

MAE 143B - Homework 7

6.7

Multiplying the first ODE by m_u and subtracting the product of the second ODE with m_s , we get

$$\begin{aligned} m_s m_u (\ddot{x}_s - \ddot{x}_i) + m_u b_s (\dot{x}_s - \dot{x}_u) + m_u k_s (x_s - x_u) + m_s b_s (\dot{x}_s - \dot{x}_u) + m_s k_s (x_s - x_u) - \\ m_s b_u (\dot{x}_u - \dot{y}) - m_s k_u (x_u - y) = \\ m_s m_u \ddot{x} + b_s (m_u + m_s) \dot{x} + k_s (m_u + m_s) x - m_s b_u \dot{z} - m_s k_u z = 0. \end{aligned}$$

Similarly, we can subtract $k_u y + b_u \dot{y} + m_u \ddot{y}$ from the second ODE to get

$$\begin{aligned} m_u (\ddot{x}_u - \ddot{y}) + (b_u + b_s) \dot{x}_u + (k_u + k_s) x_u - b_s \dot{x}_s - k_s x_s - k_u y - b_u \dot{y} = \\ m_u \ddot{z} + b_u \dot{z} + k_u z - b_s \dot{x} - k_s x = -m_u \ddot{y} \end{aligned}$$

6.8

We can find the transfer function by first setting up a state-space model with input $\ddot{y}$ and output $x + z$. Notice here that we can choose $\ddot{y}$ as input without affecting the poles since $\ddot{y}$ is just the second time-derivative of y , resulting in two extra zeros at the origin but no extra poles for the transfer function from y to $x + z$. Given the two coupled second-order ODEs, our state needs to have four elements, which we choose as $x_1 = x$, $x_2 = \dot{x}$, $x_3 = z$ and $x_4 = \dot{z}$. The state-space matrices for this model are then

$$\begin{aligned} A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k_s(m_s + m_u)/(m_s m_u) & -b_s(m_s + m_u)/(m_s m_u) & k_u/m_u & b_u/m_u \\ 0 & 0 & 0 & 1 \\ k_s/m_u & b_s/m_u & -k_u/m_u & -b_u/m_u \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} \\ C = [1 \quad 0 \quad 1 \quad 0], \quad D = 0. \end{aligned}$$

We can now choose a set of parameters and compute the poles and thus damping and natural frequencies of the system using MATLAB. The following code computes the natural frequency and damping ratio of the dominant poles for the initial guess $k_s = 16000$ N/m and $b_s = 1600$ Ns/m, respectively.

```
ms = 600;
mu = 40;
ku = 200000;
bu = 0;

ks = 16000;
bs = 1600;

A = [0 1 0 0;
     -ks*(ms+mu)/(ms*mu), -bs*(ms+mu)/(ms*mu), ku/mu, bu/mu;
     0 0 0 1;
     ks/mu, bs/mu, -ku/mu, -bu/mu];
B = [0,0,0,-1]';
C = [1,0,1,0];
D = 0;
```

```
sys = ss(A,B,C,D);
[Wn, Z] = damp(sys);
```

```
Wn/2/pi
Z
```

Using this initial guess, we get a damping ratio $\zeta \approx 0.232$ and natural frequency $f_n \approx 0.8$ Hz for the dominant poles. After a few iterations, we get $\zeta \approx 0.08$ and $f_n \approx 2.5$ Hz for the parameters $k_s = 464000$ N/m and $b_s = 22400$ Ns/m. The resulting poles are located at $p_1 \approx -567.27$, $p_2 \approx -27.54$ and $p_{3/4} \approx -1.26 \pm 15.68j$, respectively.

6.9

Given a constant velocity v , it takes the car $T = w/v$ to pass the pothole, which corresponds to a road profile signal $y(t) = d1(t) - d1(t - T)$. By linearity and time-invariance of the system, the output $x + z$ is the difference of two scaled step responses, one of which is delayed by T . The resulting plots for $v = 10$ km/h and $v = 100$ km/h are shown in Figure 1.

For a brief initial period (0.036s), the behavior is the same in both cases. At $T = 0.36$ s, the pothole is passed using the faster velocity, while it takes $T = 0.36$ s to pass it with the faster velocity. After these two times, both plots show damped oscillations with frequency approximately 10Hz^1 . As anticipated, the fourth-order system responds just like a second-order system with the given characteristics, which is caused by the two slow (i.e. dominant) poles being located much closer to the imaginary axis than the remaining two poles. The plots are generated using the following code, which relies on the results from P6.8. You could also use `lsim`, `SIMULINK` or your favorite alternative method provided your plots look like Figure 1.

```
% b)
d = 0.05;
w = 1;

v = 10/3.6;
T = w/v;

tf_y2xz = tf(sys)*tf([1 0 0],1);

tvec      = 0:T/100:2;

[stp1,~] = step(tf_y2xz,tvec);
stp2      = [zeros(100,1) ; stp1(1:end-100)];

xz_out    = d*( stp1 - stp2 );

figure;
subplot(1,2,1);
plot(tvec,xz_out,'LineWidth',2)
grid on;
xlabel('Time, in s')
ylabel('x+z, in m')
title('v = 10 km/h')

v = 100/3.6;
T = w/v;

tf_y2xz = tf(sys)*tf([1 0 0],1);
```

¹Of course, the frequency is slightly lower than 10Hz following the discussion on p.140 of the reader.

```

tvec      = 0:T/100:2;

[stp1,~] = step(tf_y2xz,tvec);
stp2      = [zeros(100,1) ; stp1(1:end-100)];

xz_out    = d*( stp1 - stp2 );

subplot(1,2,2);
plot(tvec,xz_out,'LineWidth',2)
grid on;
xlabel('Time, in s')
ylabel('x+z, in m')
ylim([-0.05,0.05]);
title('v = 100 km/h')

```

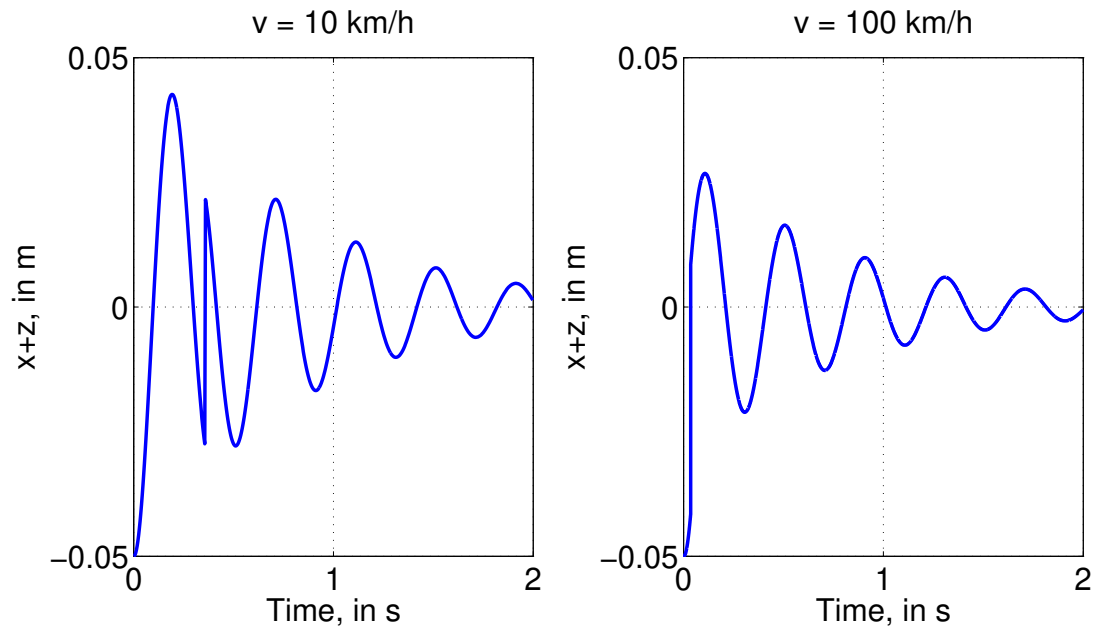


Figure 1: Responses to pothole in P6.9.

6.10

Given what we know about responses of second-order systems and dominance of the slow poles in this example, we would expect the worst-case frequency of a sinusoidal input to be $\omega_d = \omega_n \sqrt{1 - \zeta^2} \approx 15.7$ rad/s. We can use the frequency response of our system to confirm this suspicion. Figure 2 shows a Bode plot of our system. These plots visualize magnitude and phase of our frequency response as functions of frequency ω (you will learn much more about them in the coming weeks). As we can see, the magnitude plot has a peak at approximately 15.7 rad/s, confirming our initial guess. There are no noticeable peaks close to the frequencies corresponding to the remaining two poles because they are close to zeros of the transfer function. Conclusively, the worst-case velocity of our car travelling over this sinusoidal road profile is

$$v = \frac{\lambda \omega_d}{2\pi} \approx 2.5 \lambda / s.$$

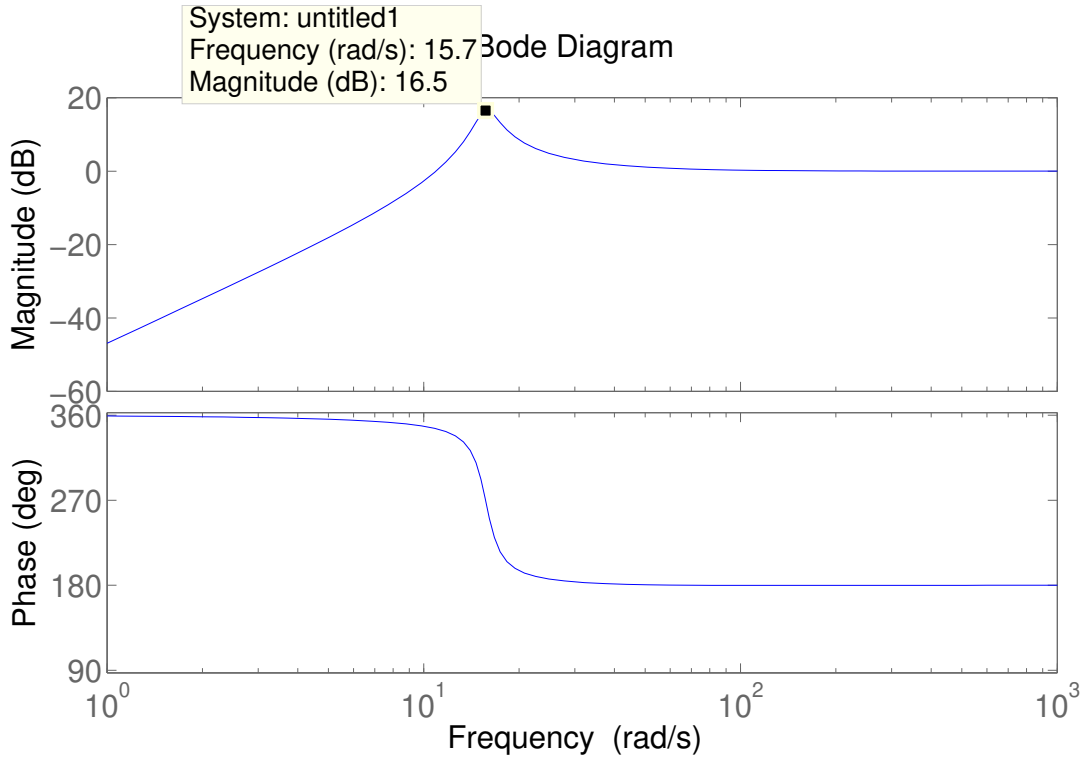


Figure 2: Bode plot for P6.10.

6.16

Taking Laplace transforms, we find the transfer function of our plant as

$$G(s) = \frac{\Omega_2(s)}{Y(s)} = \frac{r_1 \Omega_1(s)}{r_2 T(s)} = \frac{r_1 r_2 / J_r}{s + b_r / J_r}.$$

The transfer function of our controller is $K(s) = K/s$. The open-loop poles are located at $p_1 = -b_r/J_r$ and $p_2 = 0$. There are no open-loop zeros. Given our experience with root-locus plots, we know that if we increase $K > 0$, the closed-loop poles start moving off the open-loop poles towards each other before branching out from the real axis with angles $\phi_0 = \pi/2$ and $\phi_1 = 3\pi/2$. The poles branch out at $c = -b_r/(2J_r)$, which constitutes the design point specified in the problem statement. The closed-loop characteristic polynomial is

$$p(s) = s^2 + \frac{b_r}{J_r}s + \frac{r_1 r_2}{J_r}K$$

with roots

$$s_{1/2} = -\frac{b_r}{2J_r} \pm \sqrt{\frac{b_r^2}{4J_r^2} - \frac{r_1 r_2}{J_r}K},$$

such that $s_{1/2} = -b_r/(2J_r)$ for

$$K = \frac{b_r^2}{4J_r r_1 r_2} = 12.78.$$

The following code computes K and gives you the root-locus plot in Figure 3. The point marked in the plot is very close to the real axis with gain approximately 12.8, confirming our computation above. Given that the controller $K(s)$ has a pole at the origin, the closed-loop system can track a constant reference input $\bar{\omega}_2$ (see p.81 in the reader).

```

clear all, close all;

% P 6.16
r1 = 0.05;
r2 = 0.25;
m1 = 1;
m2 = 10;
b1 = 0.125;
b2 = 6.25;

J1 = m1*r1^2/2;
J2 = m2*r2^2/2;

Jr = J1*r2^2 + J2*r1^2;
br = b1*r2^2 + b2*r1^2;

K = br^2/(4*Jr*r1*r2);

G = tf(r1*r2/Jr,[1 br/Jr]);
C = tf(1,[1 0]);

rlocus(C*G);

```

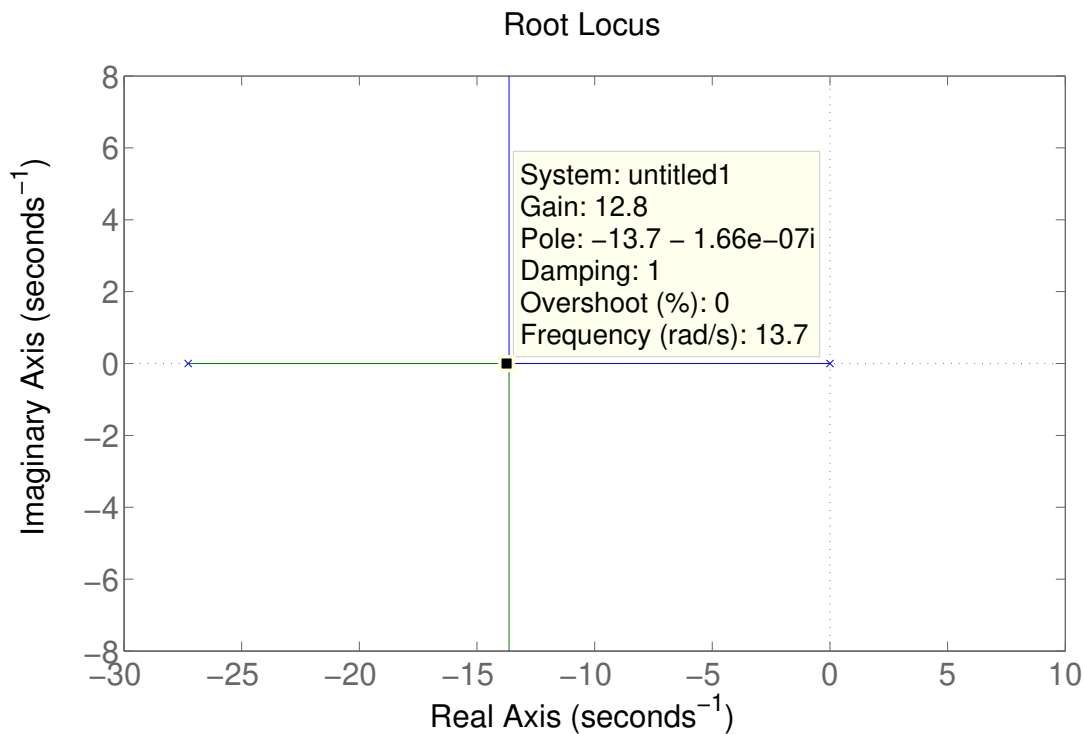


Figure 3: Root-locus plot for P6.16.

6.17

Using the same transfer function for our plant as in P6.16, our controller now has transfer function $K(s) = K_p + K_i/s$. Clearly, our closed-loop system will still be able to track a constant reference input $\bar{\omega}_2$ as long as $K_i \neq 0$. However, we have to proceed in a different way to choose our control gains K_p and K_i to meet the given design specifications.

The open-loop poles are independent of the control gains and located at $p_0 = -b_r/J_r$ (machine pole) and $p = 0$ (pole of controller). Moreover, the open-loop system has a real-valued zero located at $z = -K_i/K_p$. We can choose the location of this zero and thus fix the ratio of our control gains, leaving us with only a single degree of freedom for the remaining design process, which in turn allows us to draw a root-locus plot. That is, we get a root-locus plot with the two real-valued open-loop poles and a real-valued zero that we can choose anywhere on the real axis.

For the remainder of the problem, we also assume $K_i, K_p > 0$.² Under this assumption, we use our rule about the real axis in root-locus plots to conclude that we only have a chance to meet the design specifications if the open-loop zero has real part smaller than the two poles. This is because we are not allowed to cancel the machine pole using the zero and for all greater values of the zero with negative feedback, the open-loop pole at the origin will drift into the right-half plane instantaneously.

Choosing a zero with real part smaller than the machine pole, we are guaranteed to meet the given design criteria for all sufficiently large control gains satisfying the ratio satisfied by our placement of the zero. For instance, Figure 4 shows a root locus plot with $z = -K_i/K_p = -2b_r/J_r$ in terms of K_p . One possible choice for the remaining gain K_p is marked with $K_p \approx 10.9$, which in turn yields

$$K_i = \frac{2b_r}{J_r} K_p \approx 595.55.$$

Of course, there are many other options for your control gains, which are perfectly fine as long as the design criteria are met.

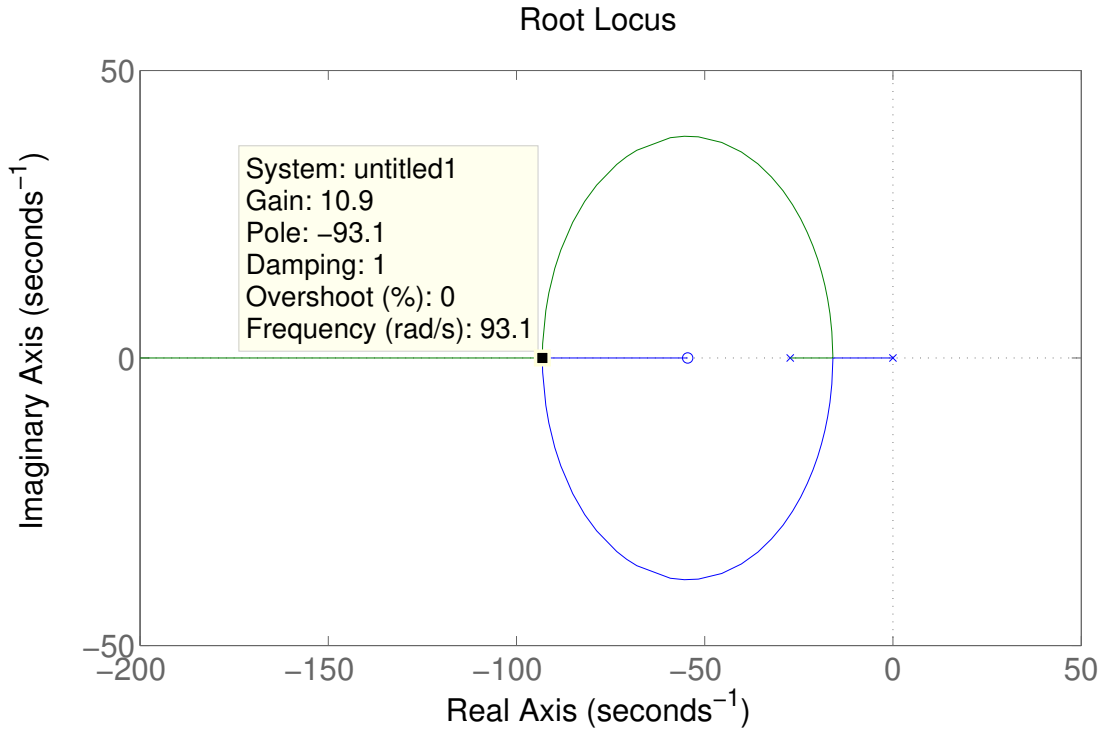


Figure 4: Root-locus plot for P6.17 with $K_i/K_p = 2b_r/J_r$.

²Without this assumption, we would end up with positive instead of negative feedback in the closed-loop system. In terms of the root-locus discussion in the reader, this means $\alpha < 0$, inverting your real-axis rule for drawing root-locus plots.