Understanding PID control in the sense of differential equations

Yang, Hanzhi

Recall that a PID control law is defined as

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)\, d\tau + k_d \frac{d}{dt} e(t) \tag{1}$$

where $e(t)$ is the target tracking error. You may also find the block diagram of a typical closed loop control system below.

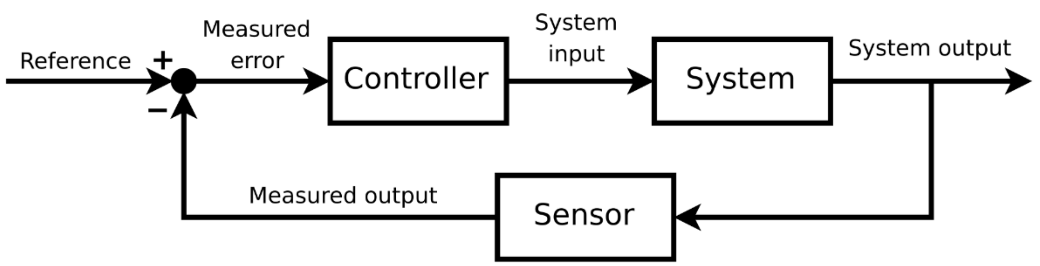

Figure 1. Closed-loop control system diagram

Technically, you can always blindly tune the three coefficients k_p, k_i, k_d. There have been many methodologies developed for that, like the Ziegler–Nichols method. But the knowledge of the system model can always be helpful to gain the desired control performance.

For instance, in this lab, we let you tune a PID controller for a very simple model: using the velocity (u) of the TurtleBot to control its position (x), as shown below.

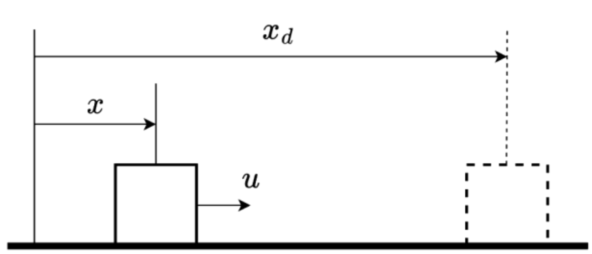

Figure 2. Controlling position using robot's command velocity

Then we can identify the system model as

$$\dot{x} = u \tag{2}$$

Following Fig. 1, we know that the tracking error $e(t) = x_d - x$, so we can further build the error dynamics:

$$\dot{e} = \dot{x_d} - \dot{x} = -u \tag{3}$$

Note that since x_d is a constant in this case, $\dot{x_d} = 0$. Now if we substitute Eq. 1 to Eq. 3, we have

$$\dot{e} = -k_p e - k_d \dot{e} - k_i \int_0^t e \, d\tau \tag{4}$$

Taking everything to the left side of the equation yields

$$(1 + k_d)\dot{e} + k_p e + k_i \int_0^t e \, d\tau = 0 \tag{5}$$

To solve this differential equation, let's take the derivative of both sides of the equation,

$$(1 + k_d)\ddot{e} + k_p \dot{e} + k_i e = 0 \tag{6}$$

A linear second order homogeneous differential equation! Then solving it can be easy by taking its characteristic equation

$$(1 + k_d)r^2 + k_p r + k_i = 0 \tag{7}$$

which has two roots

$$r = \{r_1, r_2\} = \frac{1}{2}\left[-\frac{k_p}{1 + k_d} \pm \frac{1}{1 + k_d} \cdot \sqrt{k_p^2 - 4k_i(1 + k_d)}\right] \tag{8}$$

And with the two roots, we know that the general solution of Eq. 6 is

$$e(t) = c_1 \exp(r_1 t) + c_2 \exp(r_2 t) \tag{9}$$

Now we face two possibilities- whether r's are real numbers or complex numbers. First let's see what if r's are complex numbers. To have complex roots, we need to have

$$k_p^2 - 4k_i(1 + k_d) < 0 \tag{10}$$

and then the roots would be (in complex form)

$$r = \{r_1, r_2\} = \frac{1}{2}\left[-\frac{k_p}{1 + k_d} \pm \frac{i}{1 + k_d} \cdot \sqrt{4k_i(1 + k_d) - k_p^2}\right] \tag{11}$$

We can be sure of the system's stability because the real part of the root is $Re\{r\} = -\frac{k_p}{2(1+k_d)}$, which is always negative since all the three PID coefficients are positive. But how would the system output, the position of the robot, perform? Since the roots are so long, let's rewrite them as $r =$

$\{r_1, r_2\} = a \pm ib$, where $a = -\frac{k_p}{2(1+k_d)}$ and $b = \frac{\sqrt{4k_i(1+k_d)-k_p^2}}{2(1+k_d)}$. Plugging the roots back to Eq. 9, we get the general solution

$$e(t) = c_1 \exp[(a+ib)t] + c_2 \exp[(a-ib)t] \tag{12}$$

Remember the Euler's equation (and he surely has a lot of equations)?

$$\exp(r+i\theta) = \exp(r)\,[\cos(\theta) + i \cdot \sin(\theta)] \tag{13}$$

Using Eq. 13, we may rewrite Eq. 12 as

$$e(t) = c_1' \exp(at)\cos(bt) + c_2' \exp(at)\sin(bt) \tag{14}$$

And let's see its plot:

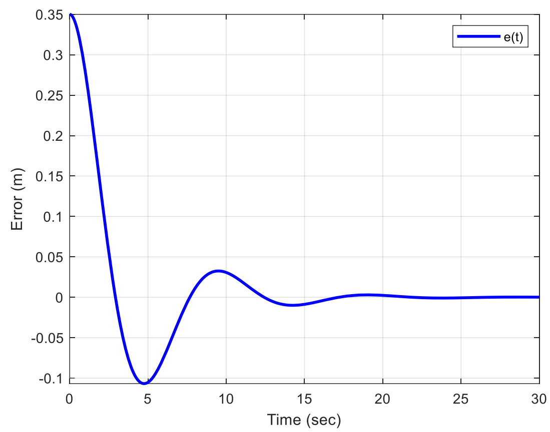

Figure 3. Tracking error with complex roots

You can see the trend of convergence governing by the exponential function, but you may also notice the oscillation introduced by the trigonometric functions.

Now let's see the second case, in which the roots are real numbers. In this case, we should have the PID coefficients satisfying

$$k_p^2 - 4k_i(1+k_d) > 0 \tag{15}$$

We can easily prove the stability of the system, since $\sqrt{k_p^2 - 4k_i(1+k_d)} < k_p$ always holds as long as the coefficients are kept positive. Then let's plot the solution:

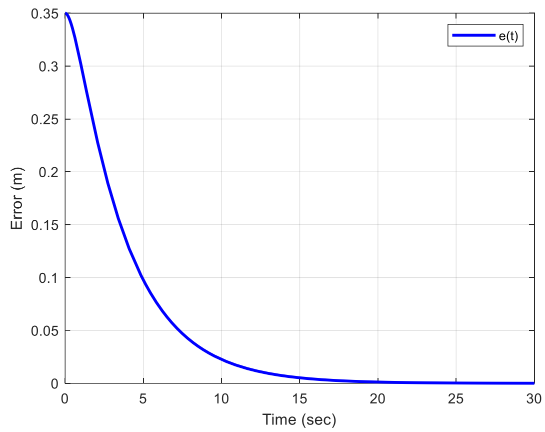

Figure 4. Tracking error with real roots

Voilà! No oscillations now, because the tracking error is now governed only by the exponential functions.

But wait a minute, we have covered the cases where either Eq. 10 or Eq. 15 holds, but what if we have

$$k_p^2 - 4k_i(1 + k_d) = 0 \tag{16}$$

Well, that really is a special case, isn't it? Here we have a tracking error plot like this:

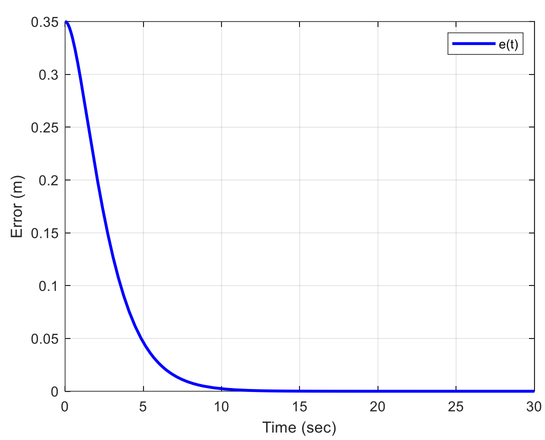

Figure 5. Critical tracking error

Looks similar to what we have in Fig. 4 above, right? But look at how the rise time in Fig. 5 is smaller than that in Fig. 4 but, at the same time, larger than that in Fig. 3, which means that at this point, if we want the system to react faster, we will have oscillations in the output, but if we don't want the oscillation, the responding time will be longer.

Here we want to introduce a concept called "damping". By definition, damping is an influence within or upon an oscillatory system that has the effect of reducing or preventing its oscillation. And to characterize how damped a system is, we usually use a dimensionless measure called "damping ratio" (ζ). It varies from "undamped" ($\zeta = 0$), "underdamped" ($\zeta < 1$) through "critically damped" ($\zeta = 1$) to "overdamped" ($\zeta > 1$). Following these definitions, we can find the error performance in Fig. 3 to be underdamped, the one in Fig. 4 to be overdamped, and the one in Fig.5 to be critically damped. Recall that for a second order system, we always have a characteristic equation of

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \tag{17}$$

where ω_n is the natural frequency. Comparing Eq. 17 to Eq. 6, we can get the damping ratio of our error dynamics:

$$\zeta = \frac{k_p}{2\sqrt{k_i(1+k_d)}} = \frac{1}{2} \cdot \sqrt{\frac{k_p^2}{k_i(1+k_d)}} \tag{18}$$

Then substituting Eqs. 10, 15, 16 into Eq. 18 separately gives $\zeta < 1$, $\zeta > 1$, and $\zeta = 1$, which explains their damping behaviors.

Now can you see how the choice of your PID coefficients may lead to different output behaviors by affecting the control system's damping ratio? This is why knowing the system model always helps to tune your PID. With the knowledge of the model, you can have a simple prediction of the possible output behaviors and choose the three numbers (k_p, k_i, k_d) accordingly.