10.450 Process Dynamics, Operations, and Control
Lecture Notes - 26

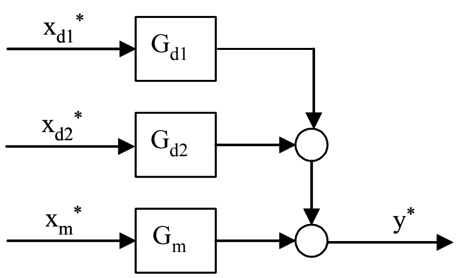
Lesson 26. Cascade control loops.

26.0 Context

We have implicitly assumed a single loop: one manipulated variable and one controlled variable. It turns out that single loop feedback control can be improved in some cases by adding other loops. This is not multivariable control – we are still working with a single controlled variable. In this lesson, we consider controllers in series.

26.1 Introducing cascade via intermediate variables

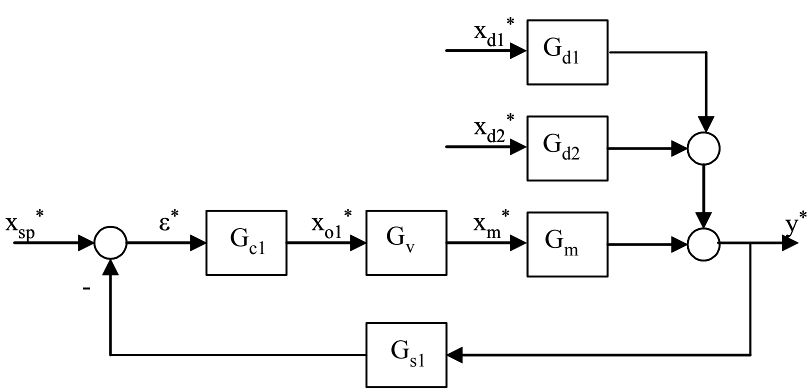
Consider a process with several inputs and an output that is to be controlled:



The equivalent Laplace domain equation is

$$y^*(s) = G_m x_m^*(s) + G_{d2} x_{d2}^*(s) + G_{d1} x_{d1}^*(s) \qquad (26.1.1)$$

The normal single-loop control scheme gives

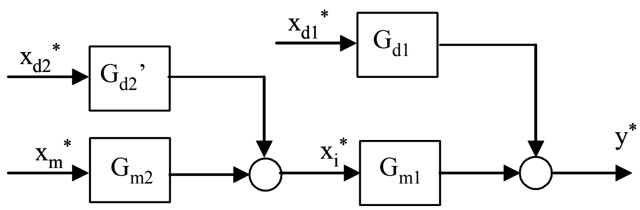


In some cases, we may be able to improve the response of y. Suppose, for example, that the process response $G_m(s)$ is slow, or that a particular disturbance variable x_{d2} is persistent and troublesome. IF we can find an intermediate process variable that

- we can measure
- clearly indicates the disturbance x_{d2}
- responds to changes in x_m

- responds faster to x_m than does y

then we may introduce cascade control. First redraw the block diagram of the process.

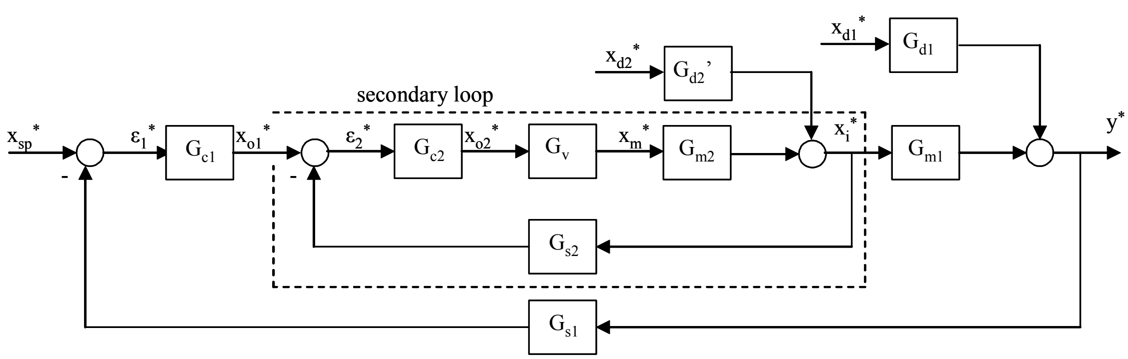


The diagram represents the same process as before; we have merely distinguished the effect of x_{d2} on the intermediate variable x_i. The Laplace domain equation is

$$y^*(s) = G_{m2}G_{m1}x_m^*(s) + G'_{d2}G_{m1}x_{d2}^*(s) + G_{d1}^*x_{d1}^*(s) \qquad (26.1.2)$$

Of course, (26.1.2) must equal (26.1.1), showing that the disturbance transfer function G_{d2} has been adjusted to G'_{d2} because of its new position in the diagram.

The intermediate variable x_i is called the *secondary variable*, and the control scheme now features a new *secondary* loop within the original *primary* loop.



The secondary loop controls the intermediate (secondary) variable x_i by adjusting the manipulated variable x_m. The primary loop controls the controlled variable y by manipulating the set point of the secondary controller x_{o1}. Thus we have the same controlled variable and set point as before, but the valve has been augmented by an inner control loop.

Disturbances x_{d2}^* are rejected by the secondary loop before they affect the full process, and thus response is quicker and the impact on y* less. The primary loop is necessary to handle the other disturbances, such as x_{d1}^*, that always exist. The extra layer of control does not degrade the response to x_{d1}*, because the process is usually much slower than the controller.

Cascade control is still feedback control, performed with conventional PID control algorithms. The improvement comes because we're looking inside the process, discriminating among disturbances, and applying feedback with increased deftness. The next logical step is to react directly to the disturbance, predicting what the manipulated variable should do, not waiting for a process response. This is the topic of feedforward control.