QUANSER
INNOVATE·EDUCATE

Concept Review

Introduction to Control

Why Use Controllers?

Unlike open-loop systems, closed-loop systems use sensory feedback that allows the system to adjust its performance to meet a desired output. Examples of everyday closed-loop systems include the temperature-control system at home or the cruise control in an automobile. A more advanced example would be an aircraft landing system.

Selecting a feedback controller depends on numerous factors such as the dynamic behavior of the system and the type of feedback sensor used. Common feedback control techniques include on-off control, unity feedback, and proportional-derivative (PD) control etc. Understanding these requires a fundamental grasp of compensation terms and how they affect the overall control strategy.

This document covers the basics of proportional terms, integral terms and derivative terms, and strategies to identify which term is needed.

Introduction to a Controller

Start with a plant model $P(s)$ that relates an action $U(s)$ to a measured output $Y_m(s)$. For example, consider a servo motor. The action is represented by a voltage, and the output could be represented by the speed the motor spins at. A voltage to speed transfer function provides the plant model,

$$P(s) = \frac{Y_m(s)}{U(s)} = \frac{\omega(s)}{V(s)} = \frac{K}{\tau s + 1}$$

Although it is not unimaginable to intuitively compute an action $U(s)$ ready that drives the plant's output $Y_m(s)$ towards a desired value $Y_d(s)$, it is often desirable to automate this process by designing a controller $C(s)$ that takes the necessary actions to enable this by monitoring $Y_m(s)$. This control strategy can be visualized using the unity feedback loop,

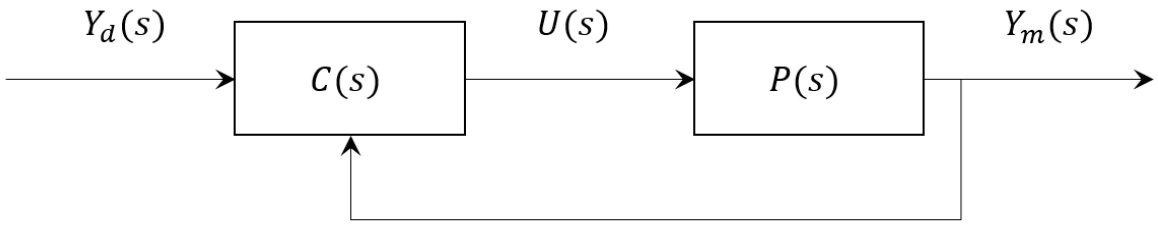

Figure 1 – Unity feedback loop describing the general structure of a controller and plant

Given the system above, we can define an error term $e(s)$, as,

$$e(t) = y_d(t) - y_m(t)$$

In the frequency domain,

$$E(s) = Y_d(s) - Y_m(s)$$

It is the controller's task to minimize this error term, and drive the output $Y_m(s)$ to $Y_d(s)$ following the desired specifications provided, such as peak time, offshoot, etc.

Performance characteristics

An in-depth description of system modeling and the effect of a step response on a first order and second order system is covered in detail by Modeling Concept Review. In this document we will relate the effect of a second order system with their compensation terms.

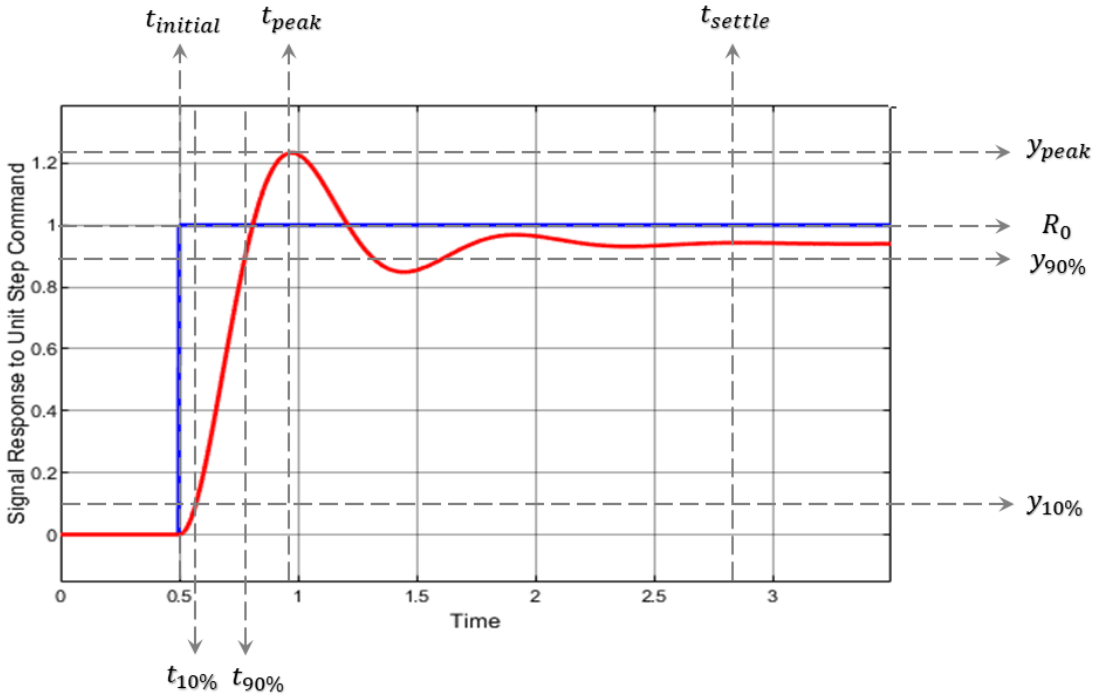

Figure 2 Response plot

Figure 2 is an example of a step response with a controller that has the characteristics of a second order system.

Percentage overshoot (PO):

PO is the maximum value of your modelled signal (y_{peak}), minus the step value (R_0), divided by the step value and multiplied by 100 to be expressed as a percentage. Or, more concisely:

$$PO = \frac{y_{peak} - R_0}{R_0} \times 100$$

Rise time:

This is the time t_r taken by the response to rise from 10% to 90% of its steady-state value ($y_{10\%}$ to $y_{90\%}$).

$$t_r = t_{90\%} - t_{10\%}$$

Settling time:

Settling time t_s is the time it takes for the response to settle to the steady-state value.

$$t_r = t_{settle} - t_{initial}$$

Peak time:

Peak time t_p is the time required by the response to reach its first peak. If the system has no overshoot, the peak time is not applicable.

$$t_p = t_{peak} - t_{initial}$$

By tuning the gains of the controller used in our system, we can modify these performance characteristics for a specific application.

Compensation terms

With the system presented in Figure 1 as the definition of the error term, there are three popular controller terms that must be considered,

- **Proportional control** is directly proportional to the magnitude of error. This is important at the beginning of a step response since the system can react immediately. When the system is close to the desired setpoint having a proportional control can lead the system to hunt and oscillate or go unstable if the control action is too high and the system has large overshoot. This can be represented by the equation,

$$U(s) = k_p\, E(s) = k_p\, (Y_d(s) - Y_m(s))$$

$$u(t) = k_p\, e(t) = k_p\, (y_d(t) - y_m(t))$$

- **Integral Control** focuses on the steady-state error by looking at the history of the system and trying to remove error accumulation over time. It's important to be careful when including integral control due to controller windup. This phenomenon occurs when the error starts to accumulate over time and the system has not reacted. In this scenario the controller stores a lot of energy and when the plant can react the integral term can overtake both the Proportional and Derivate terms. Resetting the integral term is a common practice when the control action is not currently required. This can be represented by the equation,

$$U(s) = k_i \frac{E(s)}{s} = k_i \frac{Y_d(s) - Y_m(s)}{s}$$

$$u(t) = k_i \int e(t)\, dt = k_i \int (y_d(t) - y_m(t))\, dt$$

- **Derivative control** is focused on the rate of change of the error. A controller which is only derivative control will not work unless there is an input which causes the system states to change. To overcome this fault a common control structure is a PD

controller where the proportional term quickly moves the system to the desired setpoint while the derivative term slows down the system response to avoid overshooting. This can be represented by the equation,

$$U(s) = k_d E(s)\, s = k_d (Y_d(s) - Y_m(s))\, s$$

$$u(t) = k_i \frac{de(t)}{dt} = k_i \frac{d}{dt}(y_d(t) - y_m(t))$$

Although these terms are often used together, for example, PID, or PI, or PD, they have niche uses in robotics. For example, a pure integral controller can be used for sensitive current-based control applications for robotic grippers.

Solutions for teaching and research. Made in Canada.