

EET 438a
Control Systems Technology
Laboratory 3
Introduction to Control Systems Modeling with MATLAB/Simulink

Laboratory Learning Objectives

After completing this laboratory you will be able to:

- 1.) Convert a given differential equation model of a mechanic system to a transfer function.
- 2.) Use MATLAB control system toolbox functions and the MATLAB program to determine dynamic system responses.
- 3.) Determine mechanical system time response to a step function.
- 4.) Compute and interpret the frequency response of a dynamic system using MATLAB to perform computations and plot graphs.
- 5.) Use Simulink software to compute time responses of simple systems.

Theoretical and Technical Background

Modern computational packages and programs allow control systems designers to develop and simulate control systems without intensive hand calculations. MATLAB is a popular scientific and engineering computational package that allows engineers to focus on technical problem solving by providing libraries of mathematical functions for various types of analysis. The control systems toolbox and Simulink are MATLAB tools for control systems analysis and design.

This laboratory introduces control system functions and a control system simulator. An introduction to the MATLAB program environment shows how to use the programs basic functionality, create and analyze control system transfer functions, and use the graphical simulation capabilities of Simulink to test control system designs. Students should view the online videos prior to attempting any of the analysis and design activities given later in this document. Access these videos through the course website or through the Desire-to-Learn platform.

This lab uses MATLAB and Simulink to study the dynamic response of a simplified automotive suspension system. Figure 1 shows the system this lab examines. The variable of interest is the centerline position of the axle that holds the tire, $x(t)$. The input to the system is the road surface $y(t)$, which can vary independently. We assume that the mass of the car body does not move with respect to the car axle. This

eliminates the vertical car body motion, which simplifies the analysis. This may not be a realistic assumption but gives a first approximation of the suspension performance.

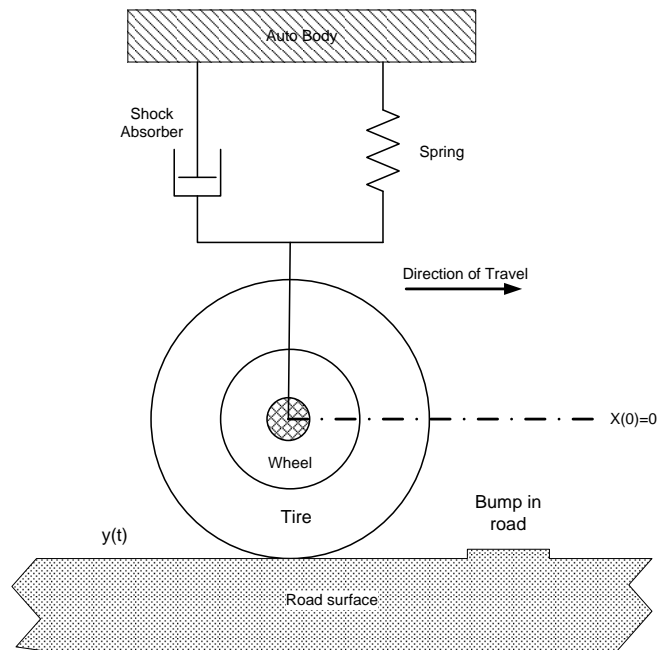


Figure 1. Auto Suspension System Used In Analysis.

Figure 2 shows the mechanical model used for the analysis. A mass represents the tire/wheel assembly and a spring with a spring constant of, K_1 , represents the “springiness” of tire.

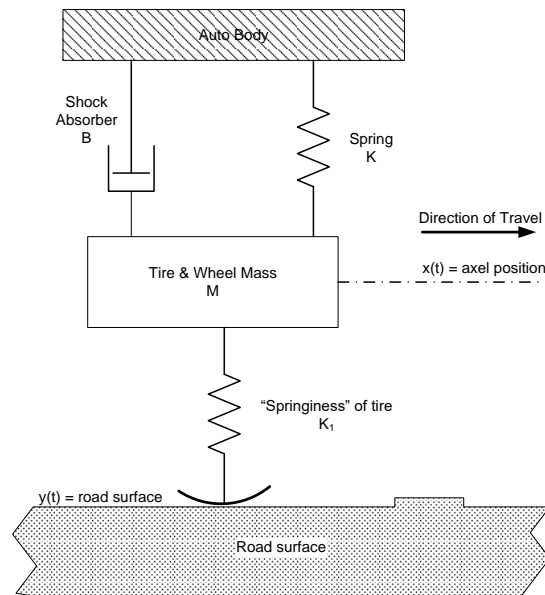


Figure 2. Mechanical Model of the Auto Suspension System Showing the Tire as a Spring.

A unit step function will represent the bump in the road surface. Table 1 lists the model parameters for the initial analysis. The first step in analyzing this system is to reduce the system model in Figure 2 to a free-body diagram that shows all forces and their assumed direction. Figure 3 shows the forces on the free-body diagram. These forces must be written in terms of the axle and road surface positions. Notice that all forces

Table 1 Model Parameters

Parameter	Value
K	1.751×10^5 N/m
K_1	3.5×10^4 N/m
M	14 Kg
B	536 N-s/m

are written as functions of time. Figure 3 also shows the mathematical representation of a bump on the road surface. A unit step function represents a rapid change in the road at some time, t_b , after the auto starts in motion. The surface has a level change of 0.05 m (5 cm).

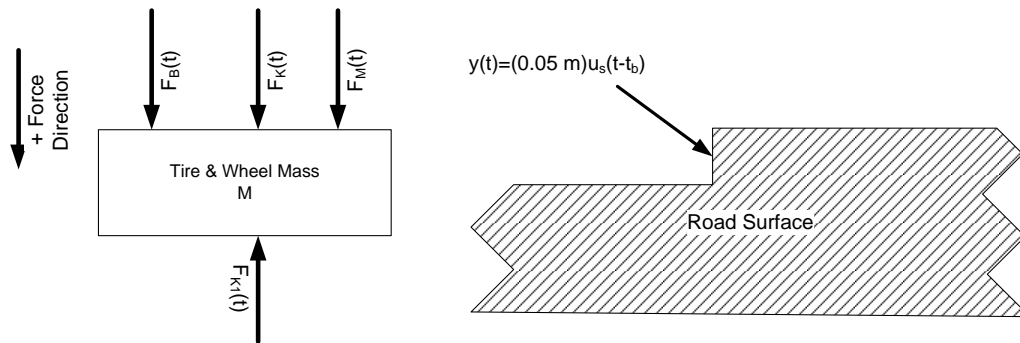


Figure 3. Free-Body Diagram and Mathematical Model of Road Bump.

Summing forces with down assumed to be the positive direction gives the following force balance equation.

$$F_B(t) + F_K(t) + F_M(t) - F_{K_1}(t) = 0 \quad (1)$$

$$F_B(t) + F_K(t) + F_M(t) = F_{K_1}(t)$$

Now represent the forces as functions of the axle position, $x(t)$ and the road surface, $y(t)$ using the force definitions from the course lecture presentations. The force $F_{K_1}(t)$ depends on the difference between the axle and the road surface. Substituting these

$$F_B(t) = B \cdot \frac{dx(t)}{dt} \quad F_{K_1}(t) = K_1 \cdot (y(t) - x(t))$$

$$F_K(t) = K \cdot x(t) \quad F_M(t) = M \cdot \frac{d^2x(t)}{dt^2}$$

relationships into equation (1) yields a second order differential equation with an independent function , $y(t)$, representing a changing road surface and the unknown axle position function, $x(t)$. Equation (2d) is the simplified version of the system model

$$\begin{aligned}
 M \cdot \frac{d^2x(t)}{dt^2} + B \cdot \frac{dx(t)}{dt} + K \cdot x(t) &= K_1 \cdot (y(t) - x(t)) & (a) \\
 M \cdot \frac{d^2x(t)}{dt^2} + B \cdot \frac{dx(t)}{dt} + (K + K_1) \cdot x(t) &= K_1 \cdot y(t) & (b) \\
 \left[\frac{M}{K_1} \right] \cdot \frac{d^2x(t)}{dt^2} + \left[\frac{B}{K_1} \right] \cdot \frac{dx(t)}{dt} + \left[\frac{(K + K_1)}{K_1} \right] \cdot x(t) &= y(t) & (c) \\
 \left[\frac{M}{K_1} \right] \cdot \frac{d^2x(t)}{dt^2} + \left[\frac{B}{K_1} \right] \cdot \frac{dx(t)}{dt} + \left[\frac{K}{K_1} + 1 \right] \cdot x(t) &= y(t) & (d)
 \end{aligned}
 \tag{2}$$

showing the parameters in the differential equation.

Equation (2d) is the differential equation model of the auto suspension with all parameter included. Taking the Laplace transform of this equation and solving for the ratio $X(s)/Y(s)$ produces the transfer function of the dynamic system. Equation (3d) is the transfer function of the system for an arbitrary input for the road surface.

$$\begin{aligned}
 \left[\frac{M}{K_1} \right] \cdot \frac{d^2x(t)}{dt^2} + \left[\frac{B}{K_1} \right] \cdot \frac{dx(t)}{dt} + \left[\frac{K}{K_1} + 1 \right] \cdot x(t) &= y(t) & (a) \\
 \left[\frac{M}{K_1} \right] \cdot s^2 \cdot X(s) + \left[\frac{B}{K_1} \right] \cdot s \cdot X(s) + \left[\frac{K}{K_1} + 1 \right] \cdot X(s) &= Y(s) & (b) \\
 X(s) \left[\left[\frac{M}{K_1} \right] \cdot s^2 + \left[\frac{B}{K_1} \right] \cdot s + \left[\frac{K}{K_1} + 1 \right] \right] &= Y(s) & (c) \\
 \frac{X(s)}{Y(s)} &= \frac{1}{\left[\frac{M}{K_1} \right] \cdot s^2 + \left[\frac{B}{K_1} \right] \cdot s + \left[\frac{K}{K_1} + 1 \right]} & (d)
 \end{aligned}
 \tag{3}$$

If we assume a unit step function for the road surface with a displacement of y_d then the following derivation results.

$$\left[\frac{M}{K_1} \right] \cdot \frac{d^2x(t)}{dt^2} + \left[\frac{B}{K_1} \right] \cdot \frac{dx(t)}{dt} + \left[\frac{K}{K_1} + 1 \right] \cdot x(t) = y_d \cdot u_s(t) \quad (a)$$

$$\left[\frac{M}{K_1 \cdot y_d} \right] \cdot \frac{d^2x(t)}{dt^2} + \left[\frac{B}{K_1 \cdot y_d} \right] \cdot \frac{dx(t)}{dt} + \left[\frac{K}{K_1 \cdot y_d} + \frac{1}{y_d} \right] \cdot x(t) = u_s(t) \quad (b)$$
(4)

Equation (4b) is the differential equation model of the auto suspension with a road bump represented by a unit step function that has a vertical displacement of y_d .

We can use equations (3d) or (4b) to determine the response of the suspension system by entering these functions into the MATLAB computational program. View the tutorial videos to learn the basics of the MATLAB programming environment.

MATLAB is a powerful programming, graphing, and computational environment used by scientists and engineers to solve problems that require extensive numerical calculations and plot results¹. It comes with an large library of functions and a dynamic systems' simulation program called Simulink. Simulink allows a user to make a block diagram model of a dynamic system very similar to the block diagrams covered in the lecture of this course. The program then solves the block diagram and displays results with time plots.

A user can enter MATLAB instructions at the command prompt and perform calculations or enter a series of commands into a text file and type the filename on the command prompt. The commands in the text file form a script that MATLAB executes like a standard program. The control toolbox functions of MATLAB can create a digital representation of a transfer function from arrays of coefficients representing the numerator and denominator of a derived transfer function.

Appendix B lists the code used to generate the suspension response to a 5 cm abrupt jump in the road surface. Entering these instructions into a text file, which is called an M-file (script in current terminology) in the MATLAB programming system produces a time response and a frequency response plot. The program also prints the maximum deflection of the axle and the final position change in centimeters. The lab will use this code later.

Figures 4 and 5 are the step input time response and the system frequency response respectively. These plots are the responses for the parameter values given in Table 1. The numerical results of the simulation show that the axle's maximum displacement is 1.34 cm and the final displacement is 0.832 cm. Most of the surface bump is absorbed by the "springiness" of the tire. Figure 6 shows how this might look as the tire deforms due to the road surface shift.

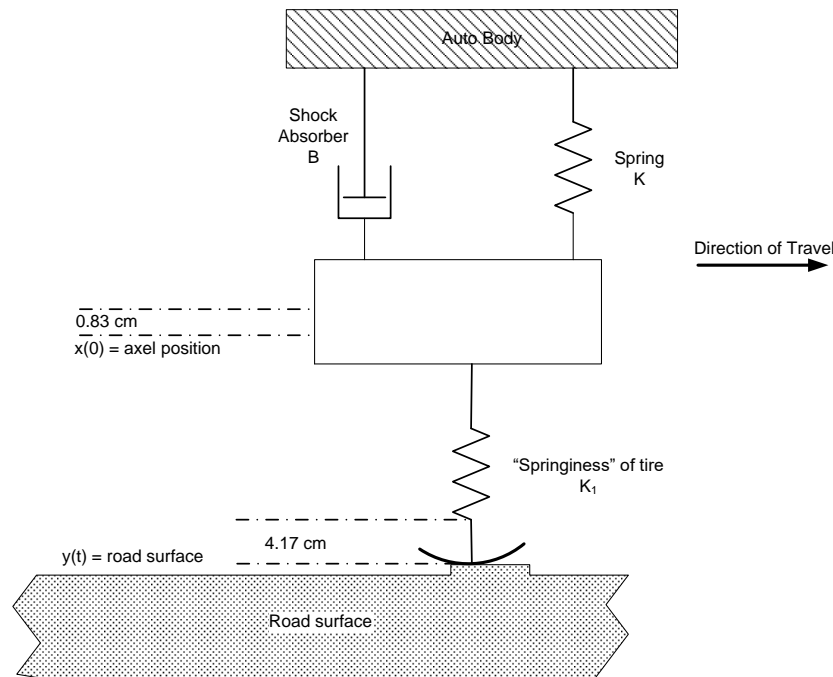


Figure 6. Suspension System after Encountering a 5 Cm Road Displacement.

The axle of the suspension system will have a final displacement of 0.83 cm from the zero position referenced to the flat road surface. The tire absorbs the remaining 4.17 cm of the shift as it deforms to meet the new road height. This assumes that the auto body does not move.

The time plot in Figure 4 shows that the axle exhibits a damped oscillation that settles out by 300 mS. The maximum displacement occurs during the first upward swing of the axle and has decreasing amplitudes with each successive oscillation. Decreasing the damping provided by the shock-absorbers will cause the axle oscillations to last longer as the energy stored in the mass and the springs cannot dissipate. With softer shocks the wheel will bounce for a longer time.

The Bode plot shows there is a system resonant peak at approximately 120 rad/sec (19 Hz). Any sinusoidal displacements that occur at this frequency are transmitted through the suspension with a gain of -31.4 dB. The greatest axle displacements will occur at this frequency. Stimulations of frequencies less than the peak pass with a gain of approximately -40 dB. The system gain falls off quickly after the peak frequency. Disturbances with periods in this range are attenuated, so the axle will not respond to these higher frequency disturbances. The axle displacement will be less since the gains are lower.

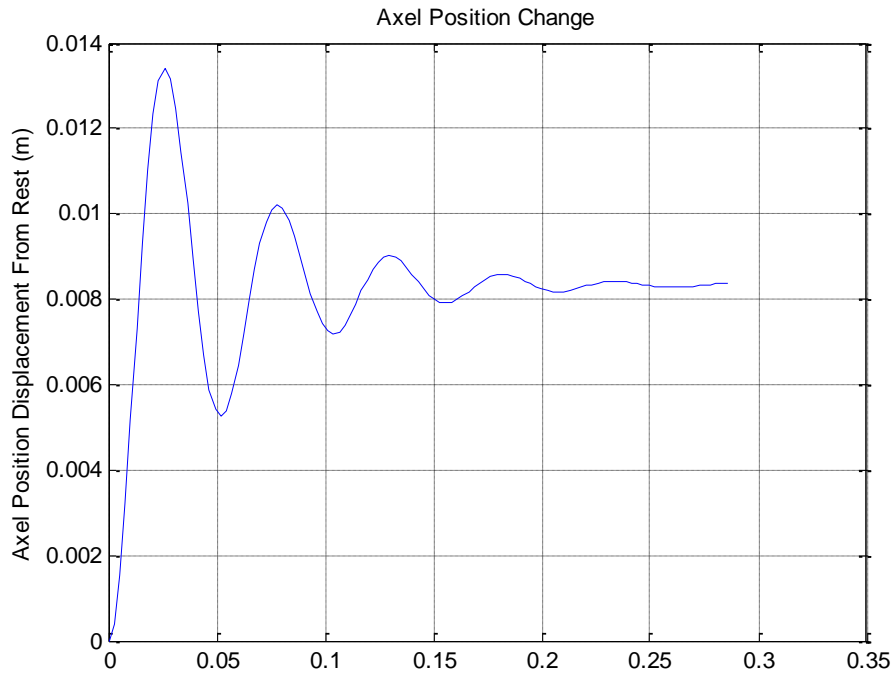


Figure 4. Suspension System Response Due to a 5 cm Displacement in the Road Surface

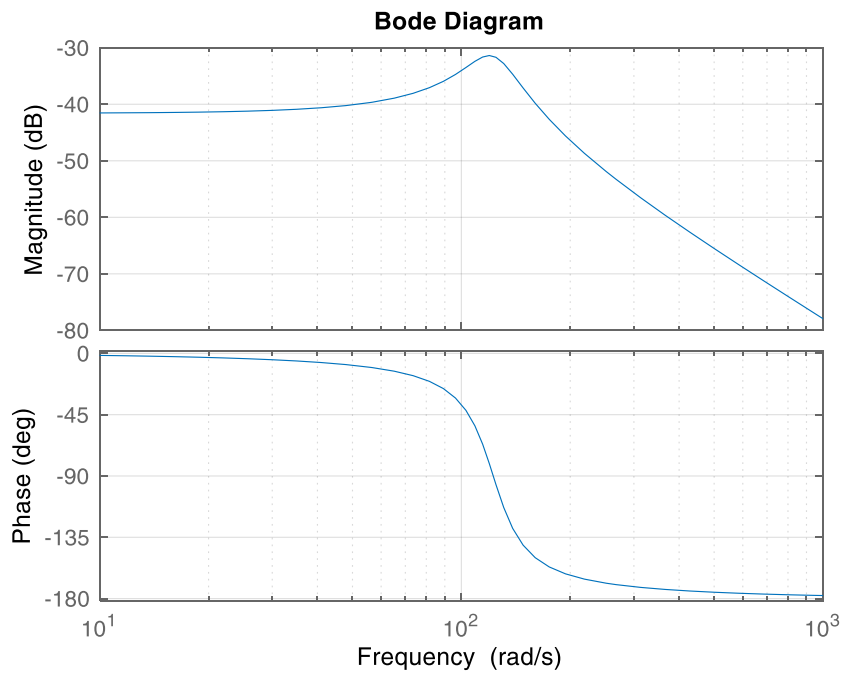


Figure 5. Frequency Response of Auto Suspension System.

The frequency response plot shows how the suspension system reacts to periodic sinusoidal functions. Studies show that paving machines lay road surfaces with a wavelength of 15.4 m (50 ft)². The amplitude of this variation is 7.62 cm (3 in.). Ripples with shorter wavelengths can occur usually of roads that have a "washboard" pattern. This pattern is approximately 0.308 m (1 ft). These roads have surface variations of smaller amplitudes in the range of 1.27 cm (0.5 in.).

The vehicle velocity and the wavelength determine the frequency of the system input function. Equation (5) is the relationship between velocity, wavelength and frequency.

$$\lambda = \frac{v}{f} \tag{5}$$

Where: λ = the wavelength (m)
 v = vehicle velocity (m/s)
 f = frequency (Hz)

The Bode plots give frequency, ω , in radians/second. Solving (5) for frequency and converting Hertz to radians/second produces equation (6).

$$\omega = \frac{2\pi \cdot v}{\lambda} \tag{6}$$

Table 2 lists the paving wavelengths and two vehicle test velocities along with the computed values of ω .

Table 2. Suspension Test Frequencies

	Wavelength (m)	
Velocity (m/s)	15.4	0.308
27 m/s (60 mph)	11.06 rad/s	553 rad/s
13.5 m/s (30 mph)	5.53 rad/s	276 rad/s

Finding the intersections of these frequencies with the Bode plot in Figure 5 gives the suspension system gain to the specified frequencies in Decibels. Converting the dB value into a gain value and multiplying it by the amplitudes listed above computes the axle motion. Equations (7ab) show the formulas used for this calculation. The variable G is the gain of the suspension system to a given frequency. The variable dB is the gain value read from the Bode plot gain response plot. The quantity x_{max} is the maximum amplitude of the sinusoidal stimulation to the suspension system.

$$G = 10^{\frac{dB}{20}} \quad (a)$$

$$x_{axle} = G \cdot x_{max} \quad (b)$$
(7)

Figure 7 shows the data cursor reading for the 11.7 rad/s frequency. The cursor in MatLAB figures only allows alignment with computed plot points so the gain value estimate is -41.5 dB. Table 3 shows the gain values for all sinusoidal input

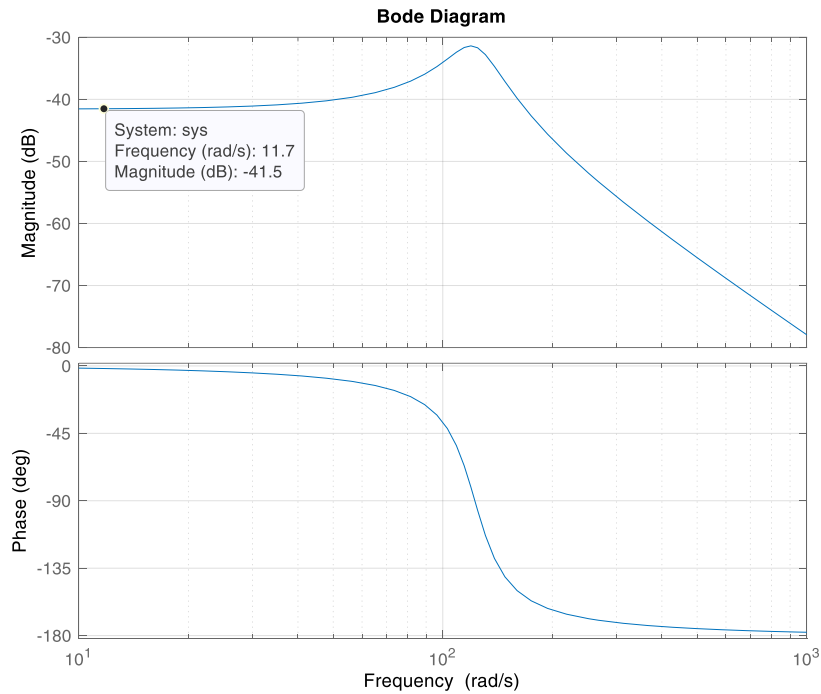


Figure 7. Suspension System Bode Plot Showing Data Cursor Located Near 11.7 Rad/S.

frequencies. The suspension system cannot respond to the higher frequency displacements. The large negative dB values produce very small gains resulting

Table 3. Suspension System Frequency Response

Frequency (rad/s)	Gain (dB)	Gain (A)	Maximum Displacement (cm)	Axle Displacement (cm)
5.53	-41.5	0.0084	7.62	0.0632
11.70	-41.5	0.0084	7.62	0.0632
276	-54	0.0020	1.27	0.00254
553	-67.3	0.00043	1.27	0.00055

in tiny changes in axle position. The lower frequencies produce much larger axle displacements. The suspension system acts as a low-pass filter allowing a fraction of the sinusoidal displacements at lower frequencies to disturb the axle position while rejecting the higher frequencies. Frequencies between 80 and 170 rad/s fall in the resonant peak and transmit greater amplitudes of displacement to the axle.

Simulink allows users to draw a block diagram of a dynamic system and study its response without writing m-file scripts. Simulink has a variety of input functions and can mathematically model very complex systems. Scope-like blocks connected to signal paths display simulation results for user interpretation. Figure 8 shows the Simulink

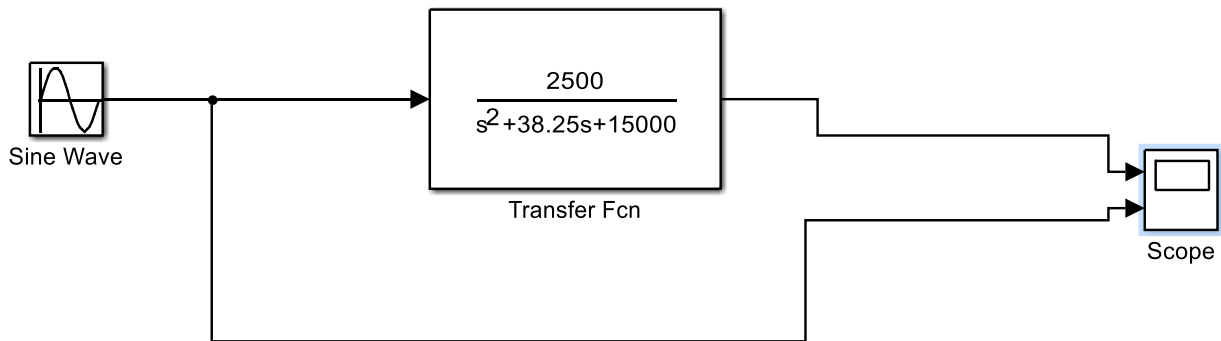


Figure 8. Simulink Model of Suspension System.

model of the suspension system. A sine wave input excites the normalized suspension system transfer function. This transfer function output scale is 0-1. The scope icon displays the system input and output on two separate plots since there are significant differences between the amplitudes. View the tutorial video presentations to learn how to use Simulink.

Simulink produces numerical solutions to the differential equations describing dynamic systems. All calculations assume that time is the independent variable. The maximum simulation time must be set in the program to produce usable time plots. An accompanying tutorial video shows how to set this parameter. Figure 9 shows simulation results for a sinusoidal input 276 rad/sec with maximum amplitude of 1.27

cm. After the initial transient period, the peak amplitudes closely agree with the value found in Table 3.

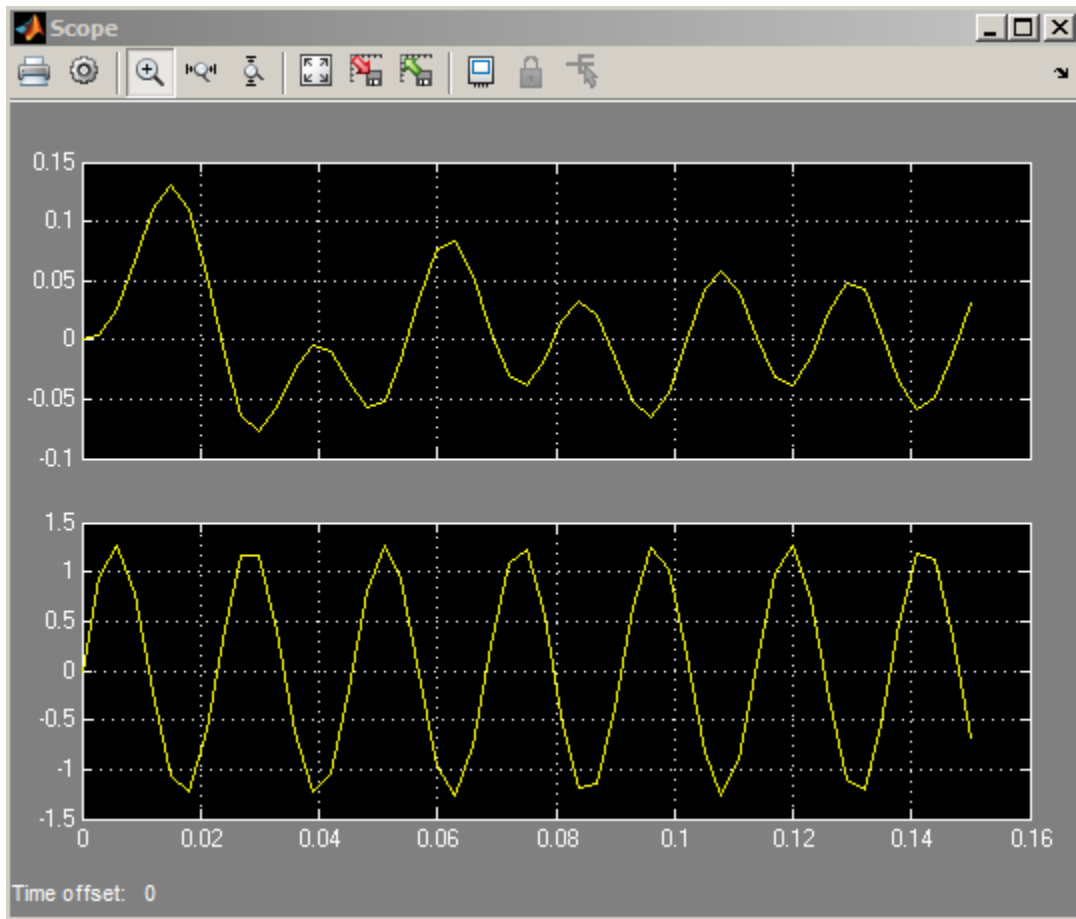


Figure 9. Simulink Scope Output for 276 rad/sec Sinusoidal Input $x_{\max}=1.27$ cm

Procedure

1. Acquire a copy of the student version of MatLAB and install it on your computer. Alternately you can make use of the copies of MatLAB installed on computers in D122, and D30 on campus.
2. Open a new m-file and enter the code listed in Appendix B. Save this file under the name **shock**. Remember that MatLAB is case sensitive so use all lower case letters. Typing this name in at the command line will run the program prompting the user for the required inputs. Run the program with the parameters from Table 1 to check your code for errors. The program should produce two plots that look like Figures 4 and 5. Fix any syntax errors or other code mistakes if they occur.
3. Run the program for each set of parameters in Table C-1 located in Appendix C. Record the numerical values of the maximum axle displacement and the final axle displacement in the table. Cut and paste each plot the program generates into a Word document and save them for later analysis.
4. Run the program for each set of parameters in Table C-2. Copy down the transfer function coefficients from each case and save them for future use. Cut and paste the frequency response plots for each case into a Word document for later analysis. Use the data cursors on the plots to determine the axle displacement response in dB to a road-induced vibration of 100 rad/s.
5. Create a Simulink model using Figure 8 as a guide. Use the transfer function coefficients found in step 4 above to define the transfer function block in the Simulink model. The sinusoidal input should have a 2 cm peak input and a frequency of 100 rad/s. Run the model simulation and record the estimated peak values of the output sinusoidal response in Table C-3
6. Changes in tire pressure change the value of K_1 . Table C-4 lists these values of K_1 and the other test parameters. Use the shock program to determine the impact changing tire pressure has on the final axle displacement. Save the transfer function coefficients computed after each program run for later use. Record the final displacement values the program computes and save them for later analysis.
7. Change the input of the Simulink program shown in Figure 8 to a unit step function with a final value of 0.05 m. Repeat step 6 using Simulink. Estimate the final value of the axle from the scope plots. Place these results in Table C-4 also.

Lab 3 Assessment

Submit the following items for grading and perform the listed actions to complete this laboratory assignment.

1. Plot the maximum axle displacement against the shock absorber damping, B using the data collected in part 3 of the procedure. Write a page describing the changes in the time and frequency response of the suspension as the damping increased.
2. Use Equation (7) to compute the magnitude of the axle displacement, x_{axle} , that occurs for each shock damping value give in Table C-2 from the dB value recorded. Create a table that lists in columns the values of B , dB at 100 rad/s, G , axle displacement. Use a x_{max} value of 2 cm in Equation (7) for all cases. Write a paragraph that explains the response of the system to the changes in shock damping.
3. Compare the Simulink model sinusoidal peak values to those found from the frequency response plots that used the same parameters. Are the values nearly equal? Write a paragraph that explains why these two responses are similar.
4. Plot the final displacement values (y-axis) found from varying the parameter K_1 against the value of K_1 (x-axis). Write a short explanation that accounts for the trend in the axle displacement.
5. Compare the results from the MatLAB program and the Simulink model. Is there good agreement between the two programs?

References

1. Matlab Users Guide, Student Edition, Mathworks Inc.
2. Control Systems Engineering, William J. Palm, III, John Wiley and Sons, Inc, 1986.

Appendix A
Lab 3 MATLAB Function Reference

TF Creation of transfer functions or conversion to transfer function. This function creates a transfer function from two arrays of coefficients.

Creation:

`SYS = TF(NUM,DEN)` creates a continuous-time transfer function `SYS` with numerator(s) `NUM` and denominator(s) `DEN`. The output `SYS` is a TF object.

STEP Step response of LTI models. This function plots the response of a system to a unit step input.

`STEP(SYS)` plots the step response of the LTI model `SYS` (created with either TF, ZPK, or SS). For multi-input models, independent step commands are applied to each input channel. The time range and number of points are chosen automatically.

`STEP(SYS,TFINAL)` simulates the step response from $t=0$ to the final time $t=TFINAL$. For discrete-time models with unspecified sampling time, `TFINAL` is interpreted as the number of samples.

`STEP(SYS1,SYS2,...,T)` plots the step response of multiple LTI models `SYS1,SYS2,...` on a single plot. The time vector `T` is optional. You can also specify a color, line style, and marker for each system, as in

`step(sys1,'r',sys2,'y--',sys3,'gx')`.

BODE Bode frequency response of LTI models. This function produces a Bode plot for a specified linear time-invariant (LTI) system representing a transfer function.

`BODE(SYS)` draws the Bode plot of the LTI model `SYS` (created with either TF, ZPK, SS, or FRD). The frequency range and number of points are chosen automatically.

`BODE(SYS,{WMIN,WMAX})` draws the Bode plot for frequencies between `WMIN` and `WMAX` (in radians/second).

INPUT Prompt for user input.

`R = INPUT('How many apples')` gives the user the prompt in the text string and then waits for input from the keyboard.

The input can be any MATLAB expression, which is evaluated, using the variables in the current workspace, and the result returned in `R`. If the user presses the return key without entering anything, `INPUT` returns an empty matrix.

`R = INPUT('What is your name','s')` gives the prompt in the text string and waits for character string input. The typed input is not evaluated; the characters are simply returned as a MATLAB string.

The text string for the prompt may contain one or more `'\n'`. The `'\n'` means skip to the beginning of the next line. This allows the prompt string to span several lines. To output just a `'\'` use `'\\'`.

Appendix B

M-File for Generating Step and Bode Response of Auto Suspension

Note: MATLAB is case sensitive. (i.e. H does not equal h)

```
% This MATLAB script file holds a list of commands that are executed
% in the order they are typed. This file will produce a step response to a
% transfer functions of the auto suspension system. The user must enter the
% parameters

% clear all variable from memory and close all existing plots.
clear all;
close all;

%Read all the input values.
k=input('Enter the value of K(N/m): ');
k1=input('Enter the value of K1 N/m): ');
B=input('Enter the value of B (N-s/m): ');
M=input('Enter the mass of the tire and wheel (Kg): ');
yd=input('Enter the bump displacement in cm. Enter 0 for no bump(cm): ');

%take care of the zero entry since this parameter is in the denominator of the
transfer function formula
if yd==0
    yd=1;
else
    yd=yd/100; %scale the bump to meters
end

%compute the coefficient values.

a2=M/(k1*yd);
a1=B/(k1*yd);
a0=k/(k1*yd)+1/yd;

%create the arrays for the TF function
num=1; % define the numerator and denominator arrays
dem=[a2 a1 a0];
sys=tf(num,dem) %display this on the screen by not ending the statement
with semicolon.
[y t]=step(sys); % this function produces the step response
plot(t,y);
% these lines of code format the plot giving it a title and axis names
title('Axle Position Change');
grid on;
if yd==1
    ylabel('Axle Position Displacement From Rest (per unit)');
else
    ylabel('Axle Position Displacement From Rest (m)');
end
figure; %create an new figure for the bode plot
bode(sys);
grid on;
```

```

max_x=max(y); % find the maximum displacement
N=length(y); % find the length of the array to get final value
final_x=y(N); % find the final value in the plot

%print the numerical values to the consol.
fprintf('\n\n'); % some line feeds to get some space
if yd==1
% Scale the values to per unit and print
fprintf('The maximum axle displacement is: %8.6f (cm)\n',max_x);
fprintf('The final axle displacement is: %8.6f (cm)\n',final_x);
fprintf('\n\n');

else
% Scale the values to cm and print
fprintf('The maximum axle displacement is: %8.6f (cm)\n',max_x*100);
fprintf('The final axle displacement is: %8.6f (cm)\n',final_x*100);
fprintf('\n\n');
end

```

Appendix C
Lab Data Tables

For all cases let:

$K=175,100$ N/m

$K_1=35,700$ N/m

$M= 14$ Kg

Bump Displacement: 5 cm

Table C-1. Shock Damping Test Cases

Case Number	B (Shock Damping) N-s/m	X_{max} (cm)	X_{final} (cm)
1	600		
2	800		
3	1000		
4	1200		
5	1500		
6	2000		
7	2500		
8	3000		

For all cases let:

$K=175,100$ N/m

$K_1=35,700$ N/m

$M= 14$ Kg

Table C-2. Frequency Responses

Case Number	B (Shock Damping) N-s/m	Bump Displacement (cm)	dB at 100 rad/s
1	600	0	
2	1200	0	
3	2000	0	
4	3000	0	

$$G = 10^{\frac{dB}{20}}$$

$$X_{axle} = G \cdot X_{max}$$

Table C-3. Simulink Suspension Output Peak Values

Case Number	B (Shock Damping) N-s/m	Estimated Peak Axle Displacement (cm)
1	600	
2	1200	
3	2000	
4	3000	

For all cases let:

$K=175,100$ N/m

$B=800$ N-s/m

$M= 14$ Kg

Bump Displacement: 5 cm

Table C-4. Parameter K_1 Changes Due To Tire Pressure

Case Number	K_1 Tire Spring Constant	Final Axle Displacement (cm) Shock Program	Final Axle Displacement (cm) Simulink
1	15,000		
2	35,000		
3	40,000		
4	50,000		