To: D. W. Mueller, Jr., Ph.D., P.E.

From: Mason Averill

Subject: Special Problem 3

Date: 5 October 2020

PURDUE UNIVERSITY FORT WAYNE

Purpose:

The purpose of this memo is to communicate the methodology and results of Special Problem 3 for ME-544: Modeling and Simulation of Mechanical Engineering Systems, completed October 5th 2020.

Purpose and Scope of Assignment:

The purpose of this assignment was to determine the pitch and bounce about the center of gravity of an automobile traveling down a rough road. Information regarding a model describing the geometry of the road was supplied, as well as the mass of the automobile, the front and rear suspension equivalent spring constants, the mass moment of inertia of the automobile about its center of gravity, and the distances to the front and rear suspension systems with reference to the center of gravity.

Mathematical Modeling of Problem:

This system was modeled as a lumped mass with inputs from the front and rear suspension system. An equation of motion was derived for the displacement about the center of gravity in the transverse direction relative to the direction of motion for the automobile (bounce), as well as the angular displacement of the automobile about its center of gravity. The derivation of these two equations of motion and the final results can be seen in Figures 1 and 2.

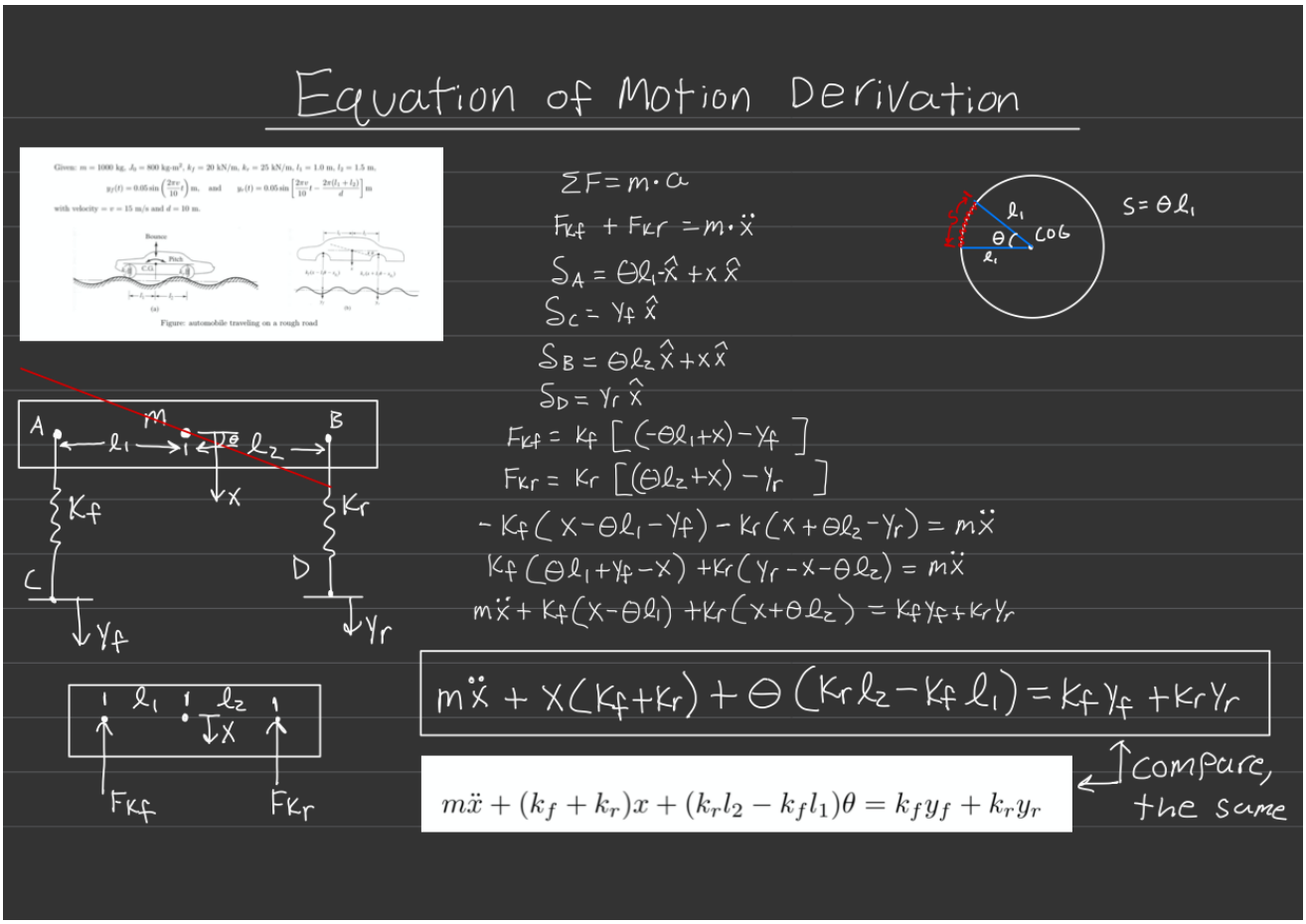

Figure 1: Vertical Displacement (Bounce) Equation of Motion Derivation

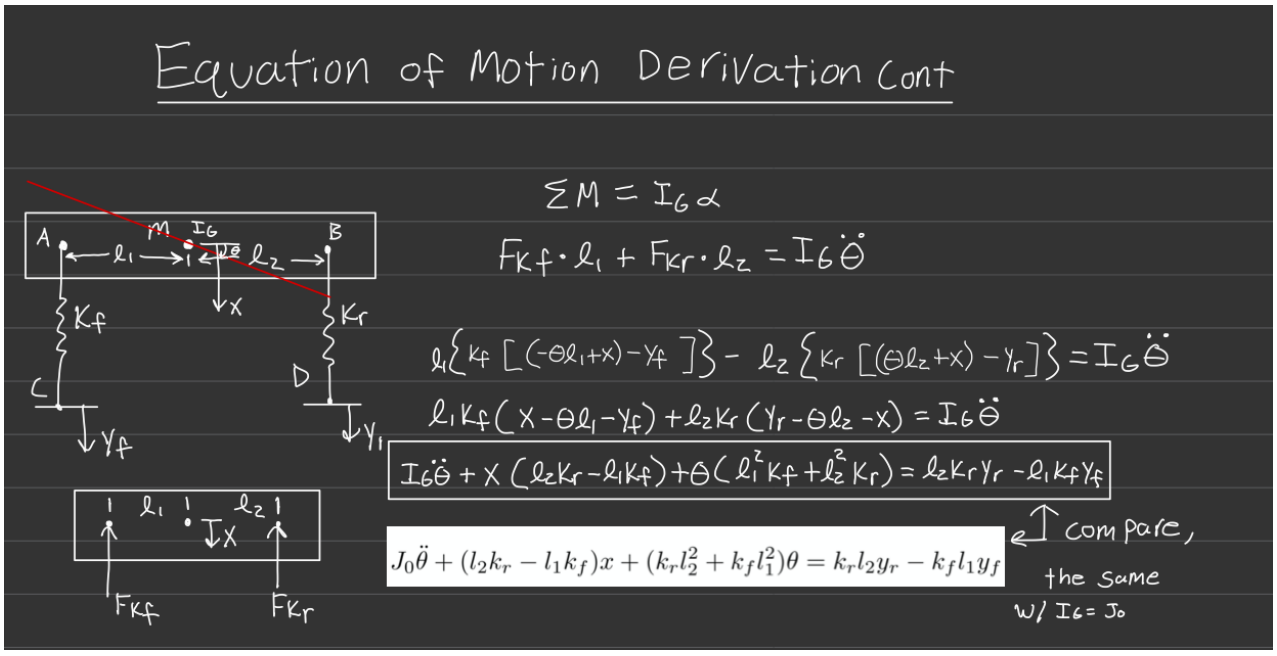

Figure 2: Angular Displacement (Pitch) Equation of Motion Derivation

Development of Simulation:

In order to solve the second order, non-homogenous, ordinary differential equations of motion simultaneously, a state space model of the system was developed, as shown in Figures 3 and 4.

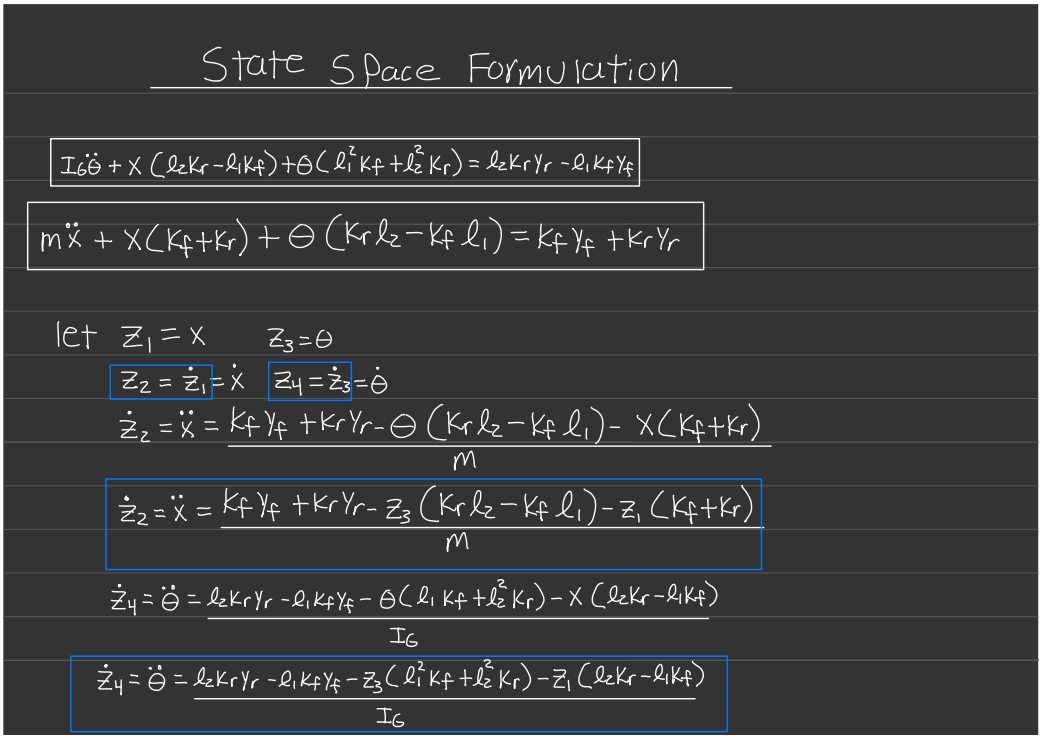

Figure 3: State Space Formulation

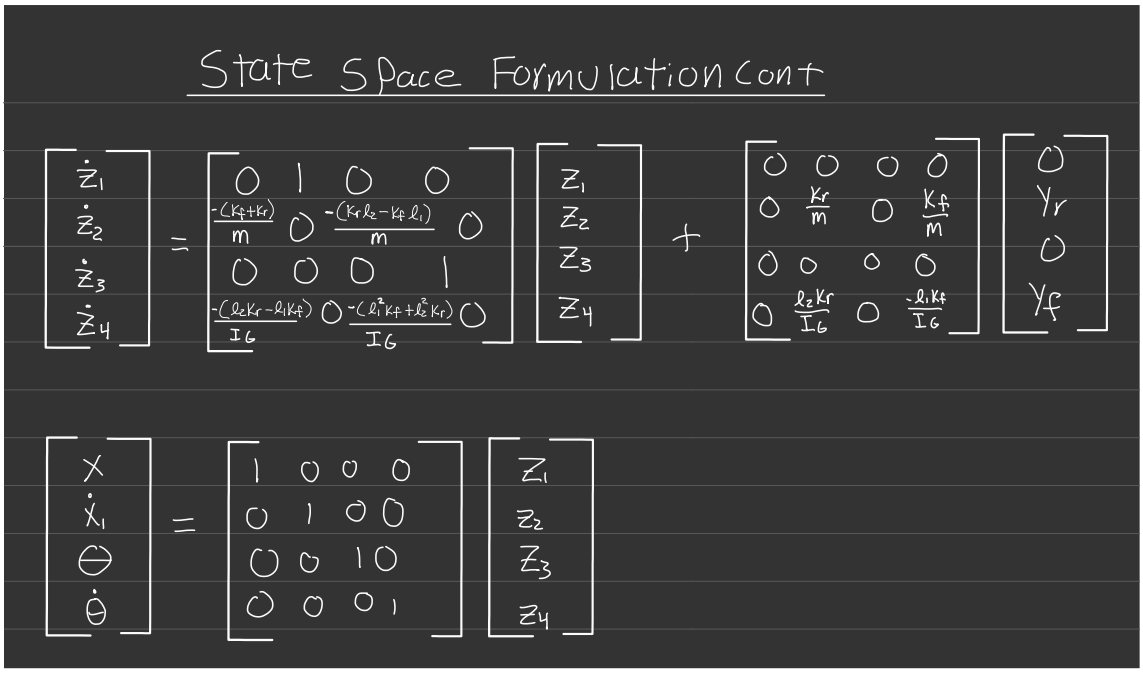

Figure 4: State Space Formulation Continued

With the State Space Formulation completed, a MATLAB script was written to solve the equations of motion simultaneously.

Verification/Validation of Model:

Figure 5 serves as a baseline to which other Figures will be compared to after modifications to input parameters.

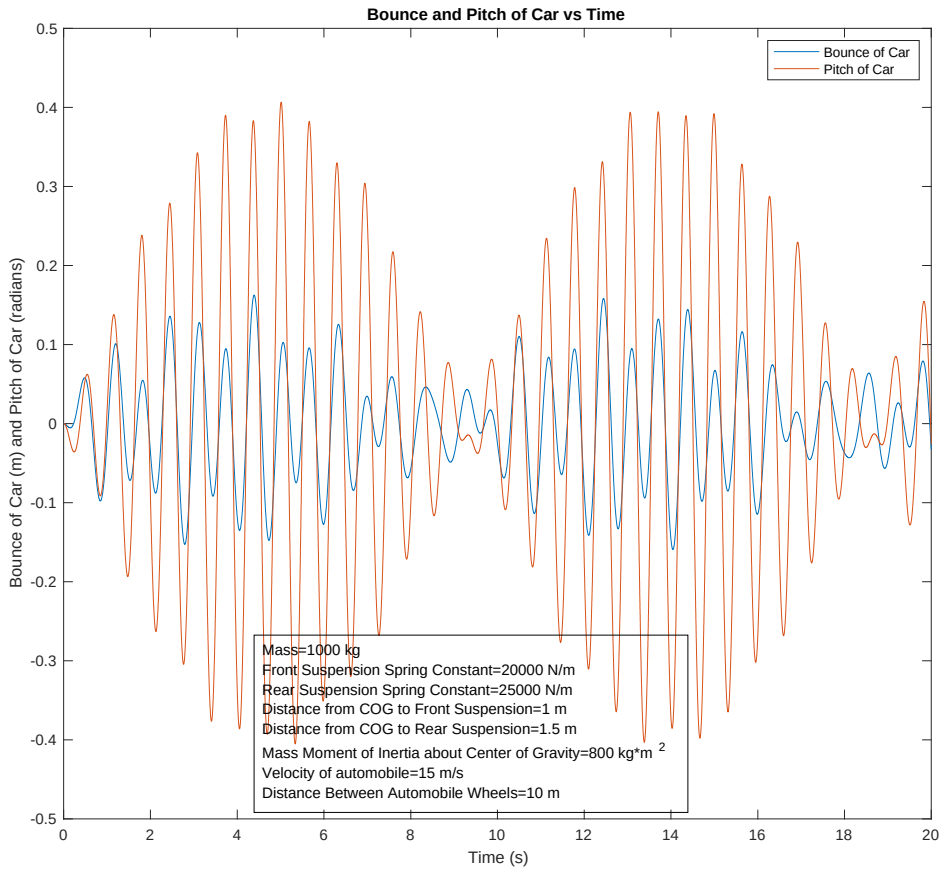

Figure 5: Baseline Output

Figure 6 shows the results of increasing the mass moment of inertia about the center of gravity by a factor of 10. As would be expected, the magnitude of the variation in the angular position decreases drastically.

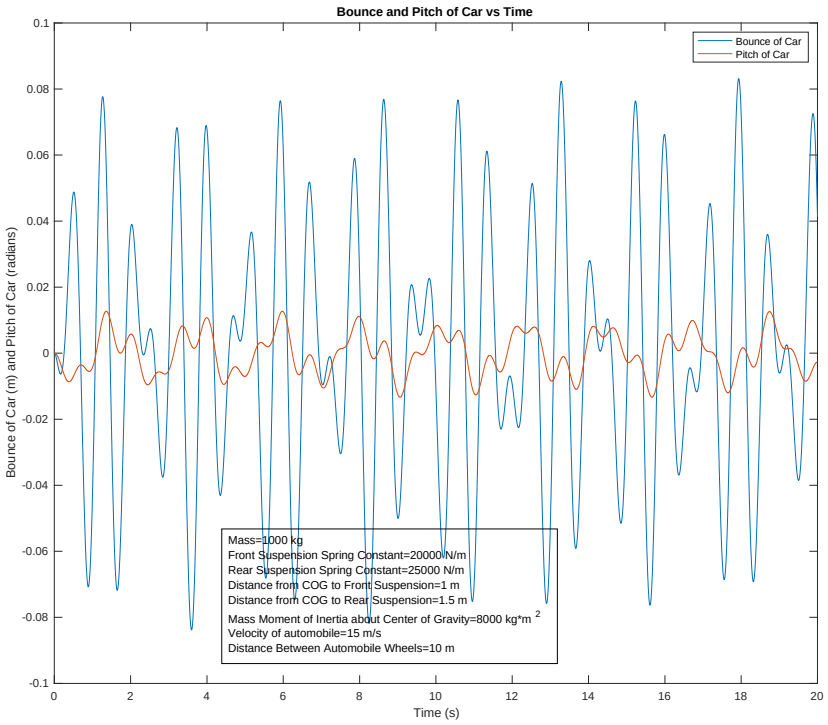

Figure 6: Mass Moment of Inertia About COG Increased by a Factor of 10

Figure 7 shows the results of increasing the mass of the automobile by a factor of 10. As would be expected, the magnitude of the variation in position decreases drastically.

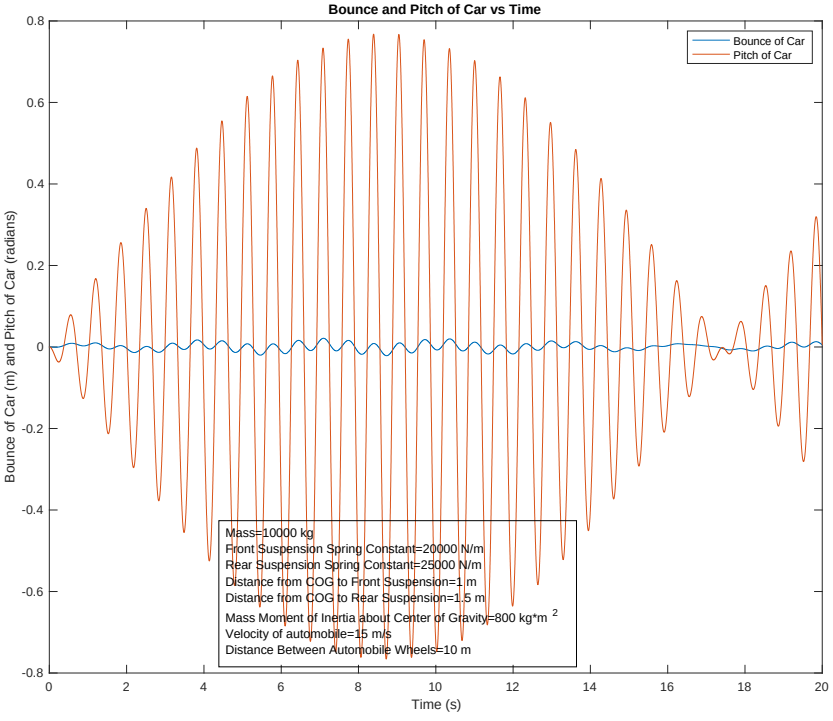

Figure 7: Mass Increased by a Factor of 10

Figure 8 shows the results of decreasing the road roughness by a factor of 5. As would be expected the magnitude of the variation in position and angular position decreases drastically.

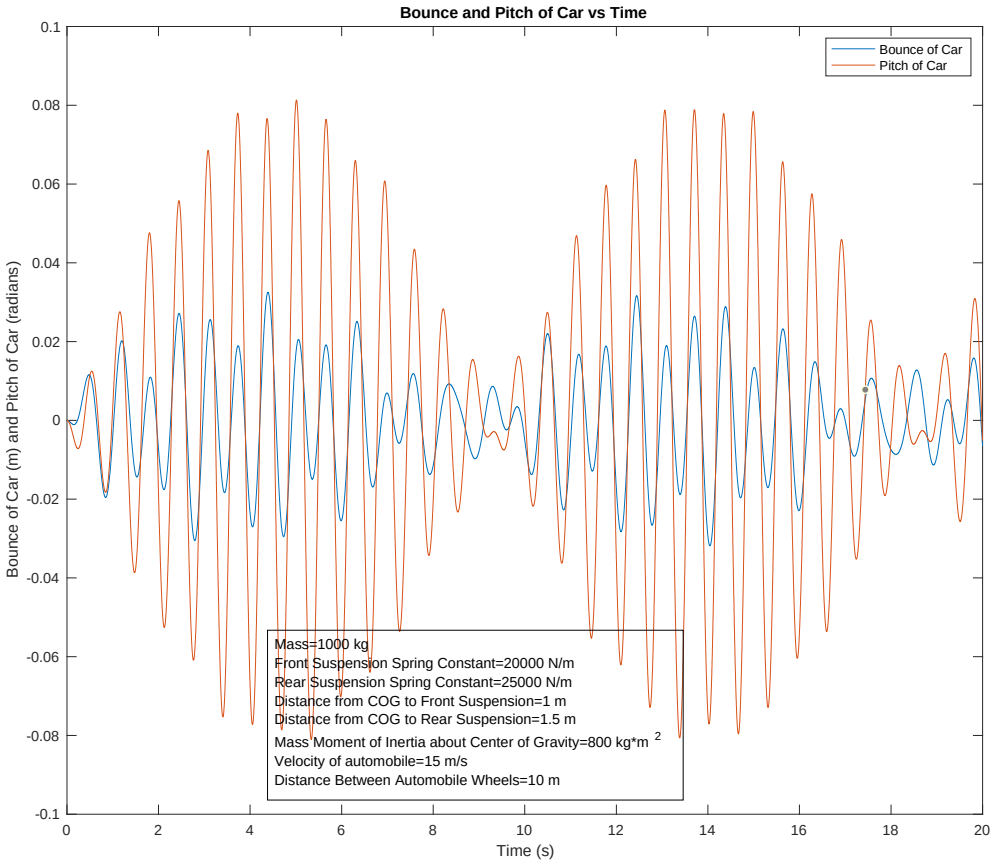

Figure 8: Road Roughness Decreased by a Factor of 5

Execution of Simulation to Determine the Bounce and Pitch of the Automobile vs Time

Figure 5 and Figures 9-11 show the results of running the simulation with the prescribed input parameters.

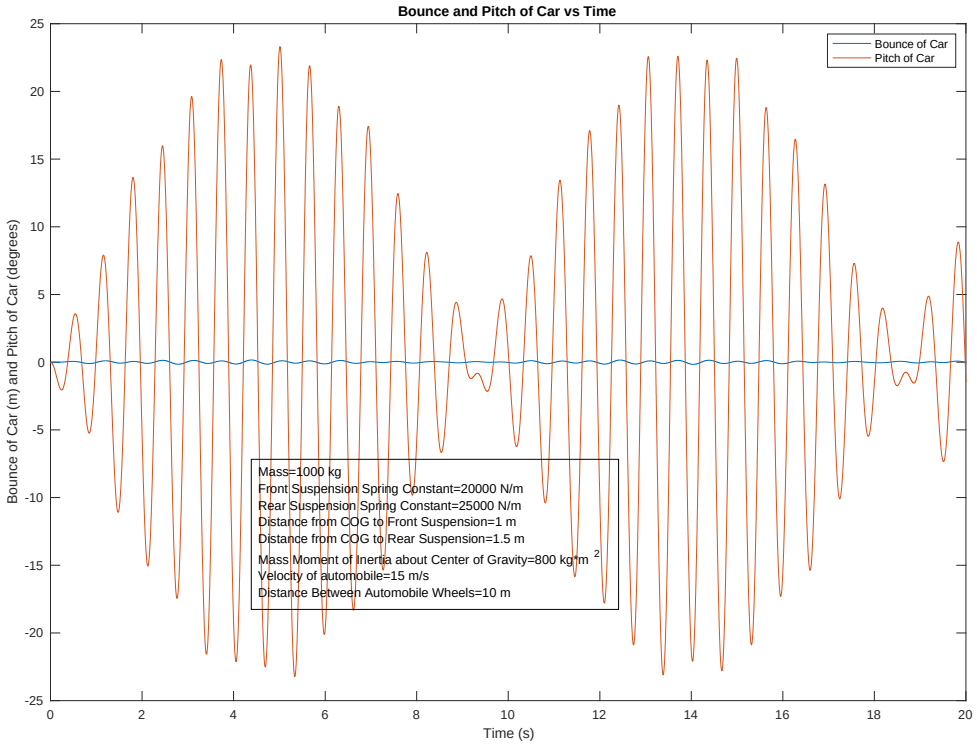

Figure 9: Bounce and Pitch of Car vs Time

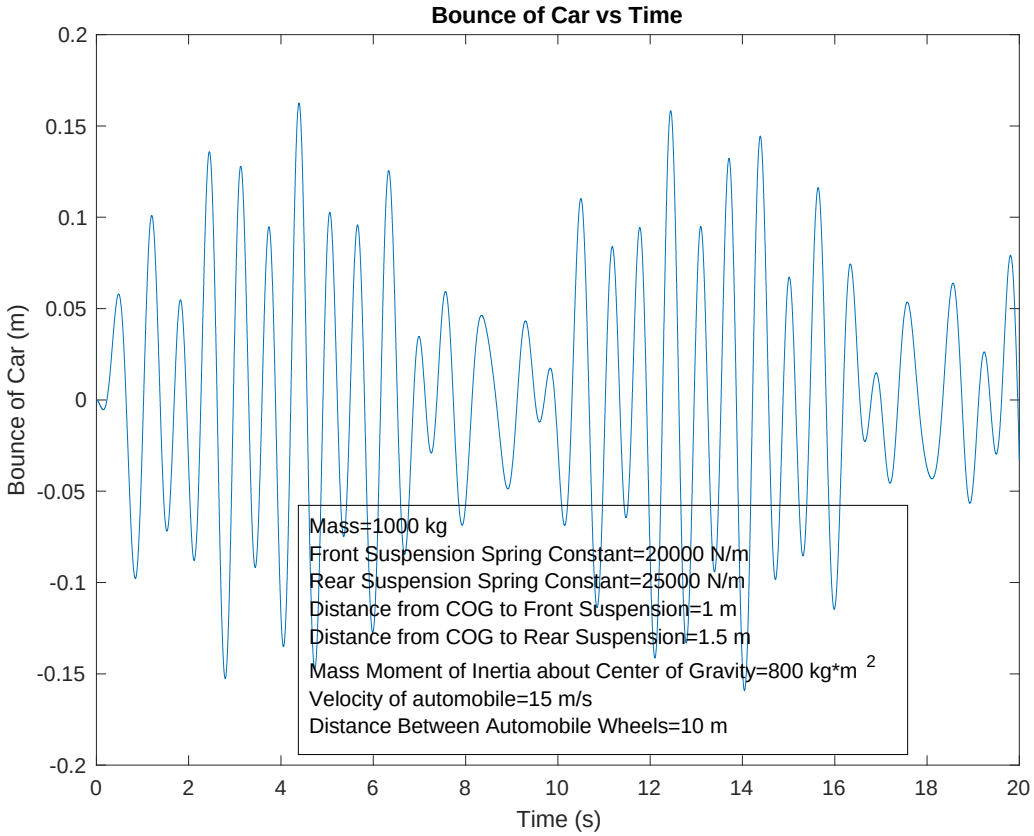

Figure 10: Bounce of Car vs Time

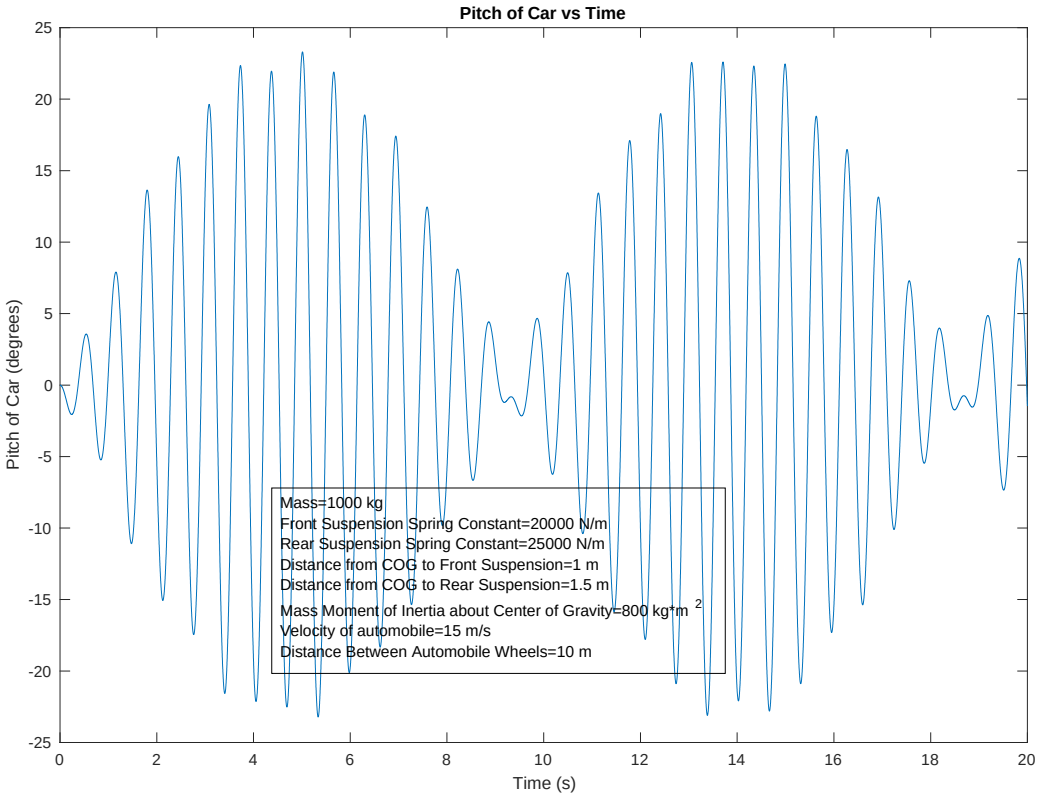

Figure 11: Pitch of Car vs Time

Conclusion:

The objective of this project was completed by deriving the equations of motion governing the response of the automobile as it traversed across a rough road. In order to solve the second order, non-homogenous, ordinary differential equations describing the motion of the automobile the equations of motion were put into state space form. Once the state space matrices were determined, a MATLAB script was written to solve the equations of motion simultaneously. As can be seen by reviewing Figures 9-11, the position of the center of gravity in the vertical direction as well as the angular displacement about the center of gravity was successfully solved for over a time period of interest.

Overall, this project helped to further improve my ability to computationally solve differential equations and better reinforce my ability to derive equations of motion as is learned in system dynamics.

References:

N/A for this project. Professor Kang ensured I was fully prepared to analyze this type of problem by covering all necessary content in ME-331: System Dynamics.

MATLAB Script:

```
%Mason Averill
%ME-544 Fall 2020, 10/5
%Special Problem 3


m=1000;%mass of car in kg
kf=20*10^3;%spring constant of front shock in N/m
kr=25*10^3;%spring constant of rear shock in N/m
l1=1;%length from COG to front shock in m
l2=1.5;%length from COG to rear shock in m
IG=800;%Mass moment of inertia about the COG in kg*m^2
v=15;%velocity of vehicle in m/s
d=10;%distance between wheels in m

max_time=20;%max time period of interest in seconds
step_size=0.001;%adjust this value to increase the resolution of the model,
with a lower value indicating a higher resolution
t=0:step_size:max_time; % time span of interest and fixed time step

yf=.05*sin(2*pi()*v*t/d);%displacement function describing road for front
shock
yr=.05*sin(2*pi()*v*t/d-2*pi()*(l1+l2)/d);%displacement function describing
road for rear shock

initial_conditions=[0;0;0;0]; % initial condition for position and velocity,
in x and theta
A=[0 1 0 0;-(kf+kr)/m 0 -(kr*l2-kf*l1)/m 0;0 0 0 1; -(l2*kr-l1*kf)/IG 0 -
(l1^2*kf+l2^2*kr)/IG 0]; % state matrix A
B=[0 0 0 0;0 kr/m 0 kf/m; 0 0 0 0; 0 l2*kr/IG 0 -l1*kf/IG]; % input matrix B
C=[1 0 0 0;0 1 0 0; 0 0 1 0; 0 0 0 1]; % output matrix C
D=[0 0 0 0;0 0 0 0; 0 0 0 0; 0 0 0 0]; % direct transmission matrix D


%populating the displacement matrix with values for the downward ground
%displacemnt for the front and rear wheels for each increment in time
%considered
[a1,a2]=size(t);
t(a2)=[];
u=zeros(4,a2-1);
i=1;
while(i<a2)
u(2,i)=yr(i);
u(4,i)=yf(i);
i=i+1;
end

%displacement matrix is now populated with values

%passing all matrices to the lsim function built into MATLAB

z=lsim(ss(A,B,C,D),u,t,initial_conditions);
z1=z(:,1); % extract z1, the position in x in meters
```

```
z2=z(:,2); % extract z2, the velocity in x in meters/second
z3=z(:,3); % extract z3, the angular position in radians
z4=z(:,4); % extract z4, the angular velocity in rad/s

%convert the angular displacement (pitch) of the automobile into degrees
pitch_degrees=z3*180/pi();



string_mass=num2str(m);
string_front_suspension_spring_constant=num2str(kf);
string_rear_suspension_spring_constant=num2str(kr);
string_distance_to_front_suspension=num2str(l1);
string_distance_to_rear_suspension=num2str(l2);
string_mass_moment_of_inertia=num2str(IG);
string_velocity=num2str(v);
string_distance_between_wheels=num2str(d);

string_mass=strcat('Mass= ',string_mass,' kg');
string_front_suspension_spring_constant=strcat('Front Suspension Spring
Constant= ',string_front_suspension_spring_constant,' N/m');
string_rear_suspension_spring_constant=strcat('Rear Suspension Spring
Constant= ',string_rear_suspension_spring_constant,' N/m');
string_distance_to_front_suspension=strcat('Distance from COG to Front
Suspension= ',string_distance_to_front_suspension, ' m');
string_distance_to_rear_suspension=strcat('Distance from COG to Rear
Suspension= ',string_distance_to_rear_suspension, ' m');
string_mass_moment_of_inertia=strcat('Mass Moment of Inertia about Center of
Gravity= ', string_mass_moment_of_inertia, ' kg*m^2');
string_velocity=strcat('Velocity of automobile= ',string_velocity,' m/s');
string_distance_between_wheels=strcat('Distance Between Automobile Wheels=
',string_distance_between_wheels, ' m');

string_for_print=sprintf('%s\n%s\n%s\n%s\n%s\n%s\n%s\n%s',string_mass,string_
front_suspension_spring_constant,string_rear_suspension_spring_constant,strin
g_distance_to_front_suspension,string_distance_to_rear_suspension,string_mass
_moment_of_inertia,string_velocity,string_distance_between_wheels);


%Bounce and Pitch vs Time
figure(1)
plot(t,z1)
title('Bounce and Pitch of Car vs Time')
xlabel('Time (s)')
ylabel('Bounce of Car (m) and Pitch of Car (degrees)')
hold on
plot(t,pitch_degrees)
legend('Bounce of Car','Pitch of Car')
annotation('textbox',[0.3 .1 0.4
0.3],'String',string_for_print,'FitBoxToText','on');
grid on
hold off

%Bounce vs Time
figure (2)
```

```
plot(t,z1)
title('Bounce of Car vs Time')
xlabel('Time (s)')
ylabel('Bounce of Car (m)')
annotation('textbox',[0.3 .1 0.4
0.3],'String',string_for_print,'FitBoxToText','on');
grid on

%Pitch vs Time
figure (3)
plot(t,pitch_degrees)
title('Pitch of Car vs Time')
xlabel('Time (s)')
ylabel('Pitch of Car (degrees)')
annotation('textbox',[0.3 .1 0.4
0.3],'String',string_for_print,'FitBoxToText','on');
grid on

%Bounce and Pitch vs Time
figure(4)
plot(t,z1)
title('Bounce and Pitch of Car vs Time')
xlabel('Time (s)')
ylabel('Bounce of Car (m) and Pitch of Car (radians)')
hold on
plot(t,z3)
legend('Bounce of Car','Pitch of Car')
annotation('textbox',[0.3 .1 0.2
0.2],'String',string_for_print,'FitBoxToText','on');
grid on
hold off
```