Suspension Systems and Control

Matlab Case Study for Signals and Systems

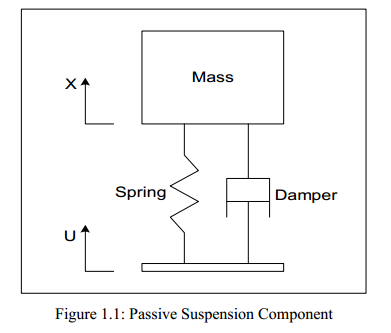
Suspension systems are an invisible phenomenon of our daily lives. Although they are built into nearly every vehicle we use, we only notice their existence when they are broken - as anyone who has driven on a poorly maintained road in an old car can attest. These essential systems incorporate elements of physics, material engineering, signal processing, and control systems in order to ensure that your daily transit is a comfortable one.

In this case study, you will simulate a passive suspension system and analyze its behavior. You will study the relationship between physical properties (such as spring constant and damping coefficient), physical behavior (such as how well the system absorbs shocks), the transfer function of the system, and the location of zeros and poles. You will use these observations to simulate your own suspension system and choose its properties to absorb shocks from potholes and reduce noise from bumpy roads.

You will also explore the use of active suspension systems, which use sensors and hydraulic actuators rather than springs and damping. You will simulate an active suspension system and learn how to implement state feedback to alter the characteristics of a closed-loop transfer function.

# Passive Suspension Systems

A passive vehicle suspension system is a connection between the wheels and the body of a vehicle that is designed to prevent disturbances to the tires - such as bumps, potholes, and vibrations - from affecting the passengers. In its simplest form, it can be modeled as a mass *m* held up by an ideal spring with spring constant *k*, and a damper with damping coefficient *b*. As the vehicle travels along the road, a displacement *u(t)* is applied to one end of the spring, compressing it. This in turn moves the vehicle body with displacement *x(t)*.



Writing out Newton’s second law for this system, we arrive at:

Which we can reformulate as an input-output equation:

# Passive Case Study

Using the *passive\_suspension.m* file as a template, complete the following tasks:

* Use MATLAB to simulate the input-output equation shown above using any method of your choice. (ode45, state-space modeling, or even Simulink). Your model should use zero initial conditions and use the roadSurface vector as its input.
* Use the *animateCar()* function to visualize the results of your simulation. Does the passive suspension system effectively absorb shocks and filter out the bumpiness of the road? Record any observations in your writeup.
* Calculate the transfer function of the system and use it to plot the poles and zeros. (You may find the *tf()* and *pzmap()* functions to be very useful.) What do you notice about the location of poles and zeros? Record any observations in your writeup.
* Experiment with your model; change the spring constant and damping coefficient and note the effects of these changes on the animation and on the pole/zero map. Record any observations in your writeup.
* Follow the instructions in section 2 of the *passive­\_suspension.m* script to devise a passive suspension system that will work effectively for a vehicle that varies in mass. Record your results in your writeup.

# Active Suspension Systems

# Active Case Study

# What to turn in

* Your completed passive\_suspension.m file.
* Your completed active\_supension.m file.
* Any additional functions or scripts you wrote
* A brief writeup including your answer to the case study questions