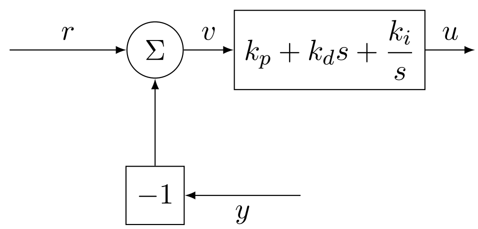
AE353 Homework #8:
PID and Time Delay

(due at the beginning of class on Friday, April 17)

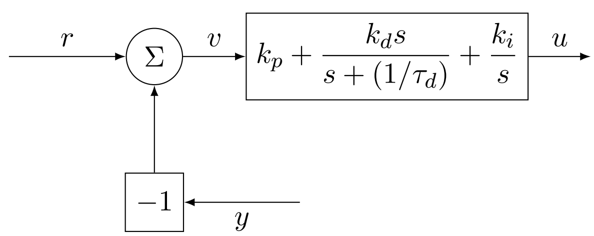
1. In this problem, you will look at three different ways to implement a PID controller. Please do all of the analysis by hand. (Don't worry, I checked this time — none of it is painful.)

   (a) Here is one way to implement a PID controller:

   Why is it bad? Look at the frequency response. Denote the transfer function from y to u by $T_{yu}(s)$. Compute $|T_{yu}(j\omega)|$, assuming that $k_p, k_d, k_i > 0$. In particular, compute $|T_{yu}(j\omega)|$ as $\omega \to \infty$. What do you conclude will happen if there is high-frequency measurement noise?

   (b) Here is another way to implement a PID controller:

   - Why is it good? Look at the frequency response. Denote the transfer function from y to u by $T_{yu}(s)$. Compute $|T_{yu}(j\omega)|$, assuming that $k_p, k_d, k_i, \tau_d > 0$. In particular, compute $|T_{yu}(j\omega)|$ as $\omega \to \infty$. What do you conclude will happen if there is high-frequency measurement noise? How is this better than before?

     To really understand what is happening, please also draw a Bode plot (both gain and phase) of the following two transfer functions on the same figure, for $\tau_d = 0.1$:

     $$T_{\text{derivative}}(s) = s \qquad\qquad T_{\text{approximate-derivative}}(s) = \frac{s}{s + (1/\tau_d)}$$

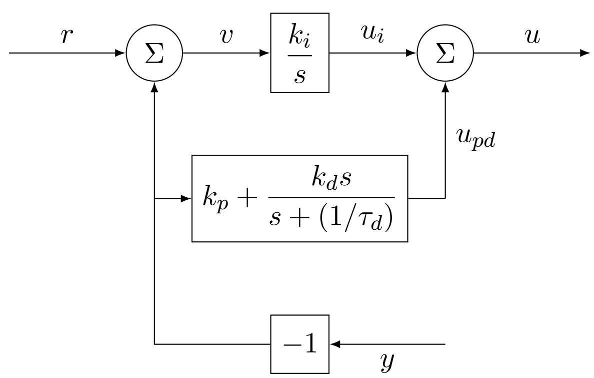
     What do you notice?

   - Why is it bad? Look at the time response. Denote the transfer function from v to u by $T_{vu}(s)$. Compute an equivalent state-space system—with input v, output u, and state z—in controllable canonical form, assuming that $k_p, k_d, k_i, \tau_d > 0$. Suppose that

     $$\left.\begin{aligned} z(t) &= 0 \\ v(t) &= 0 \end{aligned}\right\} \text{ for all } t < 0$$

and that $v(0) = 1$. What was $u(t)$ for all $t < 0$, and what is $u(0)$? What do you conclude about how the controller responds to a step change in the reference signal? Why might this behavior be undesirable?

(c) Here is yet another way to implement a PID controller:



Why is it good?

- Look at the frequency response. Denote the transfer function from y to u by $T_{yu}(s)$. Compute $|T_{yu}(j\omega)|$, assuming that $k_p, k_d, k_i, \tau_d > 0$. In particular, compute $|T_{yu}(j\omega)|$ as $\omega \to \infty$. What do you conclude will happen if there is high-frequency measurement noise? Is this behavior similar to what you found in part (b) or in part (c)? In other words, is it bad behavior or good behavior?

- Look at the time response. Assume that $k_p, k_d, k_i, \tau_d > 0$. Compute a state-space system that is equivalent to the entire block diagram, with inputs r and y and with output u. I recommend that you do this in three steps:
  - compute a state-space system in controllable canonical form that is equivalent to the transfer function from y to u_{pd};
  - compute a state-space system in controllable canonical form that is equivalent to the transfer function from v to u_i;
  - combine these two systems.

  Suppose that

$$\left.\begin{aligned} z(t) &= 0 \\ r(t) &= 0 \\ y(t) &= 0 \end{aligned}\right\} \text{ for all } t < 0$$

  and that $r(0) = 1$. What was $u(t)$ for all $t < 0$, and what is $u(0)$? What do you conclude about how the controller responds to a step change in the reference signal? How is this better than before?

2. You have seen that the rotational motion of a control moment gyro on a spacecraft can be described in state-space form as

$$\begin{aligned} \dot{x} &= -x + u \\ y &= x \end{aligned}$$

where x is angular velocity and u is applied torque. In this problem, you will look at the effect of time delay on control of this system.

(a) Find the transfer function $H(s)$ from u to y.

(b) Suppose:

$$y_{\text{delayed}}(t) = y(t - \tau_{\text{delay}})$$

Then, as we have seen in class, the transfer function from y to y_{delayed} is:

$$T_{\text{delay}}(s) = e^{-s\tau_{\text{delay}}}$$

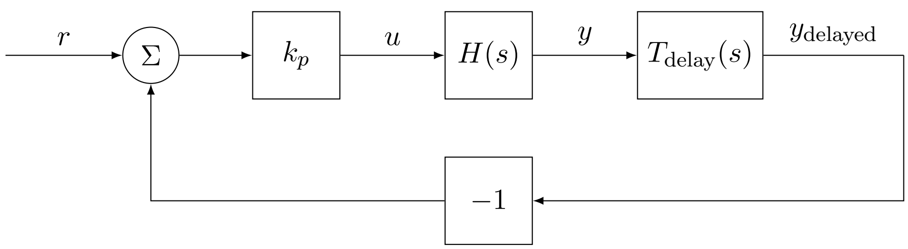
Find the gain $|T_{\text{delay}}(j\omega)|$ and phase $\angle T_{\text{delay}}(j\omega)$ of this transfer function, in terms of ω and τ_{delay}. Why would it be correct to say that time delay causes a "pure phase shift?"

(c) In what follows, it will be convenient to apply the first-order Padé approximation

$$T_{\text{delay}}(s) \approx \frac{1 - s(\tau_{\text{delay}}/2)}{1 + s(\tau_{\text{delay}}/2)}$$

The reason this approximation is useful is that it turns an exponential (which we don't know how to deal with) into a ratio of two polynomials (which we do). To understand the nature of this approximation, please draw a Bode plot (gain and phase) of both the exact and approximate T_{delay} on the same figure, in the particular case $\tau_{\text{delay}} = 2$.

(d) Suppose proportional control (just the "P" in "PID") is applied to this system:



Compute the closed-loop transfer function from r to y, using your answer from part (a) and using the Padé approximation for $T_{\text{delay}}(s)$.

(e) Suppose $k_p = 4$. What is the maximum τ_{delay} for which the closed-loop system is stable?

(f) Consider the "loop transfer function"

$$L(s) = k_p H(s) T_{\text{delay}}(s)$$

in the absence of time delay, i.e., for $\tau_{\text{delay}} = 0$. Compute the crossover frequency of this transfer function, i.e., the frequency ω_c at which

$$|L(j\omega_c)| = 1$$

Compute the phase difference

$$\angle L(j\omega_c) - (-180°)$$

at this crossover frequency (expressed in degrees). Finally, compute the phase shift

$$\angle T_{\text{delay}}(j\omega_c) = \angle e^{-j\omega_c \tau_{\text{delay}}}$$

(expressed in degrees) that would be applied to the loop transfer function by the time delay you found in part (e). What do you notice? Wow!

Congratulations, you have reinvented the concept of "phase margin" and have seen why it matters. The fact that frequency domain methods (analysis of Nyquist and Bode plots) make it easy to reason about the effects of delay are a large part of their appeal.

3. You have seen that the motion of a robot arm with one revolute joint can be described in state-space form as

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -1/5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1/5 \end{bmatrix} u$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

where the state elements are the angle (x_1) and angular velocity (x_2) of the joint, the input u is a torque applied to the arm at the joint, and the output y is a measurement of the angle. Suppose we want to apply PID control in the form that was discussed in Problem 1(c). Suppose the performance specifications are as follows:

- 2% settling time less than 1.0 seconds for a step change in the reference signal r;
- peak overshoot less than 15% for a step change in the reference signal r;
- no steady-state error for a step change in the disturbance input d;
- gain less than 10 for sinusoidal measurement noise n at all frequencies above 20Hz.

Note that 1 Hz $= 2\pi$ rad/s.

(a) Please choose k_p, k_d, k_i and τ_d to satisfy these performance specifications. Please begin by drawing a block diagram of your closed-loop system, then proceed with design by trial-and-error, through simulation with MATLAB.

(b) Please implement your PID controller in `ControlLoop` in `hw7prob03.m`. I suggest that you do so by deriving an equivalent state-space representation—just as you did in Problem 1(c)—and then by applying a finite-difference approximation. You can call

```
hw7prob03(rmag,rfreq,dmag,dfreq,nmag,nfreq)
```

from the MATLAB prompt to set the magnitude, frequency, and phase of $r(t)$, $d(t)$, and $n(t)$, all having the form

$$m \cos(\omega t)$$

for magnitude m and frequency ω.

(c) **Extra Credit.** Design an observer and a controller with integral action, however you like, to satisfy the same performance specifications. Compare your design to the PID controller, in any way you feel is appropriate. (Ease of design? Complexity of implementation? Resulting performance?)

(d) **Extra Credit.** Compute the maximum time delay that can be tolerated by the PID controller and by the controller/observer with integral action before the closed-loop system becomes unstable. Verify your results by implementation with `ControlLoop`. You can call

```
hw7prob03(rmag,rfreq,dmag,dfreq,nmag,nfreq,tdelay)
```

to specify a time delay. (Play around with it!)

This problem is open-ended. There is no "right" or "wrong" answer. The point of this problem is to give you a feel for what changes in k_p, k_d, k_i and τ_d do to a closed-loop system. Please submit whatever you think is necessary to justify your design process.