Concept Review

PID Control

Why PID Control?

When implementing advanced controllers, a grasp of the basic characteristics of the plant model is often required. Model-based controllers are suited for industrial applications in the presence of noise, disturbances, etc., but require a thorough analysis and model to begin with. In contrast, PID controllers are very robust as they generally combine the advantages of all three types of compensation terms – proportional, integral, and derivative.

This combination is straight-forward to implement given an error term, requires fundamental knowledge of the compensation terms, and has extensive support in literature on common manual and automatic tuning methods. These make PID controllers very popular amongst control engineers.

An important component of any robotics platform is the concept of controlling states to reach a desired value. A common control technique is called a **Proportional-Integral-Derivative** (**PID**) controller. This control strategy compares the current state of a system to the desired value and calculates an error term $e(t)$. As this error term changes with time a PID controller can be used to generate command inputs to actuators on any platform.

1. PID Control

The block diagram for a PID control loop system is shown in Figure 1. It includes 3 terms:

- the proportional term is based on the present error
- the integral term depends on past errors, and
- the derivative term is a prediction of future errors.

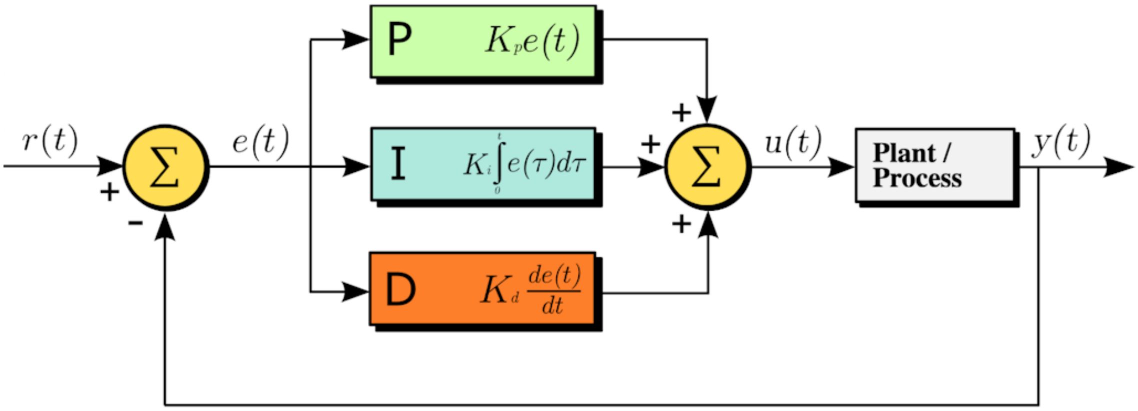

Figure 1. PID Control block diagram for a generic plant

Where:

- $\boldsymbol{r(t)}$ is the reference/desired value (control input),
- $\boldsymbol{e(t)}$ is the error between the desired value $r(t)$ and the measured value $y(t)$,
- $\boldsymbol{y(t)}$ is the measured value of the system,
- $\boldsymbol{K_p, K_i}$ and $\boldsymbol{K_d}$ are the proportional, integral and derivative gain and,
- $\boldsymbol{u(t)}$ is the control action which is the sum of the proportional, integral and derivative control gains.

The goal of the control input $u(t)$ is to minimize the error $e(t)$ over time.

The equation for the error term can be written as:

$$e(t) = y_d(t) - y_m(t) \tag{1.1}$$

The proportional, integral, and derivative control can be expressed mathematically as follows:

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)d\tau + k_d \frac{de(t)}{dt} \tag{1.2}$$

The corresponding Laplace transform of equation that describes equation 1.2 as a transfer function is the following:

$$U(s) = k_p + \frac{k_i}{s} + k_d s \qquad 1.3$$

Which would look like the block diagram in Figure 2.

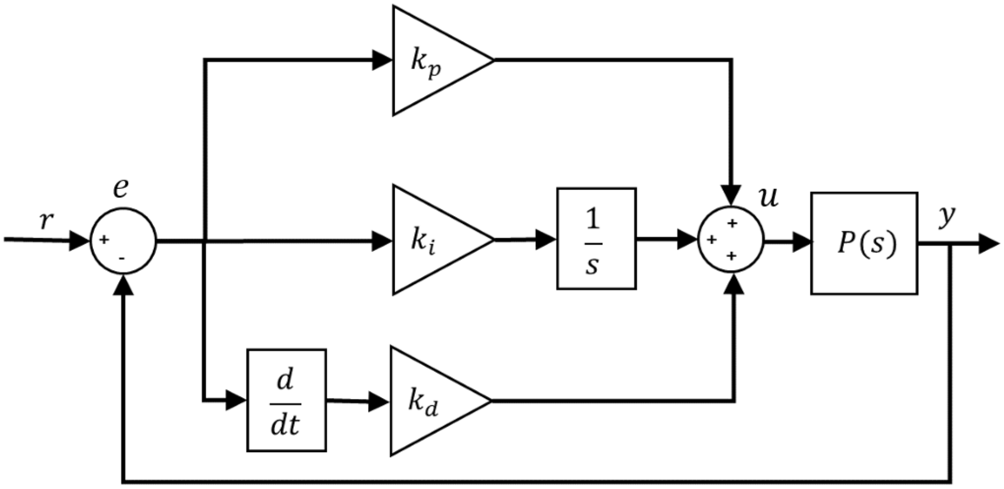

Figure 2. PID Transfer Function control block diagram

a. Unity Feedback Loop

Recall that the equation for a closed-loop system transfer function for unity feedback as shown in Figure 3 will always be given by,

$$\frac{\boldsymbol{Y(s)}}{\boldsymbol{R(s)}} = \frac{\boldsymbol{C(s) \cdot P(s)}}{\boldsymbol{1 + C(s) \cdot P(s)}} \qquad 1.4$$

Where $P(s)$ is the plant of the system and $C(s)$ is the controller that will be applied to the system. In the case of Figure 2, the controller is the PID section from the image.

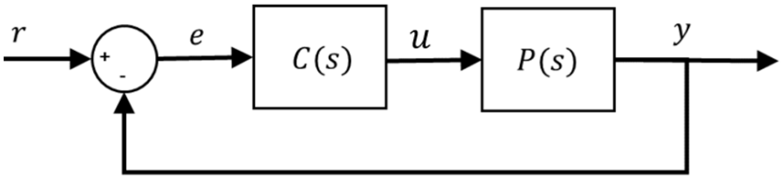

Figure 3. Unity feedback loop

We can find the error transfer function $E(s)$ in Figure 3 in terms of the reference $R(s)$, the plant $P(s)$, and the controller $C(s)$. The Laplace Transform of the error is:

$$\boldsymbol{E(s) = R(s) - Y(s)} \qquad 1.5$$

2. Proportional Control

A proportional only control removes the integral and derivative gain, if we remove those terms from Figure 2, the resulting system looks like Figure 4.

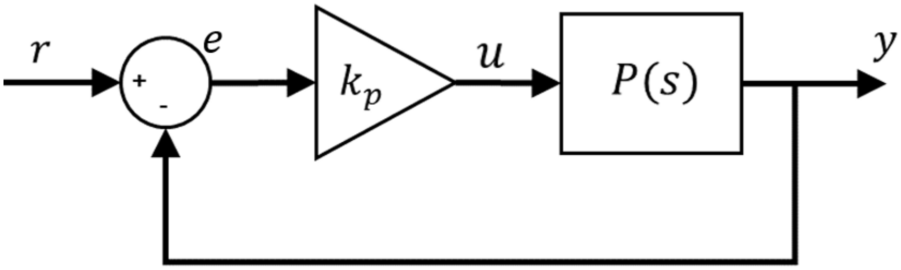

Figure 4. Proportional control block diagram

Where:

- k_p is the proportional gain
- $u(t) = vm(t)$ is the control input.
- $P(s)$ is the plant, which is the representation of the system.
- $R(s) = L[r(t)]$ is the Laplace transform of the setpoint.
- $Y(s) = L[y(t)]$ is the Laplace transform of the measured value.

3. For Qube – Servo

The voltage $V_m(s)$ to speed $\Omega_m(s)$ transfer function of a DC motor is:

$$\frac{\Omega_m(s)}{V_m(s)} = \frac{K}{\tau s + 1} \tag{3.1}$$

The voltage $V_m(s)$ to position $\Theta_m(s)$ transfer function is

$$\frac{\Theta_m(s)}{V_m(s)} = \frac{K}{s\,(\tau s + 1)} \tag{3.2}$$

Where:

- K is the model steady-state gain
- τ is the model time constant
- $\Omega_m(s) \;=\; L\,[\omega_m(t)]$ is the Laplace transform of the motor/disc speed
- $\Theta_m(s) \;=\; L\,[\theta_m(t)]$ is the Laplace transform of the motor/disc position, and
- $V_m(s) \;=\; L\,[v_m(t)]$ is the Laplace transform of the applied motor voltage.

Note how the voltage to position transfer function is the same as equation 3.1 with an integrator in series. It multiplied by $1/s$ to integrate it in time to find the position of the motor.

NOTE: If no modeling lab has been done, for Qube-Servo 3, $K = 24$ and $\tau = 0.1$ are good defaults if this equation needs to be used.

a. Proportional Control

We can use a proportional system like the one shown in Figure 4 to control both the position and the speed of a Qube-Servo. Using that diagram as a reference:

- k_p is the proportional gain
- $u(t) = v_m(t)$ is the control input $V_m(s) = L[v_m(t)]$ is the Laplace transform of the applied motor voltage.

$P(s)$, $R(s)$, and $Y(s)$ are different depending on the system doing position or speed control:

For Position Control
- $P(s)$ is the plant. It is the voltage to position transfer function $\Theta_m(s)/V_m(s)$ (equation 3.2)
- $R(s)$ is the setpoint (desired angle), $\Theta_d(s) = L[\theta_d(t)]$.
- $Y(s)$ is the measured angle, $\Theta_m(s) = L[\theta_m(t)]$.

For Speed Control
- $P(s)$ is the plant. It is the voltage to speed transfer function $\Omega_m(s)/V_m(s)$ (equation 3.1)
- $R(s)$ is the setpoint (desired angular velocity), $\Omega_d(s) = L[\omega_d(t)]$.
- $Y(s)$ is the measured velocity, $\Omega_m(s) = L[\omega_m(t)]$.

Proportional Position Control

When doing position control, **the plant $P(s)$** is the motor's voltage to position transfer function shown in equation 3.2. Using Figure 3 and Figure 4 as a reference, **the controller $C(s)$** is the k_p, the equivalent system diagram is shown in Figure 5.

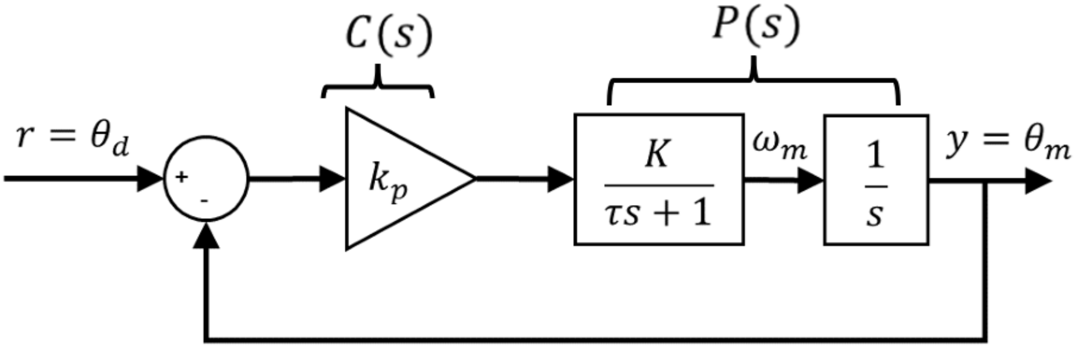

Figure 5. Proportional control of servo position

Using equation 1.4, we can get the closed-loop transfer function for the reference input (desired position) $R(s) = \Theta_d(s)$, to the output (measured position) $Y(s) = \Theta_m(s)$:

$$\frac{\boldsymbol{Y(s)}}{\boldsymbol{R(s)}} = \frac{\boldsymbol{\Theta_m(s)}}{\boldsymbol{\Theta_d(s)}} = \frac{\boldsymbol{Kk_p}}{\boldsymbol{\tau s^2 + s + Kk_p}} \qquad 3.3$$

Proportional Speed Control

When doing speed control, **the plant $\boldsymbol{P(s)}$** is the motor's voltage to speed transfer function shown in equation 3.1. similar to a proportional position control, **the controller $\boldsymbol{C(s)}$** is the k_p, the equivalent system diagram is shown in Figure 6.

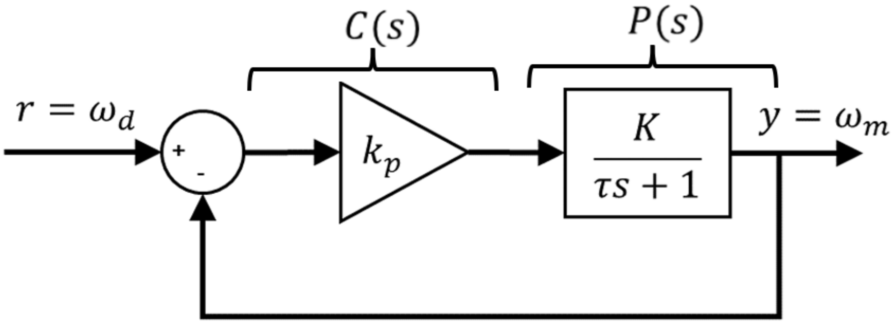

Figure 6. Proportional control of servo speed

Using equation 1.4, we can get the closed-loop transfer function for the reference input (desired speed) $R(s) = \Omega_d(s)$, to the output (measured speed) $Y(s) = \Omega_m(s)$:

$$\frac{\boldsymbol{Y(s)}}{\boldsymbol{R(s)}} = \frac{\boldsymbol{\Omega_m(s)}}{\boldsymbol{\Omega_d(s)}} = \frac{\boldsymbol{Kk_p}}{\boldsymbol{\tau s + 1 + Kk_p}} \quad 3.4$$

b. PD Position Control

The PID controller described by Equation 1.2 or Equation 1.3 is an ideal PID controller.

In the case of a Qube-Servo, to control position, a variation of the classic PD control can be used. This system uses the speed of the system instead of the derivative of the error for the gain, which allows the transfer function to match the standard second-order transfer function. This enables the user to design proper gains for a desired system response. Compared to classic PD control, this proportional with rate feedback approach (as shown in Figure 7) can lead to slightly better performance, with less overshoot, and ensures the closed-loop equation matches the standard 2nd order equation, rather than having a response that does not quite meet expectations in terms of overshoot and peak time.

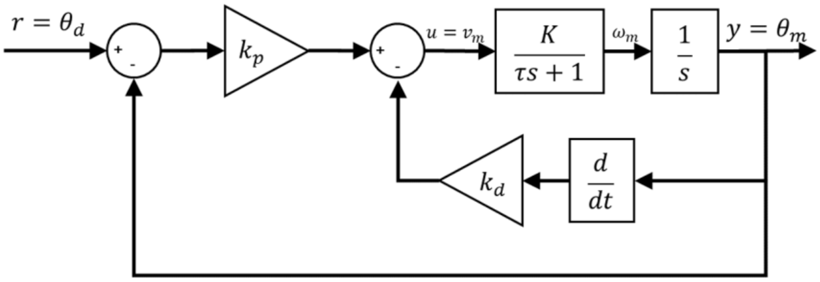

Figure 7. Block diagram of proportional plus rate feedback control

If we follow the diagram in Figure 7, the proportional plus rate feedback control has the following structure:

$$u = k_p\big(r(t) - y(t)\big) - k_d\dot{y}(t) \qquad 3.5$$

Where:

- k_p is the proportional gain,
- k_d is the derivative (velocity) gain,
- $r = \theta_d(t)$ is the setpoint or reference motor/load angle,
- $y = \theta_m(t)$ is the measured load shaft angle, and
- $u = V_m(t)$ is the control input (applied motor voltage).

The closed-loop transfer function of the Qube-Servo is denoted $Y(s)/R(s) = \Theta_m(s)/\Theta_d(s)$. Taking the Laplace transform of Equation 3.5 yields:

$$U(s) = k_p\big(R(s) - Y(s)\big) - k_d sY(s) \qquad 3.6$$

which based on Figure 7, $Y(s) = U(s)P(s)$, and the plant $P(s)$ is the voltage-to position transfer function of the Qube-Servo as shown in equation 3.2, this leads to:

$$Y(s) = \frac{K}{s(\tau s + 1)}\Big(k_p\big(R(s) - Y(s)\big) - k_d s \cdot Y(s)\Big) \qquad 3.7$$

Solving for $Y(s)/R(s)$, we can get the closed-loop transfer function for the reference input (desired position) $R(s) = \ \Theta_d\,(s)$, to the output (measured position) $Y(s) = \Theta_m(s)$:

$$\boldsymbol{\frac{Y(s)}{R(s)} = \frac{Kk_p}{\tau s^2 + (1 + Kk_d)s + Kk_p}} \qquad 3.8$$

This is a second-order transfer function. Recall the standard second-order transfer function:

$$\boldsymbol{\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}} \qquad 3.9$$

c. PID Position Control

The PID controller described by Equation 1.2 or Equation 1.3 is an ideal PID controller. For a PD controller, we use a modified version of this (equation 3.8) that ensures the closed-loop equation matches the standard 2nd order equation, rather than having a response that does not quite meet expectations in terms of overshoot and peak time.

In the case of a Qube-Servo, to control position using a PID controller, we can add an integral term to this system shown in Figure 7.

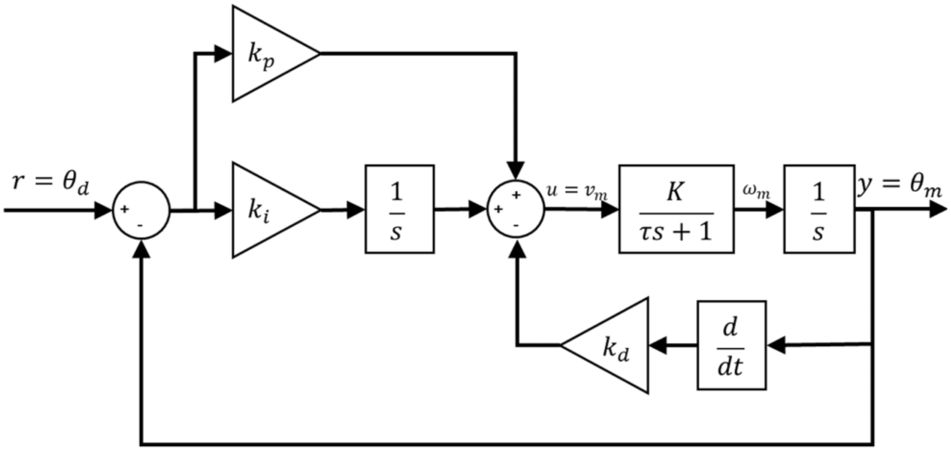

Figure 8. Block diagram of a modified PID servo position control

Using the diagram in Figure 8, the time domain equations of this PID control is

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau - k_d \frac{dy(t)}{dt} \tag{3.10}$$

With the corresponding transfer function

$$U(s) = \left(k_p + \frac{k_i}{s}\right)\left(R(s) - Y(s)\right) - k_d Y(s) \tag{3.11}$$

Substituting the voltage to position transfer function of the Qube (equation 3.2) into the previous equation 3.11, and solve for $Y(s)/R(s)$, we get the voltage to position closed-loop transfer function of the PID system as

$$\boldsymbol{\frac{Y(s)}{R(s)} = \frac{\frac{Kk_p}{\tau} s + \frac{Kk_i}{\tau}}{s^3 + \frac{(1 + Kk_d)}{\tau} s^2 + \frac{Kk_p}{\tau} s + \frac{Kk_i}{\tau}}} \tag{3.12}$$

QUANSER
INNOVATE · EDUCATE

Solutions for teaching and research. Made in Canada.