Alexandria University
Faculty of Engineering
Division of Communications & Electronics

EE 392: Control Systems & Their Components

Lab 3: Suspension system control using root locus control method

3.1 Introduction

Designing an automotive suspension system is an interesting and challenging control problem. When the suspension system is designed, a 1/4 model (one of the four wheels) is used to simplify the problem to a 1D multiple spring-damper system. A diagram of this system is shown below in Fig.1.

This model is for an active suspension system where an actuator is included that is able to generate the control force U to control the motion of the bus body and W is the disturbance due road irregularities.

A good automotive suspension system should have satisfactory road holding ability, while still providing comfort when riding over bumps and holes in the road. When the vehicle is experiencing any road disturbance (i.e. pot holes, cracks, and uneven pavement), the vehicle body should not have large oscillations, and the oscillations should dissipate quickly.

Note: Since the distance X1-W is very difficult to measure, and the deformation of the tire (X2-W) is negligible, we will use the distance X1-X2 instead of X1-W as the output in our problem. Keep in mind that this is estimation.

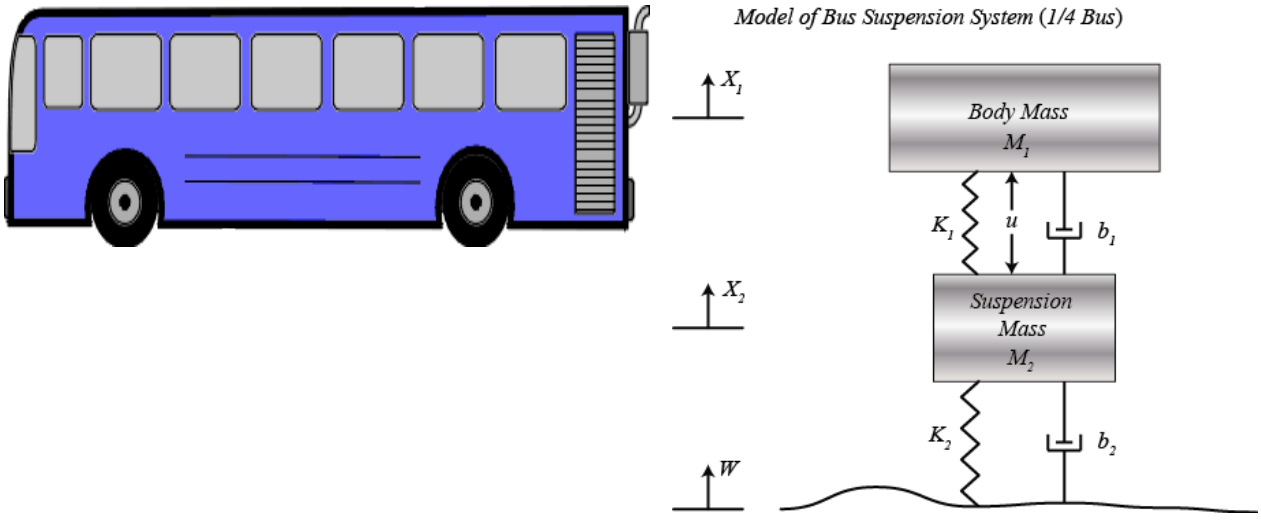


Fig. 1 Suspension system

3.2 Time Domain modeling (Equations of motion)

Applying Newton's second law of motion for masses M_1, M_2 we can get

$$M_1\ddot{X}_1 = -b_1(\dot{X}_1 - \dot{X}_2) - K_1(X_1 - X_2) + U$$
$$M_2\ddot{X}_2 = b_1(\dot{X}_1 - \dot{X}_2) + K_1(X_1 - X_2) + b_2(\dot{W} - \dot{X}_2) + K_2(W - X_2) - U$$

3.3 Transfer function modeling

Applying Laplace transform for the previous equations will be

$$(M_1s^2 + b_1s + K_1)X_1(s) - (b_1s + K_1)X_2(s) = U(s)$$
$$(b_1s + K_1)X_1(s) + (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2))X_2(s) = (b_2s + K_2)W(s) - U(s)$$

$$\begin{bmatrix} (M_1s^2 + b_1s + K_1) & -(b_1s + K_1) \\ -(b_1s + K_1) & (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \begin{bmatrix} U(s) \\ (b_2s + K_2)W(s) - U(s) \end{bmatrix}$$

$$A = \begin{bmatrix} (M_1s^2 + b_1s + K_1) & -(b_1s + K_1) \\ -(b_1s + K_1) & (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) \end{bmatrix}$$

$$\Delta = \det \begin{bmatrix} (M_1s^2 + b_1s + K_1) & -(b_1s + K_1) \\ -(b_1s + K_1) & (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) \end{bmatrix}$$

$$\Delta = (M_1s^2 + b_1s + K_1) \cdot (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) - (b_1s + K_1) \cdot (b_1s + K_1)$$

Find the inverse of matrix A and then multiply with inputs $U(s)$and $W(s)$ on the righthand side as follows:

$$\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) & (b_1s + K_1) \\ (b_1s + K_1) & (M_1s^2 + b_1s + K_1) \end{bmatrix} \begin{bmatrix} U(s) \\ (b_2s + K_2)W(s) - U(s) \end{bmatrix}$$

$$\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} (M_2s^2 + b_2s + K_2) & (b_1b_2s^2 + (b_1K_2 + b_2K_1)s + K_1K_2) \\ -M_1s^2 & (M_1b_2s^3 + (M_1K_2 + b_1b_2)s^2 + (b_1K_2 + b_2K_1)s + K_1K_2) \end{bmatrix} \begin{bmatrix} U(s) \\ W(s) \end{bmatrix}$$

When we want to consider the control input $U(s)$ only, we set $W(s) = 0$. Thus we get the transfer function $G_1(s)$ as in the following:

$$G_1(s) = \frac{X_1(s) - X_2(s)}{U(s)} = \frac{(M_1 + M_2)s^2 + b_2s + K_2}{\Delta}$$

When we want to consider the disturbance input $W(s)$ only, we set $U(s) = 0$. Thus we get the transfer function $G_2(s)$ as in the following:

$$G_2(s) = \frac{X_1(s) - X_2(s)}{W(s)} = \frac{-M_1b_2s^3 - M_1K_2s^2}{\Delta}$$

3.4 Building SIMULINK model

The SIMULINK model can be modeled using discrete components (without ready-made transfer functions) as the following

1. First, we will model the integrals of the accelerations of the masses.

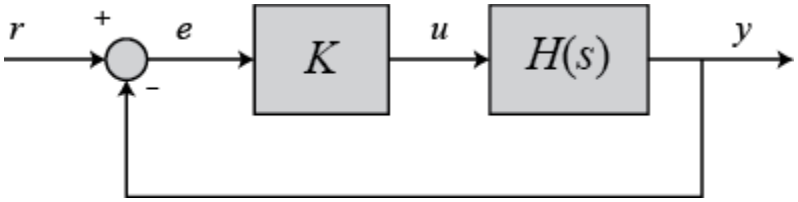
$$\int \int \frac{d^2x_1}{dt^2}\, dt = \int \frac{dx_1}{dt}\, dt = x_1$$

$$\int \int \frac{d^2x_2}{dt^2}\, dt = \int \frac{dx_2}{dt}\, dt = x_2$$

2. Next, we will start to model Newton's law. Newton's law for each of these masses can be expressed as section 3.2
3. We will add in the forces acting on each mass. First, we will add in the force from Spring 1. This force is equal to a constant, k1 times the difference X1-X2
4. We will add in the force from Damper 1. This force is equal to b1 times V1-V2.
5. We will add in the force from Spring 2. This force acts only on Mass 2, but depends on the ground profile, W. Spring 2's force is equal to X2-W.
6. We will add in the force from Damper 2. This force is equal to b2 times V2-d/dt(W). Since there is no existing signal representing the derivative of W we will need to generate this signal
7. Last force is the input U acting between the two masses.

3.5 Root Locus design

The root locus of an (open-loop) transfer function $H(s)$ is a plot of the locations (locus) of all possible closed-loop poles with proportional gain K and unity feedback.



If we write $H(s) = b(s)/a(s)$

Properties of root locus :

- We will consider all positive values of K. In the limit as $K \to 0$, the poles of the closed-loop system are $a(s) = 0$ or the poles of $H(s)$. In the limit as $K \to \infty$, the poles of the closed-loop system are $b(s) = 0$ or the zeros of $H(s)$.
- No matter what we pick K to be, the closed-loop system must always have n poles, where n is the number of poles of $H(s)$. The root locus must have n branches, each branch starts at a pole of $H(s)$ and goes to a zero of $H(s)$. If $H(s)$ has more poles than zeros (as is often the case), $m < n$ and we say that $H(s)$ has zeros at infinity.

- In this case, the limit of $H(s)$ as $s \to \infty$ is zero. The number of zeros at infinity is $n - m$, the number of poles minus the number of zeros, and is the number of branches of the root locus that go to infinity (asymptotes).
- Since the root locus is actually the locations of all possible closed-loop poles, from the root locus we can select a gain such that our closed-loop system will perform the way we want.
- If any of the selected poles are on the right half plane, the closed-loop system will be unstable.
- The poles that are closest to the imaginary axis have the greatest influence on the closed-loop response, so even though the system has three or four poles, it may still act like a second or even first order system depending on the location(s) of the dominant pole(s).

3.6 Relation between Root locus design and time domain response

The time domain response can be adjusted using root locus by first identifying the natural frequency and damping ratio as the following

$$\xi = damping\ ratio = \sqrt{\frac{(\ln OS\%)^2}{(\ln OS\%)^2 + \pi^2}}$$

$$\omega_n = natural\ freq. \approx \frac{1 - 0.417\xi + 2.917\xi^2}{t_r}$$

Where $OS\%$ is the overshoot ratio and t_r is the rise time.

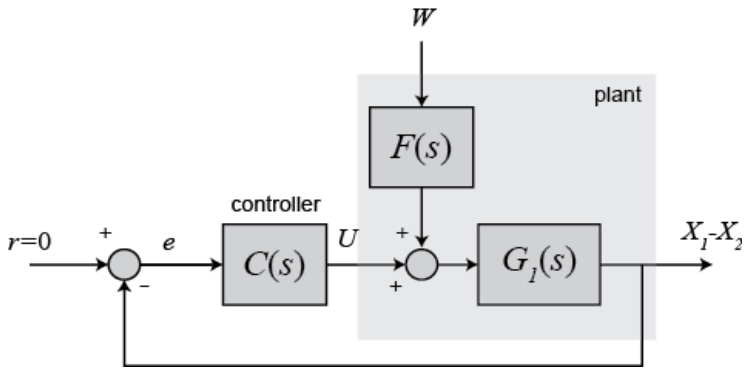
After that identify the positions of the new poles according to ξ, ω_n to make the overshoot less than $OS\%$, the poles have to be in between the two white dotted lines of ξ , and to make the rise time shorter than t_r, the poles have to be outside of the white dotted semicircle of ω_n.The positions of the new poles corresponds to the pure gain –ve feedback controller

Notes

Generally, to get a small overshoot and a fast response, we need to select a gain corresponding to a point on the root locus near the real axis and far from the imaginary axis or the point that the root locus crosses the desired damping ratio line. But in this case, we need the cancellation of poles and zeros near the imaginary axis, so we need to select a gain corresponding to a point on the root locus near the zeros and percent overshoot line

3.7 Adding a notch filter

We will probably need two zeros near the two poles on the complex axis to draw the root locus, leading those poles to the compensator zeros instead of to the plant zeros on the imaginary axis. We'll also need two poles placed far to the left to pull the locus to the left. It seems that a notch filter (2-lead controller) will probably do the job. Let's try putting two poles two zeros.



3.7 Lab requirements

1. Model the suspension system using M-file and Simulink using the following parameters

(M1)	1/4 bus body mass	2500 kg
(M2)	suspension mass	320 kg
(K1)	spring constant of suspension system	80,000 N/m
(K2)	spring constant of wheel and tire	500,000 N/m
(b1)	damping constant of suspension system	350 N.s/m
(b2)	damping constant of wheel and tire	15,020 N.s/m
(U)	control force	

2. design a feedback controller so that when the road disturbance (W) is simulated by a unit step input, the output (X1-X2) has a settling time less than 5 seconds and an overshoot less than 5%. For example, when the bus runs onto a 10 cm high step, the bus body will oscillate within a range of +/- 5 mm and will stop oscillating within 5 seconds.
3. Find the open loop poles. What is the dominant poles ?
4. Plot the root locus of the plant itself. Is the system stable?
5. Find proper damping ratio and natural frequency for the pure gain controller
6. Add a notch-filter controller by putting the poles at 30 and 60 and the zeros at 3+/-3.5i. draw the new root locus
7. Find the gain of the compensator that satisfies the design requirements.
8. Draw the step response of the open loop system and the closed loop system

3.8 Useful Matlab commands

```
tf , roots , rlocus , sgrid , step, conv (Hint: to add zeros and poles), rlocfind
```