EEENG307: Proportional-Derivative (PD) Control to Improve Transient Response*

Lecture 13

Elenya Grant, Kathryn Johnson, and Hisham Sager†

Fall 2022

Contents

1 Pre-requisite Material

This lecture assumes that the reader is familiar with the following material:

- Lecture 9: Introduction to Control Concepts and More on Stability
- Lecture 12: Time Response of Higher Order Systems

2 Intro to Proportional-Integral-Derivative (PID) Control

In this lecture, we look at the design of a simple feedback control system. The control will be restricted to have specific terms: a proportional gain K_p, an integral with a gain K_I, or a derivative with a gain K_D. (All of these three gains are tunable to achieve desired stability and performance characteristics.) Using the first initials of the three terms, this is often called a PID controller. This PID control works well and is fairly easy to tune using some rules of thumb for systems that are first or second order (or have dominant dynamics that are first or second order). There are many examples of such systems, thus the PID controller is very common.

In *open loop* (i.e. before we attach a controller) the system to be controlled can be modeled using Figure 1.

†Developed and edited by Tyrone Vincent and Kathryn Johnson, Colorado School of Mines, with contributions from Salman Mohagheghi, Chris Coulston, Kevin Moore, CSM and Matt Kupilik, University of Alaska, Anchorage

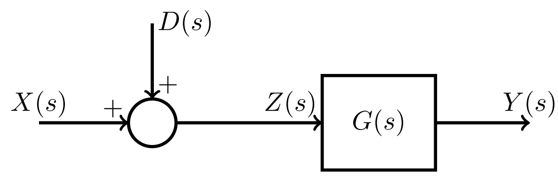


Figure 1: Open loop plant system $G(s)$ disturbed by an external, uncontrollable signal $D(s)$

- $G(s)$ - plant system to be controlled
- $D(s)$ - disturbance signal modeled as input disturbance
- $X(s)$ - actuator command (will become the output signal from the controller)
- $Y(s)$ - plant output to be regulated

Disturbances impact many physical systems. For example, the force due to gravity on the car chassis height modeling example is a disturbance – an uncontrollable force acting on the car that affects its behavior. They can appear in different places around a feedback loop, but the configuration shown in Figure 1 is fairly common.

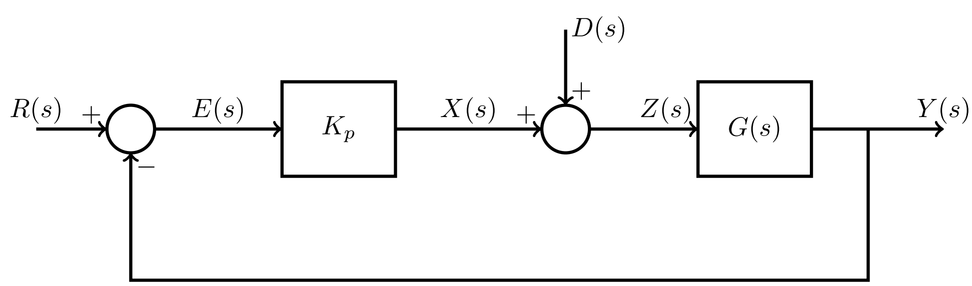
Note that the plant transfer function $G(s) = \frac{Y(s)}{Z(s)}$, where $Z(s)$ is the input to the plant and $Z(s) = X(s) + D(s)$. When modeling systems with disturbances, it's important to calculate your plant's transfer function from $Z(s)$ instead of from $X(s)$ or $D(s)$.

In this article, we will be focused on two parts of the PID controller: the **proportional term with gain** K_p and the **derivative term with gain** K_D. The integral term can be very useful for dealing with steady-state error and disturbance rejection, and we will therefore come back to it after some additional work with those concepts.

3 Proportional Control

As given in the circuit example in Lecture 9: Introduction to Control Concepts and More on Stability, to apply feedback control, the output y is measured and then compared to a desired reference r to give the error signal $e = r - y$. This error is then applied to the control algorithm as the input signal to the controller system (block). If we use a proportional controller (P) as shown in the figure below, the actuator command x is proportional to the error e. The gain of the proportional controller is usually denoted as K_p.

Proportional Control



When $G(s)$ is a first or second order system, the design of the controller can be straightforward.

1. **Collect the design specifications.** Design specifications could be in terms of transient response (rise time, settling time, overshoot), as will be the focus of this lecture, or steady state response (to be discussed in upcoming lectures).

2. **Find the *closed loop* transfer function so that the closed-loop poles can be designed to achieve transient specifications** The closed loop transfer function gives the response of the controlled system, as viewed from

a reference to the output. In future lectures, we will also look at transfer functions from the disturbance to the output and the disturbance and reference to the error. Specifically for this article, using the block diagram simplification rules,

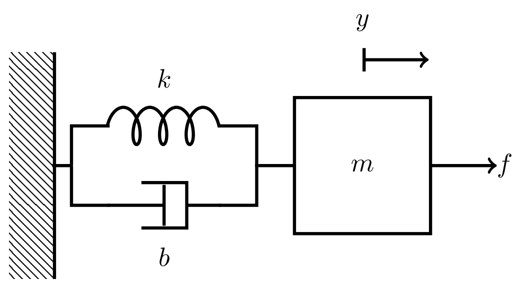
$$\frac{Y(s)}{R(s)} = \frac{K_p G(s)}{1 + K_p G(s)},$$

You will plug the plant's open loop transfer function $G(s)$ into this formula. The reason we need to find the closed-loop transfer function is that our design specifications – rise time, settling time, and percent overshoot – are closed-loop specifications.

3. **Select K_p so that the design specifications are met, or determine that no K_p exists to meet the specifications.**

Example 1. Let's take a simple mass-spring-damper example with input force $f(t)$ acting on the mass and output position $y(t)$ of the mass. Assume no disturbance, i.e., $d(t) = 0$ and thus $x(t) = z(t)$.

Mechanical Proportional Control



Assume the system parameters m, b, and k are such that the plant has transfer function

$$G(s) = \frac{2}{s^2 + 0.5s + 2}.$$

Design a proportional controller to achieve a rise time $t_r \leq 2$ s and a settling time $t_s \leq 10$ s in response to a unit step input.

Step 1: Collect and Convert Specifications

First, convert the specifications to the necessary natural frequency ω_n and damping ratio ζ.

$$t_r \leq 2 \Rightarrow \frac{2.2}{\omega_n} \leq 2 \Rightarrow \omega_n \geq 1.1 \text{ rad/s}$$
$$t_s \leq 10 \Rightarrow \frac{4.6}{\zeta\omega_n} \leq 10 \Rightarrow \zeta\omega_n \geq 0.46$$

Notice that we keep the settling time specification in terms of the *product* $\zeta\omega_n$, not either one alone. The reason for keeping the product is that it can help with specifications given as inequalities, as was described in Lecture 12.

Step 2: Closed-Loop Transfer Function

Then, find the closed-loop transfer function with proportional control in feedback as shown at the beginning of this section.

$$\begin{aligned}\frac{Y(s)}{R(s)} &= \frac{\frac{2K_p}{s^2+0.5s+2}}{1 + \frac{2K_p}{s^2+0.5s+2}} \\ &= \frac{2K_p}{s^2 + 0.5s + (2 + K_p)}\end{aligned}$$

Step 3: Match the Two Transfer Functions

Finally, match the two transfer functions to achieve the specifications, or determine if it can't be done. To meet our inequality specifications, let's choose $\omega_n = 1.5$ rad/s and $\zeta\omega_n = 0.5$. Note: the actual transfer function (with this plant and using proportional control) is the term on the left, and the desired transfer function (the one needed to meet the specifications) is on the right.

$$\underbrace{\frac{2K_p}{s^2 + 0.5s + (2 + K_p)}}_{\text{Actual TF}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

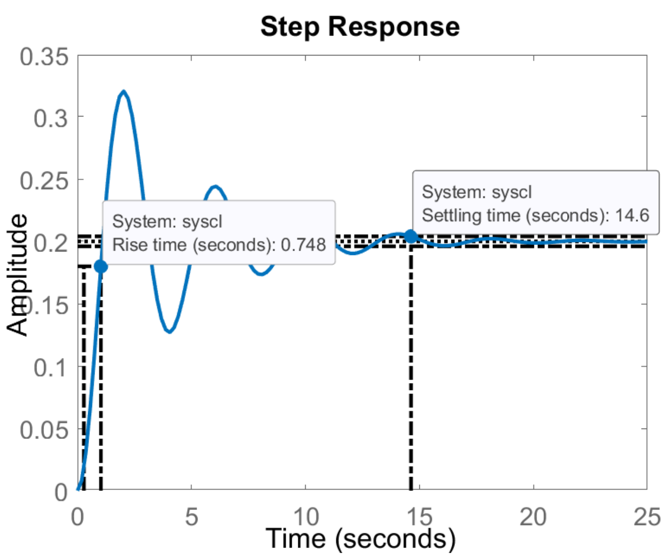
$$= \underbrace{\frac{2.25}{s^2 + s + 2.25}}_{\text{Desired TF}}$$

These can't be matched! Because K_p appears only in the s^0 term of the denominator of the actual transfer function, only the rise time term depending on ω_n can be matched. There is nothing we can do to make the coefficients on s^1 equal, which means the settling time $t_s = \frac{4.6}{\zeta\omega_n}$ depends on the plant alone, not the plant and controller.

Even without the ability to design a controller to impact the settling time, we can still choose and check the results to verify the rise time and see how far off we are for the settling time. Let's try $K_p = 2.25 - 2 = 0.25$ to match the actual and desired transfer functions.

Verification: Check the Results

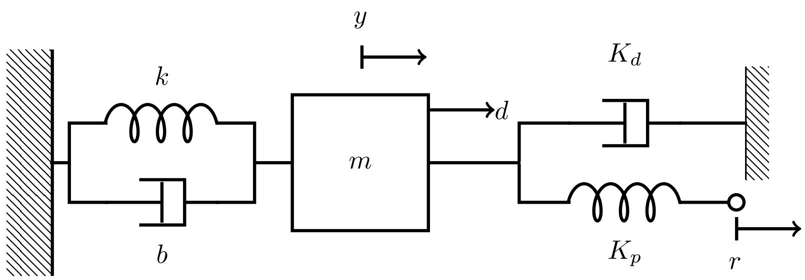
As can be seen in this step response figure plotting the position $y(t)$ of the mass vs. time, the rise time specification has been met at 0.78 s but the settling time specification has not – it's 14.6 s. We'll need to use a different kind of controller to be able to meet both.



4 Proportional/Derivative (PD) Control

As we saw in both Section 3 and Lecture 9: Introduction to Control Concepts and More on Stability, proportional control can sometimes result in large oscillations that may be undesirable. An intuitive fix for the oscillatory response that proportional control can cause is to change our control so that it implements additional damping. This is the same idea as adding damping to a mountain bike or vehicle suspension system. Using mechanical components, we could add a damper between ground and the mass.

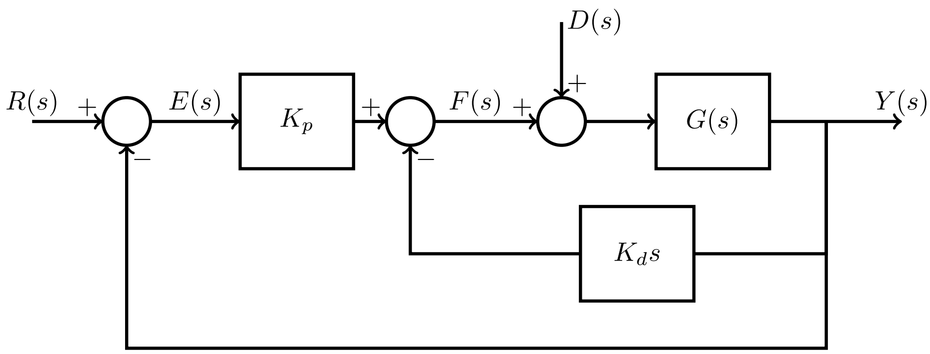
Mechanical Proportional/Derivative Control



The force that is created by this mechanical controller is

$$f(t) = K_p(r(t) - y(t)) - K_d\dot{y}(t).$$

Thus, we are adding a derivative term to our control via $\dot{y}(t)$, which creates a PD controller. This PD controller can be represented using the following block diagram:



The closed loop behavior from the reference input $R(s)$ to the mass's position $Y(s)$ (i.e. considering r as an input and y as an output) is given by

$$\frac{Y(s)}{R(s)} = \frac{K_p/m}{s^2 + ((b + K_d)/m)s + (k + K_p)/m}$$

(Another transfer function would help us relate the disturbance input $D(s)$ to the output $Y(s)$, but we're saving that for a future lecture.) We can see that there are adjustable variables – K_p and K_D – in both the coefficients on s^1 and the one on $s^0 = 1$ of the denominator polynomial, allowing us to adjust *both* damping ratio and natural frequency.

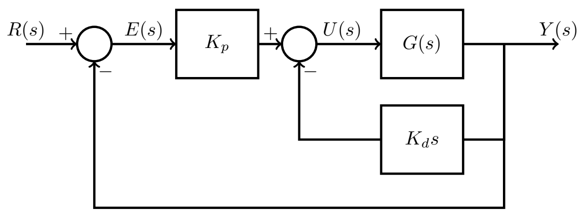
As long as we have the appropriate actuator, this configuration can be used with any second order system, not just mass-spring-damper systems! Note that the design procedure will have the same basic steps as we for proportional control in Section 3: collect the specifications, find the closed loop transfer functions, and then select the gains so that the specifications are met.

Before doing an example, however, let's take a look at two versions of a PD Control configuration.

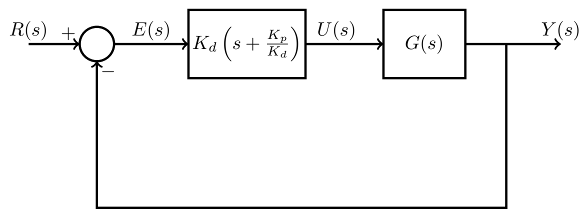
4.1 A Design Format for PD control

Although the configuration shown in the mass-spring-damper example with force damping is derived from physical principles, it's not commonly used in computer-based control. Compare this physically-derived PD configuration, shown on the left, with the more common unity gain feedback configuration on the right, which is often implemented via a computer. (In both cases, the disturbance is assumed to be zero; we will revisit disturbances in an upcoming lecture). Note that both closed-loop systems have the same denominators and therefore the same pole locations, meaning that the theoretical design basis based on pole locations only is identical.

These Two Systems Have the Same Closed Loop Denominator



Physically-Derived System



Design System

The Closed Loop Transfer Functions

$$T(s)=\frac{K_pG(s)}{1+K_d(s+\frac{K_p}{K_d})G(s)}$$

Physically-Derived System

$$T(s)=\frac{K_d\left(s+\frac{K_p}{K_d}\right)G(s)}{1+K_d\left(s+\frac{K_p}{K_d}\right)G(s)}$$

Design System

After factoring out K_d on the physically-derived system, we see that the two denominators are the same, and in both cases the closed loop poles are determined by

$$1+K_d\left(s+\frac{K_p}{K_d}\right)G(s)=0$$

Note that the "design system" has a result similar to the physically-derived control, except it adds an additional zero at $-\frac{K_p}{K_d}$. This zero could cause problems for our design unless it is significantly faster than the dominant closed-loop poles (see Lecture 12), so we always need to verify and tune our design.

4.2 PD Control Design Example

Example 2. Let's revisit the mass-spring-damper example from Section 3 but now with PD control using the "design system" configuration. The main difference is that we will now have an extra tuning parameter, K_D, which makes it easier to meet our specifications.

Using the same plant $G(s)$, the actual closed loop transfer function is now

$$\frac{Y(s)}{R(s)}=\frac{2K_Ds+2K_p}{s^2+(0.5+2K_D)s+2(K_p+1)}$$

Notice that we now have a way to adjust the coefficient on the s^1 term in the denominator: tune the derivative gain K_D. Therefore, matching this actual transfer function to the desired transfer function from Section 3 (Step 3), we get

$$0.5+2K_D=1\Rightarrow K_D=0.25$$

and

$$2(K_p+1)=2.25\Rightarrow K_p=0.125$$

The Simulink model is shown in Figure 2, where the PD controller is implemented using Simulink's "PID Control" block.

Within Simulink's "PID Controller" block, we have defined K_p and K_D as calculated and set the integral gain $K_I=0$, as shown in Figure 3. (We'll talk more about the integral gain in the PID Control Design lecture.)

Finally, we examine the results, which are shown in Figure 4. Assuming the system has reached steady-state at the end of the 20-s simulation time period, we note that the steady-state value of $y(t)$ is 0.1111 m (this value can be found by using the data cursor in the figure). Thus, the rise time is the time it takes to get from 0.0111 m to 0.1 m, which is about $t_r=0.2$ s – clearly within the 2 s specification. The 1% settling time is $t_s=10.8$ s, which is just outside of the specification and shows the importance verifying your results in Simulink! It's likely that the zero's impact on the closed-loop system response caused us to miss our spec. If we were designing this controller for production (or a class project!), we'd want to do some tuning to meet the specification. See the Quiz Yourself problems for further thinking on the topic.

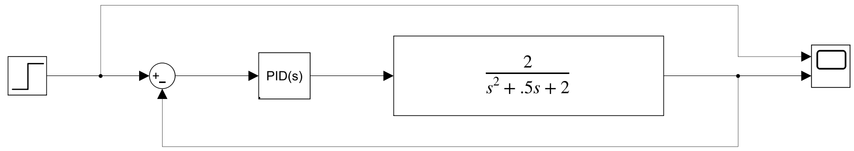


Figure 2: PD controller design implementation in Simulink using the "PID Controller" block.

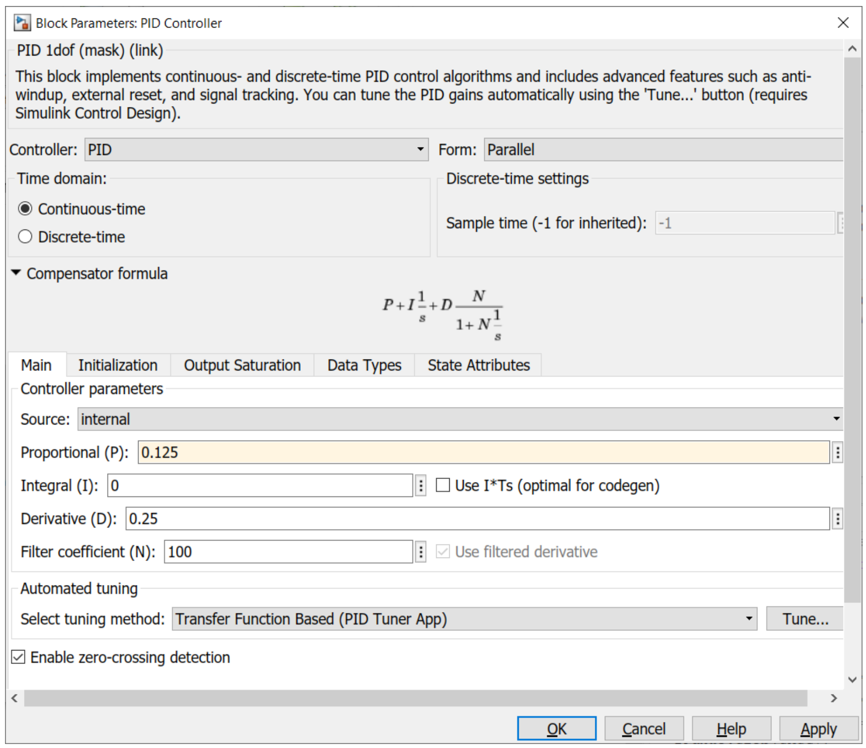


Figure 3: Details of implementation of PD controller within Simulink's "PID Controller" block.

4.3 What does the filter coefficient N do?

If you looked at the details of the "PID Controller" block shown in Figure 3, you likely noticed a lot of options in addition to the proportional, integral, and derivative gains.

It turns out that it is impossible to implement a true, ideal derivative. This is because an ideal derivative (transfer function s) is not proper, and thus not BIBO stable. For example, if a unit step is the input to an ideal derivative, the output is an impulse, which has an infinite magnitude. This means that Simulink is going to implement an approximate derivative, and the filter coefficient N determines the bandwidth of this approximation. In the PID dialog box, Simulink tells us that it will be implementing the following transfer function for the derivative term:

$$D\frac{N}{1+N\frac{1}{s}}$$

by multiplying the top and bottom by s, we get

$$Ds\left(\frac{N}{s+N}\right)$$

We can consider this an ideal derivative term (Ds) that has been modified by multiplying by a first order system

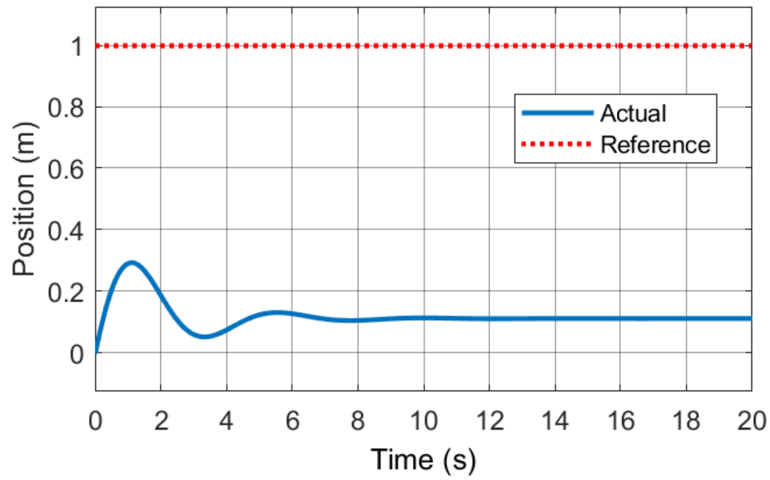


Figure 4: Results for mass-spring-damper plant control example with PD control. The transient specifications are almost satisfied, but it is clear from this plot that the actual position of the mass $y(t)$ does not track the reference value of 1 m.

$\left(\frac{N}{s+N}\right)$ that has DC gain 1 and pole magnitude $\sigma = N$. This first order term makes the overall transfer function proper, and BIBO stable (with positive N). However, if N is large enough, and thus the pole is fast enough, it will have little effect on the system. Often, we can use Simulink's default for this filter. If we want to change it, a good rule of thumb is to choose N to be 10 times the desired closed loop bandwidth.

5 Lecture Highlights

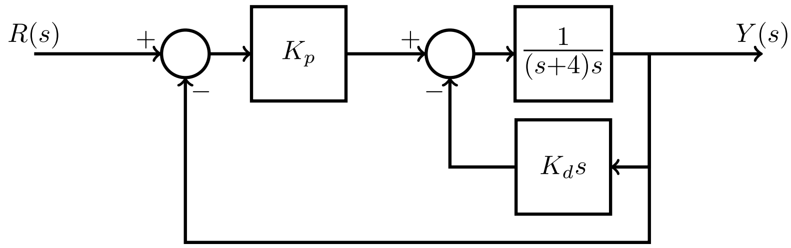
The primary takeaways from this article include

1. PD controllers can be useful for improving transient response.
2. The method for PD control design – that is, selecting the parameter values K_p and K_D in PD control design in this article is to find (1) the actual transfer function, given the controller $K_d s + K_p$ and plant $G(s)$, and (2) the desired transfer function $K\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$ from the transient specifications, then to set them equal to each other. Verification and further tuning in Simulink is usually necessary.

6 Quiz Yourself

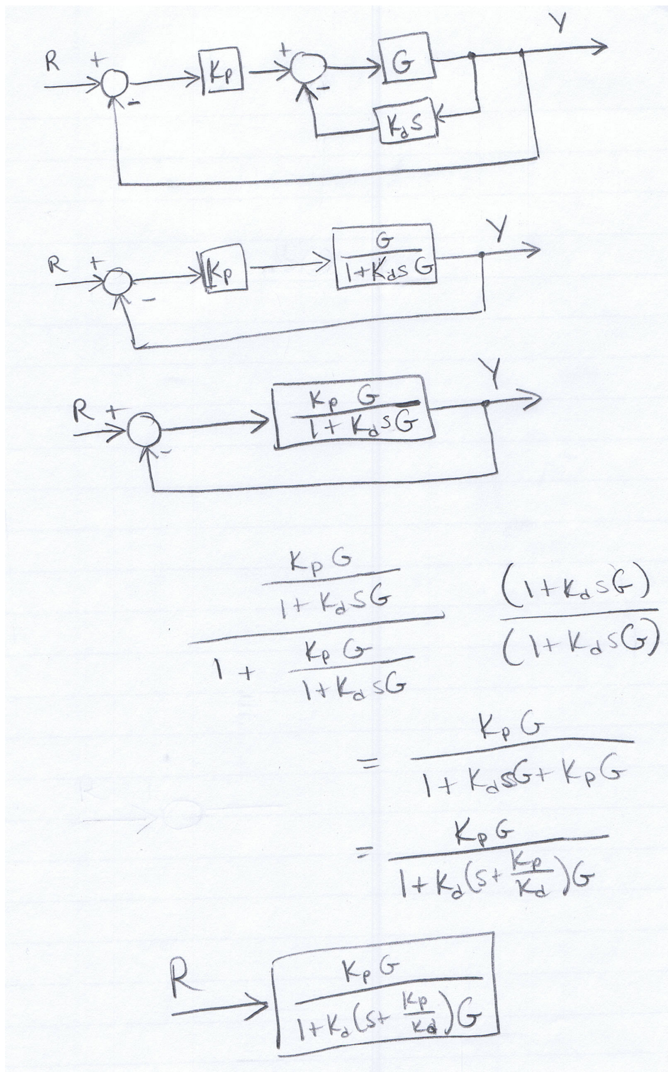
6.1 Questions

1. Use block diagram reduction techniques to derive the closed-loop transfer function for the "physically-derived" configuration in Section 4.1
2. Consider again the "Design System" form for PD control. Show that increasing the closed-loop damping ζ also results in increasing the impact of the zero on the closed-loop system.
3. Design a PD controller for the plant $G(s) = \frac{1}{(s+4)s}$ so that the closed loop has a settling time of $< .8$s and overshoot $< 10\%$.



6.2 Solutions

1.



2.

$$G(s) = \frac{1}{s^2+bs+c}$$

$$T(s) = \frac{K_d\left(s+\frac{K_p}{K_d}\right)\frac{1}{s^2+bs+c}}{1+K_d\left(s+\frac{K_p}{K_d}\right)\frac{1}{s^2+bs+c}}$$

$$= \frac{K_d\left(s+\frac{K_p}{K_d}\right)}{s^2+(b+K_d)s+(K_p+c)} \quad (1)$$

compare $T(s)$ to $K\dfrac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$ (2)

where damping ζ is in the middle term multiplying s^1. Thus, to increase damping we must increase K_d (since $b+K_d = 2\zeta\omega_n$).

But, from the numerator of (1), larger K_d means smaller $\frac{K_p}{K_d}$, which means the closed-loop zero at $-\frac{K_p}{K_d}$ is closer to the origin & has more impact on the closed-loop response.

3.

$$\frac{Y(s)}{R(s)} = \frac{K_p \cdot \dfrac{1}{s^2+(4+k_d)s}}{1 + \dfrac{K_p}{s^2+(4+k_d)s}} = \frac{K_p}{s^2+(4+k_d)s+K_p}$$

$$t_s = \frac{4.6}{\zeta\omega_n} < .8 \Rightarrow \zeta\omega_n > 5.75$$

$$\%OS < 10\% \Rightarrow \zeta > \frac{-\ln(.1)}{\sqrt{\pi^2+(\ln .1)^2}} = .59$$

choose $\zeta\omega_n = 6 \quad \zeta = .6 \quad \Rightarrow \omega_n = 10$

$$\underline{K_p = \omega_n^2 = 100}$$

$$4+k_d = 2\zeta\omega_n = 12 \Rightarrow k_d = \underline{8}$$