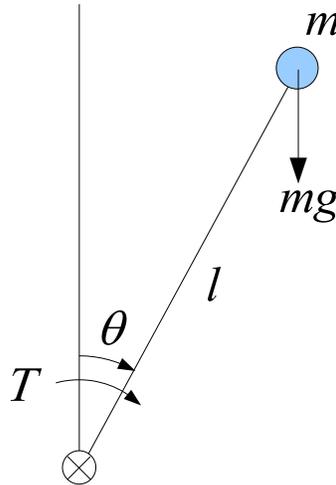


MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Mechanical Engineering
2.04A Systems and Controls
Spring 2013

Supplement to Lecture 18
Control of an inverted pendulum

- a) Derive the inverted pendulum's equation of motion; then linearize the equation that you derived by assuming that the angle θ is very small ($\theta \ll 1\text{rad}$).

Answer: From the following free body diagram, the equation of motion can be found by torque balance between the torque due to gravity, $mg \times l$; inertial torque $ml^2 \times \ddot{\theta}$; and the externally applied torque $T(t)$. We find



$$ml^2\ddot{\theta}(t) = mgl \sin \theta(t) + T(t). \\ \Rightarrow \ddot{\theta}(t) - \frac{g}{l} \sin \theta(t) = \frac{T(t)}{ml}.$$

Approximating $\sin \theta(t) \approx \theta(t)$, we find the linearized equation of motion as

$$\Rightarrow \ddot{\theta}(t) - \frac{g}{l}\theta(t) = \frac{T(t)}{ml}.$$

- b) With the angle θ as output and torque T as input, and using numerical values $l = 1.09\text{m}$, $m = 0.8417\text{kg}$, derive the transfer function of the inverted pendulum system.

Answer: Laplace transforming, we obtain

$$s^2\Theta(s) - \frac{g}{l}\Theta(s) = \frac{T(s)}{ml^2} \Rightarrow \frac{\Theta(s)}{T(s)} = \frac{1/ml^2}{s^2 - g/l}.$$

Using the given numerical values and $g = 9.81(m/s^2)$, we find the transfer function as

$$\frac{\Theta(s)}{T(s)} = \frac{1}{s^2 - 9} = \frac{1}{(s + 3)(s - 3)}.$$

Note that these questions are identical to Practice Problem Set 1, Problem 3, except here we are neglecting dissipation b .

- c) Deduce from the transfer function that the inverted pendulum system is *unstable*. Describe an experiment you could do with everyday objects to verify this result.

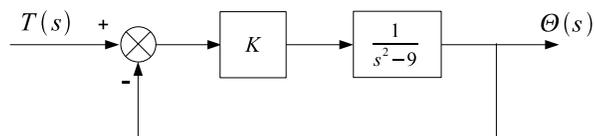
Answer: The transfer function has a pole in the right-hand plane, thus the system is unstable.

Just like the example during lectures, stabilizing a stick so that it stands vertically without falling is a good example of an unstable system. The point is that if the stick were to stand perfectly vertical, it would not fall; but the slightest angular perturbation away from the vertical position (e.g. due to wind or slight motion of your hand, etc.) will grow rapidly causing the stick to fall from its vertical position. Because of its extreme sensitivity to perturbation the inverted pendulum is the archetypal unstable system.

Notice that if we tried a longer stick, the unstable pole $+\sqrt{g/l}$ would move *closer* to the real axis, *i.e.* the time constant would *increase*. This means that the transient response become slower and easier to control! Try this with two pens, one longer than the other and you will be convinced.

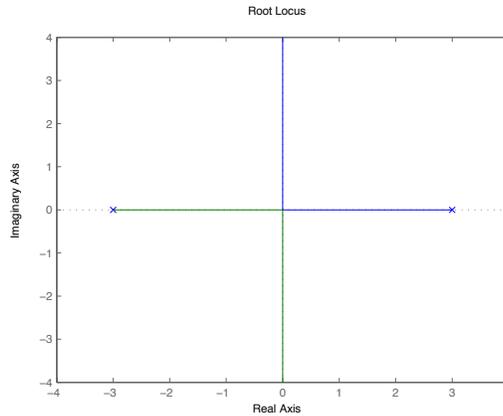
- d) We will now perform a sequence of attempts to stabilize and control the inverted pendulum. First we consider proportional (P) control. Draw the block diagram of the proportional control system, indicating clearly and thoroughly the signals flowing over the arrows and the transfer functions of the blocks in your diagram.

Answer: The block diagram with P controller is



- e) Using the root locus technique, argue that with proportional control we cannot stabilize this system; at best, we can make it marginally stable.

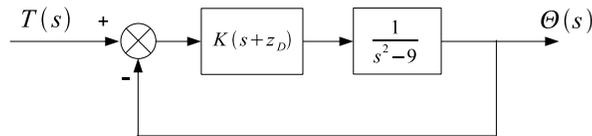
Answer: The root locus of the system is as follows:



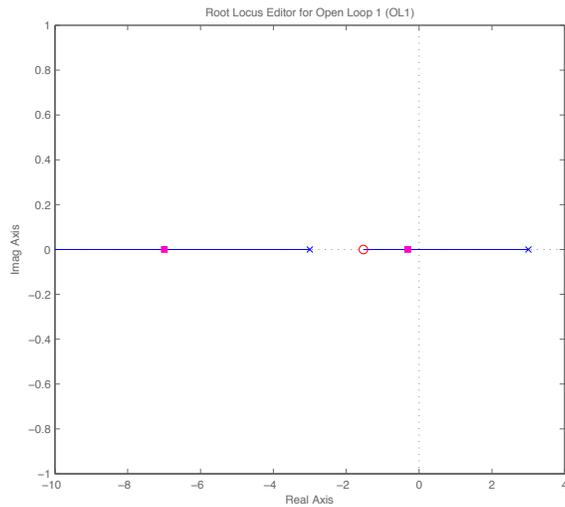
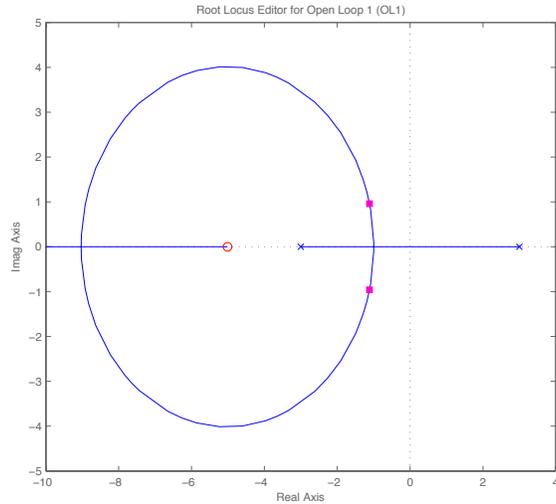
As the gain K increases, the closed-loop poles start moving from $s = \pm 3$ along the real axis. After they meet at $s = 0$, they start moving along the imaginary axis. When the poles are on the real axis, the system is unstable. When the poles are on the imaginary axis, the system is marginally stable. Thus this system cannot be stabilized with a proportional (P) controller.

- f) Next we consider proportional-derivative (PD) control. Draw the modified block diagram and root locus, and argue that the inverted pendulum system can now be stabilized with appropriate choice of gain.

Answer: The block diagram with PD controller.



To move the closed-loop poles into the left-hand plane, we need want to add a zero in left-hand plane. There are two possibilities:



The zero can be farther than $s = -3$ or in between $s = 0$ and $s = -3$. Either one can stabilize the system with a big gain. The second configuration (the zero in between the two poles) will generally need a much bigger gain to stabilize. So the first one (zero to the left of the stable pole) is more practical choice (and agrees with our general strategy for PD control that we saw earlier in this class.)

- g) What is the additional information that the PD controller accounts for to succeed in stabilizing the system where the P controller failed?

Answer: The additional information is velocity. In the PD controller, “D” stands for derivative action. The error signal in the feedback loop (reference - output) contains velocity as well as displacement. Therefore the controller sees how big the error is *and also* how fast the error changes (the *rate* of error growth or decay.)

To understand why this is useful, consider the following: suppose that the error is big, but decreasing due to previous control input, then you can afford to use smaller control input because you don't want to generate huge overshoot. Conversely, if the error is increasing because of a disturbance, the PD controller will detect the fast increase as well as the rate that the error is growing and succeed in compensating the error much better than the proportional (P) controller that take into account the error alone.

- h) Let us now design the PD controller for the following transient specifications: peak time $T_p = 1.61\text{sec}$ and overshoot 20%. Interpret these specifications as the locations of the desired closed-loop dominant poles on the s -plane. What is the natural frequency of the desired (compensated) system?

Answer: From $T_p = 1.61(\text{sec})$ and 20% OS specification,

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} = 0.4559, \quad \text{and}$$

$$T_p = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} = 1.61(\text{s}).$$

Thus, the natural frequency $\omega_n = 2.1924(\text{rad/s})$, the damped natural frequency $\omega_d = \omega_n \sqrt{1 - \zeta^2} = 1.9513(\text{rad/s})$, and $\sigma = \zeta\omega_n = 0.9766$. The desired pair of poles are $s = -\sigma \pm j\omega_d = -0.9766 \pm j1.9513$.

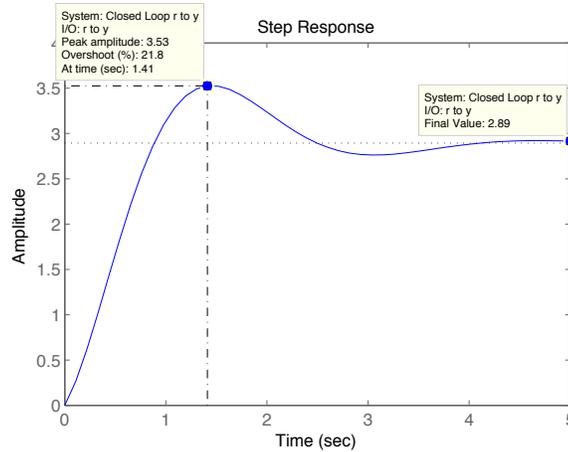
- i) Find the location of the PD controller's zero and gain K that meet our transient specifications.

Answer: There is a systematic way of carrying out this design, and it is described in detail in the textbook, section 9.3. We will skip this geometrical derivation (however, if you are seriously interested in control as subject for further study, then you should pay more attention to it; and you will see it again in more advanced control electives.) We will use MATLAB to carry out the design. Note, however, that in MATLAB we should define the plant transfer function (two real poles at $s = \pm 3$) separately from the controller transfer function. To avoid confusion with the gain, change the following option in the "Control and Estimation Tools Manager" : in the menu `Edit`→`SISO Tool Preferences...`, select the tab `Options` and under `Tuned parameters` select the `Zero/Pole/Gain` option. This way, MATLAB will display the controller's transfer function in a manner compatible to our notation in class.

With these steps, we find the approximate zero location as $z_D \approx 7$ and closed-loop gain $K \approx 2$ (MATLAB gives the more accurate value 1.9519.)

- j) Now get MATLAB to display the step response of the PD-compensated inverted pendulum. Explain why the overshoot is slightly higher and the peak time slightly shorter than our design targets.

Answer: From MATLAB's `sisotool`, the step response is as follows:



Note that $T_p = 1.41(s)$ and %OS is 21.8%. This is because the damping ratio ζ and natural frequency ω_n are computed by approximation of the actual system as a 2nd order system whose poles are equal to the dominant poles of the actual system. The actual system has an additional zero due to the PD controller; therefore, we expect the exact response to be slightly different: derivative action will cause the peak time to become slightly shorter (faster) than the approximated system, and the overshoot to become slightly larger than the approximated system.

- k) What is the steady-state error? Calculate it directly from the closed-loop transfer function of your PD-compensated inverted pendulum and then verify your result using MATLAB.

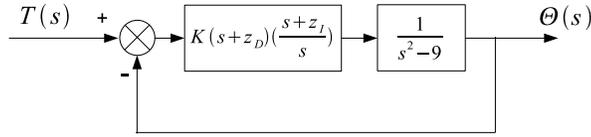
Answer: The steady state value θ_{ss} can be found as follows:

$$\theta(\infty) = \lim_{s \rightarrow 0} s \left(\frac{1}{s} \right) \frac{KG(s)}{1 + KG(s)} = \frac{7.0455K}{-9 + 7.0455K} = 2.8939,$$

where $G(s) = \frac{s + 7.0455}{s^2 - 9}$ and $K = 1.9519$. Therefore, the steady state error $e(\infty) = 1 - y(\infty) = -1.8939$ or 189.39%. Notice that the exact step response found from MATLAB in question (k) agrees well with our estimate.

- l) As a final effort to correct this system, let us attempt to eliminate the steady-state error by PID compensation without degrading the transient, if at all possible. We maintain the same derivative term that we found from the previous questions (since the desired transient has already been achieved) and cascade an integral controller. Use a safety factor of 10 for the relative size of the integral controller's zero relative to the plant's stable pole. Verify with MATLAB that indeed the PID controller has eliminated the steady-state error.

Answer: We add a PI controller. The block diagram is as follows:



Note that the transfer function of the PI controller is $\frac{s+z_I}{s}$ not pure integrator. Since the question requires to use a safety factor of 10, $z_I = 0.3$.

The open-loop transfer function is now

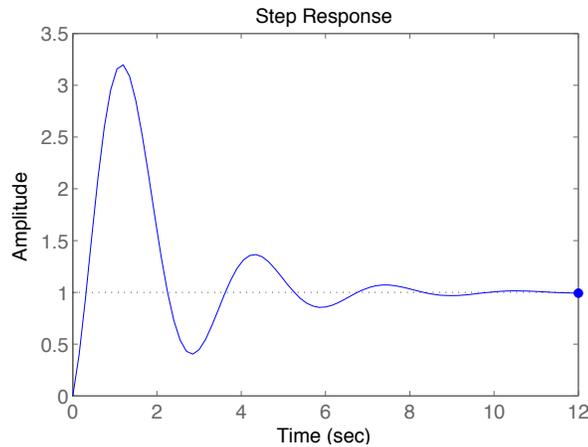
$$K(s+z_D) \frac{(s+z_I)}{s} \frac{1}{s^2-9}.$$

This system includes one free integrator, so the steady state error to a step input is zero.

We can confirm that by computing the steady state output as

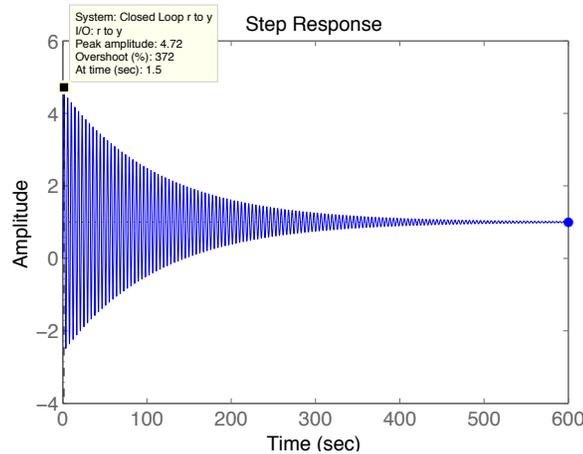
$$y(\infty) = \lim_{s \rightarrow 0} s \left(\frac{1}{s} \right) \frac{K(s+z_D) \frac{(s+z_I)}{s} \frac{1}{s^2-9}}{1 + K(s+z_D) \frac{(s+z_I)}{s} \frac{1}{s^2-9}} = 1.$$

Therefore, the steady-state error is indeed $e(\infty) = 1 - y(\infty) = 0$.



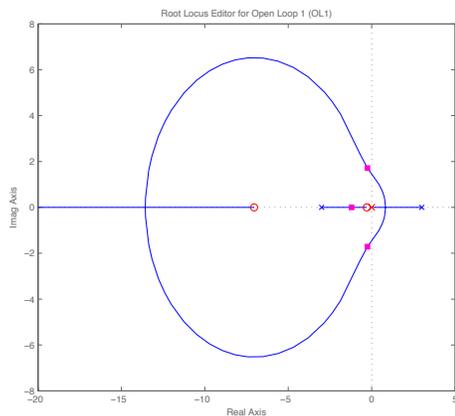
- m) Now use MATLAB to adjust the gain such that the peak time requirement is met. What goes horribly wrong when you do that? This seems to suggest that the PID controller does not work as advertised in this case, *i.e.* it does not leave the transient response unchanged. Can you explain why this system is different than the PID control example that we saw in class?

Answer: At the gain $K = 1.52$, the step response is as follows:

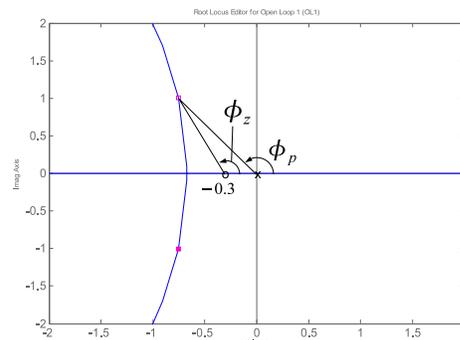


As we try to get the desired peak time as decreasing gain K , the overshoot becomes huge. Also when we decrease the gain more, then the system becomes unstable.

The reason for the excessive overshoot in the PID-compensated system is as follows: We note that the breakaway point of the PD-compensated system (before cascading the PI compensator) is at a very small negative value. This means that the root locus passes very close to the origin. Therefore, the usual PI controller assumption that the PI zero-pole pair does not have a significant total angular contribution to the (uncompensated) root locus is no longer true. Compare the schematics below.



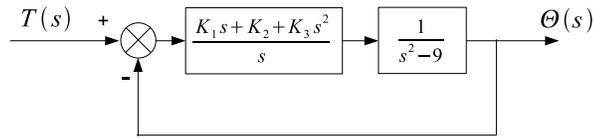
PID-compensated root locus



Zoom-in on the PD-compensated root locus (from question f) near the origin. Clearly, $\phi_z \neq \phi_p$.

- n) Suggest a strategy to get around this problem with the same PID controller but different gains, now interpreting the peak time and overshoot requirements as upper bounds: “achieve peak time *less than* $T_{p,\max} = 1.61\text{sec}$ and overshoot *no more than* 20%.” Calculate the final PID controller’s gains K_1 , K_2 , K_3 (notation same as in the Nise textbook and Lecture 33).

Answer: Now we control the individual gains of the P , I , and D components as follows:



The transfer function of the PID controller is

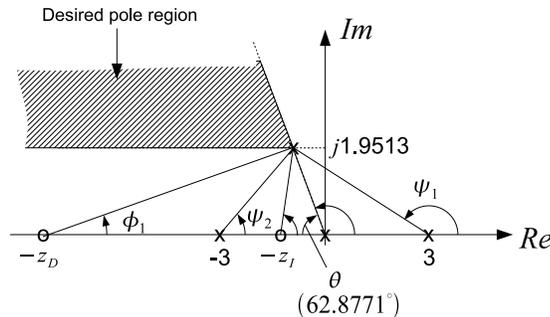
$$\frac{K_1s + K_2 + K_3s^2}{s}.$$

This is very similar to the approach that we took to control the 2.04A Tower in the lab last week.

From the new requirement, we obtain the following relations:

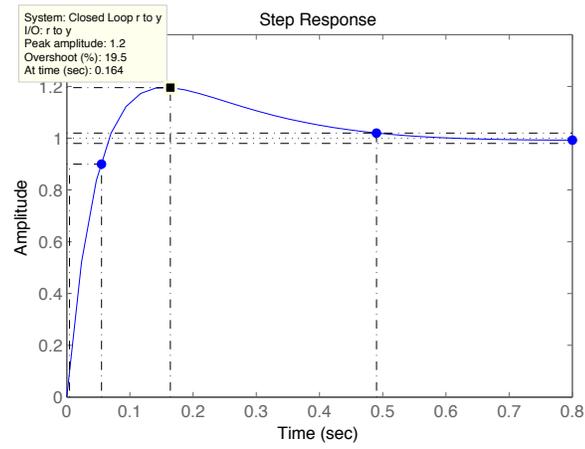
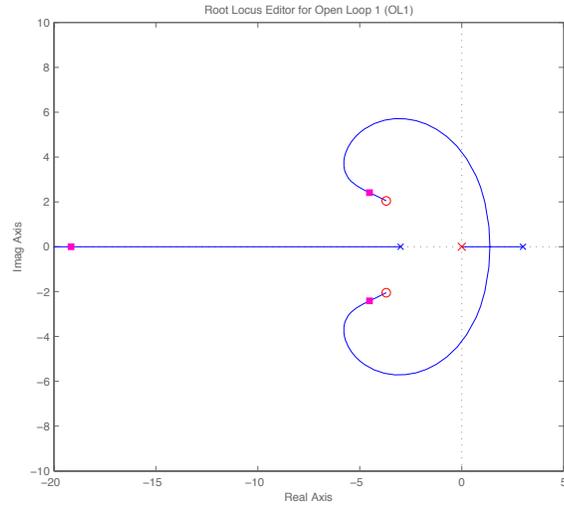
- $\omega_d > 1.9513$.
- $0.2 > \exp\left\{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right\} \Rightarrow \zeta > 0.4559$ ($\cos\theta < 0.4559 \Rightarrow \theta < 62.8771^\circ$).

The desired pole location can be drawn as follows:



All we need to do in this problem is to find K_1 , K_2 , and K_3 that make the (dominant) closed-loop poles to be in the desired region. MATLAB allows us to change K , z_D and z_I in the previous questions.

Alternatively, since we can change individual gains of K_1 , K_2 , and K_3 , we want to introduce another trick that you can play with. The transfer function of the compensator has a second-order numerator, so we can actually use a pair of complex zeros as controller. (Many of you used that approach last week for the 2.04A Tower.) For example, when we add a pair of zeros at $s = -3.7 \pm j2.04$ (and a pole at $s = 0$ of course), then you can satisfy the desired step response. The resulting gains for this particular case are $K_3 = 28.208$, $K_2 = 17.9$, and $K_1 = 7.4$; and the root locus and step response are shown below.



The step response meets the transient requirements rather well.

MIT OpenCourseWare
<http://ocw.mit.edu>

2.04A Systems and Controls
Spring 2013

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.