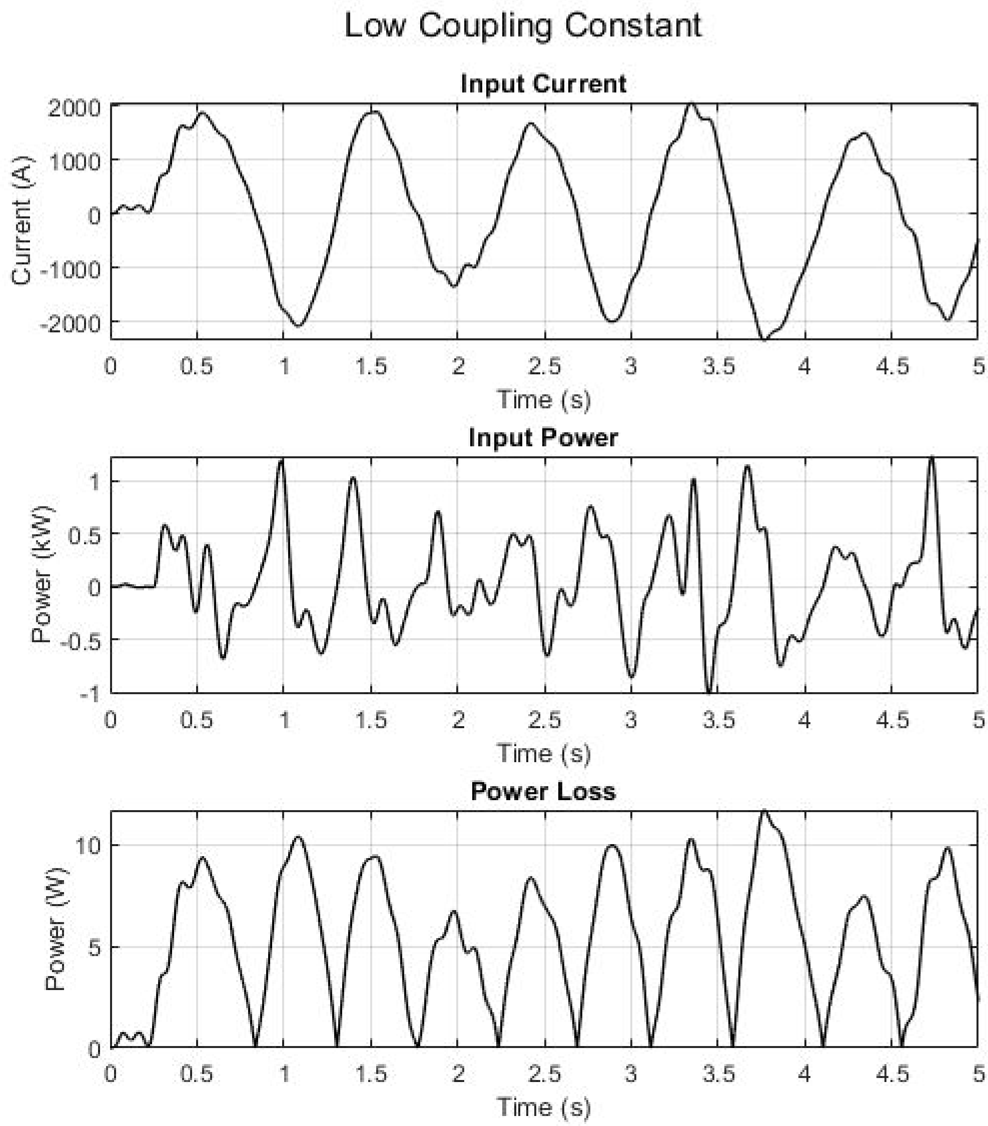


Figure 17 - Input Current, Power and Resistive Power Loss for T = 1.0 Nm/A

When we have a low coupling constant, the transformation from electrical energy to mechanical energy becomes worse. Thus, it was expected that we would see higher current requirement to operate the actuator at the same power input. It was in fact 4 times greater than the nominal case at T = 1.0 Nm/A, and 5 times greater in power loss.

Conclusion

Active control is an effective way to reduce the acceleration of the sprung mass. Under the nominal conditions, however, the system does not meet the requirements on actuator displacement or power consumption. By either increasing the mass of the actuator or the stiffness of the actuator spring, we can bring the displacement and power to acceptable maximum values. Additionally, the increased cost of winding resistance does not seem to be worth it for the negligible increased efficiency.

Lab 3 Master File

Table of Contents

The purpose of the master file is to enter system parameters, call the differential equation solver, and finally post-process and plot results.

Clean Up Workspace

```
clear all
close all
clc
```

Input System Parameters

```
global g m_us m_s k_s b_s k_t T m_a k_a b_a U X_i slope_i b_c

g=9.8;
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=.1; 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 CAN BE IN RANGE 0.001< R <0.01
T=1; %Nm/A Coupling constant CAN BE IN RANGE 1< T <10
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; %m/s Trial vehicle velocity

%Flow source
rng('default');
delta_x=.5; Length=500; %This defines a length of a road in m
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 as the road slope at each delta_x
slope_i=.005*(slope_raw-mean(slope_raw)); %This is the slope
%vector that has any average value removed. This makes the road zero
 mean slope.
%The scaling number at the front makes the passive vehicle
%without control have a sprung mass acceleration that is
%reasonable. I experimented to determine this number.
```

Define initial Conditions

The initial condition values for each system state variable are defined.

```
qto= (m_s+m_a+m_us)*g/k_t; %m, tire flex (disp)
puso=0;%N-s, un spring momentum
pso=0; %N-s, sprung mass displacement
qso=(m_s +m_a)*g/k_s; %m, suspension displacement
qao= m_a*g/k_a; %m, actuator displacement
pao=0; %N-s, car body momentum

initial = [qto;puso;pso;qso;qao;pao];
% [tire flex, un sprung momentum, sprung mass displacement, suspension
 displacement,actuator disp, car body momentum]
```

Setup Time Array

The timestep was given in problem specification

```
tspan = 0:0.0001:5;
```

Call ode45() Ordinary Differential Equation Solver

```
[t,s]=ode45(@Lab4_fxn,tspan,initial); %call ode45

qt=s(:,1); %m, tire flex (disp)
pus=s(:,2); %N-s, un spring momentum
ps=s(:,3); %N-s, sprung mass displacement
qs=s(:,4); %m, suspension displacement
qa=s(:,5); %m, actuator disp
pa=s(:,6); %N-s, car body momentum

% obtain derivatives and additional outputs
for i=1:length(t)
    [ds(i,:), ext(i,:)] = Lab4_fxn(t(i),s(i,:));
end

dqt=ds(:,1); %m, tire flex (disp)
dpus=ds(:,2); %N-s, un spring momentum
dps=ds(:,3); %N-s, sprung mass displacement
dqs=ds(:,4); %m, suspension displacement
```

```
dqa=ds(:,5); %m, actuator disp
dpa=ds(:,6); %N-s, car body momentum

Vin = ext(:,1); %input velocity
ic = ext(:,2); %input current

Psacc = dps/m_s; %acceleration of the spring mass
Powerinput = R_w*ic.*sign(ic) + ic.*(pa*(T/m_a) - ps*(T/m_s)); %input
 power of the actuator
Powerloss = R_w*ic.*sign(ic);
```

Outputs

```
figure('Name','Actuator Disp','NumberTitle','off','Color','white')
 subplot(2,1,1)
 plot(tspan, qa,'k','LineWidth',1);grid on;
 title('Actuator Displacement')
 ylabel('Distance (m)')
 xlabel('Time (s)')
 subplot(2,1,2)
 plot(tspan, Psacc, 'k','LineWidth',1);grid on;
 title('Accl. of Sprung Mass')
 ylabel('Acc (m/s^2)')
 xlabel('Time (s)')
 sgtitle('Active Control - High Coupling Constant')

 figure('Name','Power Input','NumberTitle','off','Color','white')
 subplot(3,1,1)
 plot(tspan, ic,'k','LineWidth',1);grid on;
 title('Input Current')
 ylabel('Current (A)')
 xlabel('Time (s)')
 subplot(3,1,2)
 plot(tspan, Powerinput/1000, 'k','LineWidth',1);grid on;
 title('Input Power')
 ylabel('Power (kW)')
 xlabel('Time (s)')
 subplot(3,1,3)
 plot(tspan, Powerloss, 'k','LineWidth',1);grid on;
 title('Power Loss')
 ylabel('Power (W)')
 xlabel('Time (s)')
 sgtitle('Low Coupling Constant')
```

Published with MATLAB® R2019a

Table of Contents


```
function [ds, ext] = Lab4_fxn(t,s)
```

input system parameters

```
global g m_us m_s k_s b_s k_t T m_a k_a b_a U X_i slope_i b_c
```

rename state variables

```
qt=s(1); %m, tire flex (disp)
pus=s(2);%N-s, un sprung momentum
ps=s(3); %N-s, sprung mass displacement
qs=s(4); %m, suspension displacement
qa =s(5);
pa=s(6); %N-s, car body momentum
```

input specifications

Here we define the inputs into our system. First we calculate our effort sources. Then we calculate the velocity profile of the triangular bump.

```
%Flow source
X = U*t; %pos of vehicle
slope = interp1(X_i,slope_i,X);
Vi = U*slope;
```

equations of motion

These are the dynamic equations describing our state variables given in the form s'=f(s,inputs).

```
ic = (b_c/T)*(ps/m_s);
dqt = Vi - pus/m_us;
dpus = qt*k_t - m_us*g - qs*k_s - b_s*(pus/m_us - ps/m_s);
dps = qs*k_s + b_s*(pus/m_us - ps/m_s) - m_s*g- qa*k_a - b_a*(ps/m_s -
 pa/m_a) -T*ic;
dqs = pus/m_us - ps/m_s;
dqa = ps/m_s - pa/m_a;
dpa = qa*k_a + b_a*(ps/m_s - pa/m_a) + T*ic - m_a*g;
```

stacking up the derivatives for output as a vector

```
ds=[dqt; dpus; dps; dqs; dqa; dpa];
ext(1)= Vi;  % displacement of input force
ext(2) = ic; %current input
```

Published with MATLAB® R2019a