POLYTECH
Peter the Great
St.Petersburg Polytechnic
University

Course
« Introduction to Biomedical Engineering»

Dr. Kirill Aristovich

Section 2: Control theory
Lecture 2.3: PID controller

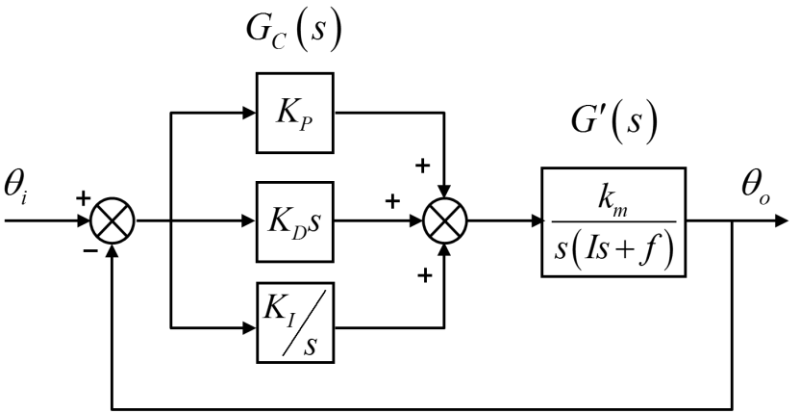
PID controller

Hello, in this lecture we will talk about the most commonly used controller out there. It is called PID controller by the first letters of individual control components: Proportional, Integral, and Derivative. Rumor goes that they account for nearly 90% of all use cases, which is a fine example of totally made up statistics, but I would not be surprised if it is true.

To see how it works and why it is used, let's start by considering each component independently. Note that we have transferred the controller from a feedback path to a forward path, to illustrate the point of manipulating with error, but this still IS the feedback controller.

So the proportional controller basically amplifies the position error. Let's see how it works. Formally, we can easily compute the new transfer function. If you evaluate the step response and track the performance, you can see that Low values of Kp give stable but slow responses, and high steady state velocity error. High values reduce steady state velocity error but response overshoots considerably. To understand why these overshoots occur, let's look at the error term over time for a purely proportional controller for, say, a motor. With proportional control the error and thus the control signal does not reach zero until the motor is at the desired position. However, due to the inertia of the system, the motor continues to move even without a control signal, resulting in an overshoot. Controlling the servo in this manner is like NOT applying the brakes until you reach your target position when driving!

Putting all three of these controllers together gives the complete PID



$$G_c(s) = K_P + K_D s + \frac{K_I}{s} = \frac{K_D s^2 + K_P s + K_I}{s}$$

Figure 1 - PID Control

In reality when driving you apply the brakes beforehand, and also dependent upon the speed you are travelling. You break earlier when you are driving faster (I hope!).

We can replicate this by considering the derivative of the error. This is negative as the servo approaches the target, so offers a way of restraining the motor forward motion. Further, if we consider the integral or total error over time: This gradually increases over time, and can be used to magnify the control signal for small errors, and improve the steady state error.

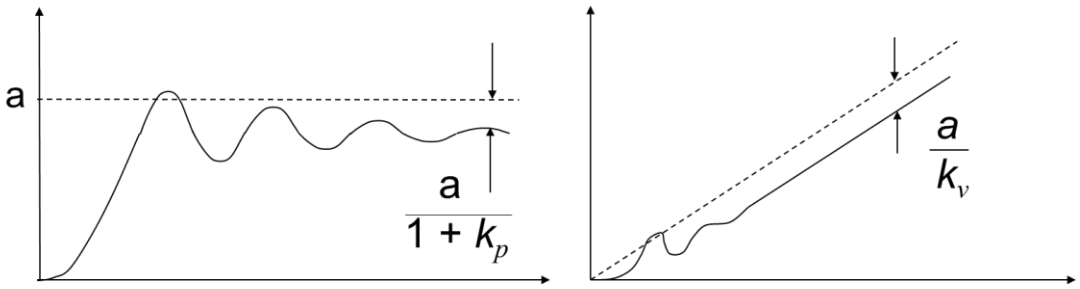


Figure 2 - Steady State Error

In order to understand this formally, let’s consider a derivative controller with unity proportional gain. Its open loop transfer function is the following, and the closed loop transfer function is on your screens. The closed-loop system is still second order, but now there is a zero in the numerator. The effect of this is subtle compared to the change in the poles. More importantly, the steady state velocity error of this system can be computed and we can see that, unlike for example, pure velocity feedback, the value is unchanged by derivative error. So overall we can improve transient response without compromising steady state error, which is great. The effect on the step response is following: The poles of the system become less oscillatory and decay quicker. This, combined with a reduced proportional gain, gives an improved transient response.

Open Loop gain $$G'(s) = \frac{k_m\left(K_D s^2 + K_P s + K_I\right)}{s^2\left(Is + f\right)}$$

The closed loop transfer function is then

$$F(s) = \frac{k_m\left(K_D s^2 + K_P s + K_I\right)}{s^2\left(Is + f\right) + k_m\left(K_D s^2 + K_P s + K_I\right)}$$

Figure 3 - Open Loop gain and The closed loop transfer function

This is clear when looking at the step response, and the relative contributions of the two error terms. At the start, the derivative term is significant and in the opposite direction to the proportional error term, but becomes negligible as the system settles.

The integral of the error is used for correcting steady state errors, as it adds a pole at zero in the transfer function. We can see that by computing closed-loop transfer function and noting that now we have s^3 in denominator.

Computing the steady state velocity error, we can see that it is zero. YAY! If you look at the s-plane, you can see that the effect of an extra pole close to origin – which would make our system very slow - is largely cancelled out by the nearby zero. This means the system is broadly similar to a proportional controller, with the exception of improved steady state performance.

Looking at ramp response it is clear how the integral term has enabled proper tracking of a ramp input! This is particularly useful for overcoming steady state errors arising from nonlinearities not included in the model which prevent proper tracking in reality.

Putting all three of these controllers together gives the complete PID. Here it is in all its glory! The closed-loop transfer function is rather big, but shows that it is possible to design a controller with improved transient response and decreased or zero steady state error. In theory. Just need to choose Kp, Ki, and Kd appropriately.

There is no unique solution for the settings for the three gains, and so the values are dependent upon the specific system and the application. Selecting these parameters is known as tuning, and it was (and still is in machine learning circles) a very active area of research.

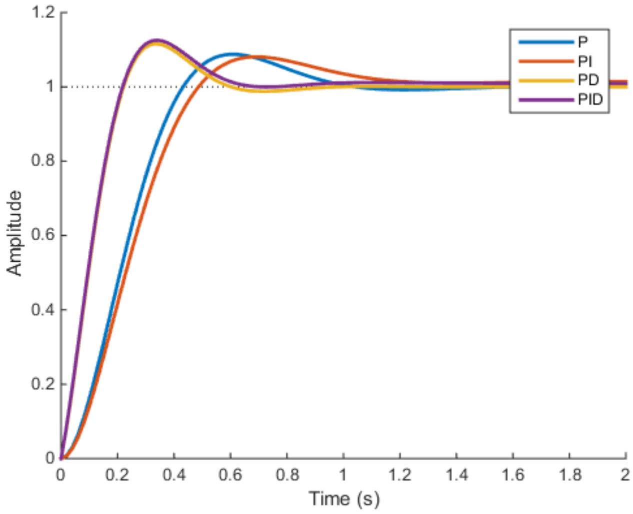


Figure 4 - PID parameters

Sometimes overshoot reduction is more important than settling time (like in heaters or ventilators). In other cases reducing settling time may be crucial and some overshoot is an acceptable trade off: like in camera focus. Many methods for “Tuning” these parameters exist, and often form the first estimate, before refinement based on testing real system.

Finally, I must repeat, that these controllers are used everywhere, and can even be applied with little or no knowledge of the system being controlled, often producing acceptable results!

This course was developed with the support of the "Open Polytech" educational project

OPEN POLYTECH

Online courses from the top instructors of SPbPU