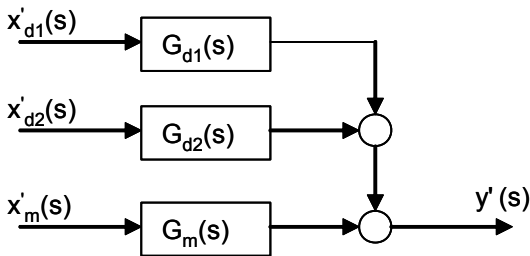


**8.0 context and direction**

Knowledge of the process is the basis for successful process control. In Lesson 7, we used process knowledge to guide, through tuning correlations, our choice of controller parameters. In this lesson, process knowledge will guide our choice of *control structure* itself: by making additional process measurements, we will augment the single-loop feedback control scheme to give it greater capability.

**DYNAMIC SYSTEM BEHAVIOR****8.1 a process with identifiable intermediate variable**

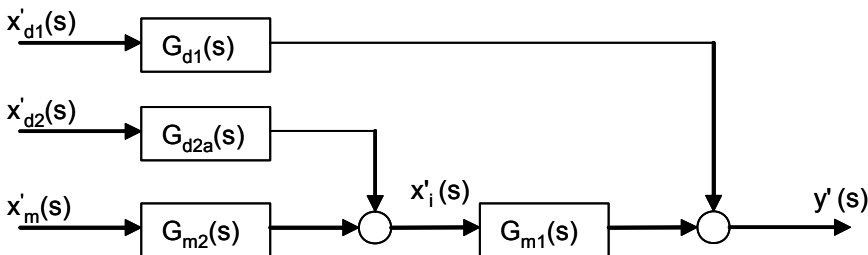
We begin with a process that has three inputs, two of them disturbances, and one output that we will wish to control. As usual, transfer functions  $G_{d1}(s)$ ,  $G_{d2}(s)$ , and  $G_m(s)$  may refer to the same assembly of equipment, but specify how the output variable  $y$  depends on each particular input.



The Laplace domain process description is then

$$y'(s) = G_m x'_m(s) + G_{d2} x'_{d2}(s) + G_{d1} x'_{d1}(s) \quad (8.1-1)$$

We imagine a case in which process  $G_m(s)$  could be divided into two parts, connected by a measurable intermediate variable  $x_i$ : this could be as simple as two tanks in series, as in Lesson 4. Having specified some of the interior structure of  $G_m$ , we consider  $x_{d2}$  to be typical of disturbances that affect the process further upstream and  $x_{d1}$  to affect the process downstream, after the intermediate variable.



The process description becomes

$$y'(s) = G_{m2}G_{m1}x'_m(s) + G_{d2a}G_{m1}x'_{d2}(s) + G_{d1}x'_{d1}(s) \quad (8.1-2)$$

Equations (8.1-1) and (8.1-2) describe the same process, so they must be equivalent. Comparing them, we find

$$G_m = G_{m2}G_{m1} \quad (8.1-3)$$

and

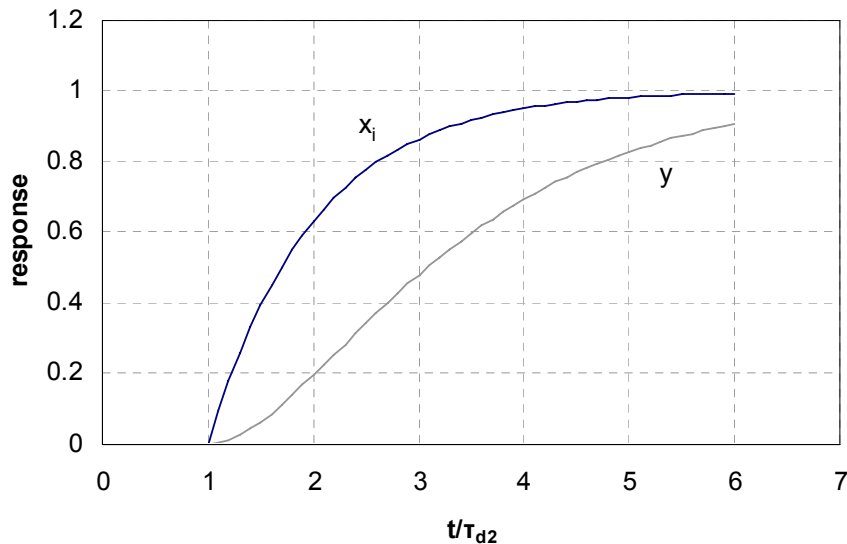
$$G_{d2} = G_{d2a}G_{m1} \quad (8.1-4)$$

Also, the intermediate variable is given by

$$x'_i(s) = G_{m2}x'_m(s) + G_{d2a}x'_{d2}(s) \quad (8.1-5)$$

## 8.2 response to disturbances

Suppose, for illustration, that we let each of these transfer functions be first order. Then the responses of  $x_i$  and  $y$  to a step in  $x_{d2}$  are shown in Figure 8.2-1.



**Figure 8.2-1. Step response of intermediate and output variables**

We observe that the intermediate variable responds before the output. Perhaps this can help us to improve the control of  $y$ .

**CONTROL SCHEME**

**8.3 step 1 - specify a control objective for the process**

Our control objective is to maintain the outlet variable  $y$  at set point.

**8.4 step 2 - assign variables in the dynamic system**

The controlled variable is  $y$ . The manipulated variable affects the controlled variable through the transfer function. These assignments are familiar from previous Lessons. However, we have some new assignments to make:

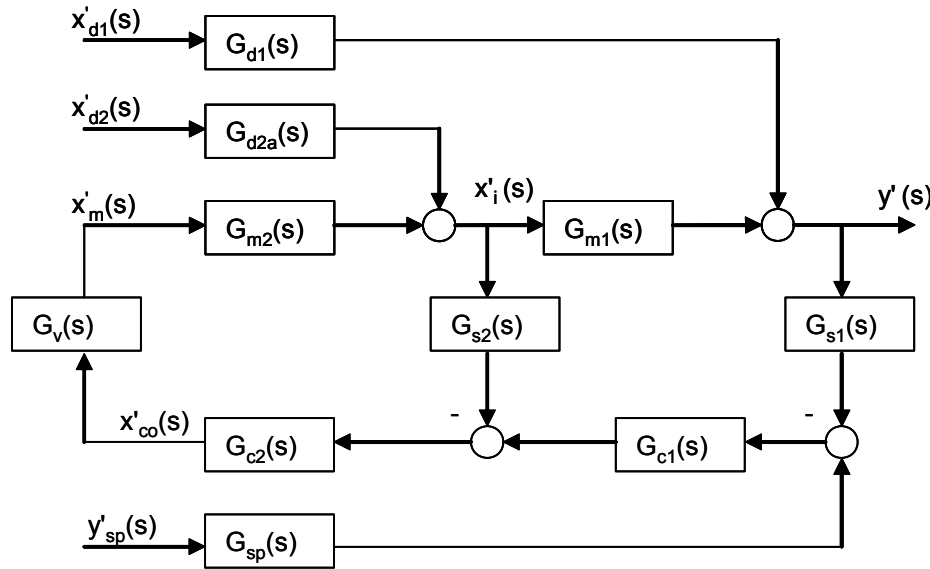
- By looking in more detail at the composition of this process, we identify an intermediate variable that influences the controlled variable, responds to disturbances before the controlled variable does, and responds to the manipulated variable, as well. We can use this extra information in a scheme called cascade control.
- Furthermore, suppose we can measure a disturbance variable that frequently disturbs the process. By this means, we can forecast when the controlled variable is about to be altered and forestall it in a scheme called feedforward control.

**8.5 step 3 - cascade control scheme**

The idea is to insert a secondary feedback control loop between the controlled variable  $y$  and manipulated variable  $x_m$ . The secondary loop controls intermediate variable  $x_i$ . This variable must hold several qualifications:

- it must respond to important disturbances (those that significantly affect the controlled variable)
- it must also convey the effects of such disturbances to the controlled variable
- it must respond to the manipulated variable

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.

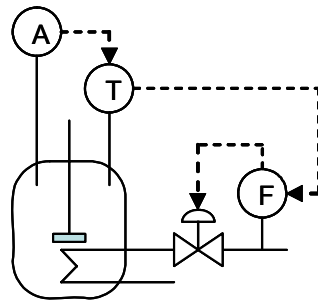


**Figure 8.5-1. Block diagram for cascade control structure**

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 can be carried to more nested levels. For example, in Figure 8.5-2 the composition controller sets a temperature set point in the secondary loop; the temperature controller in turn sets the flow set point for the tertiary loop.



**Figure 8.5-2. Three-level cascade control**

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.

### 8.6 step 3 - feedforward control scheme

We get closer to the root of the problem if we 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. We contrast simple feedback control in Figure 8.6-1 with feedforward control in Figure 8.6-2:

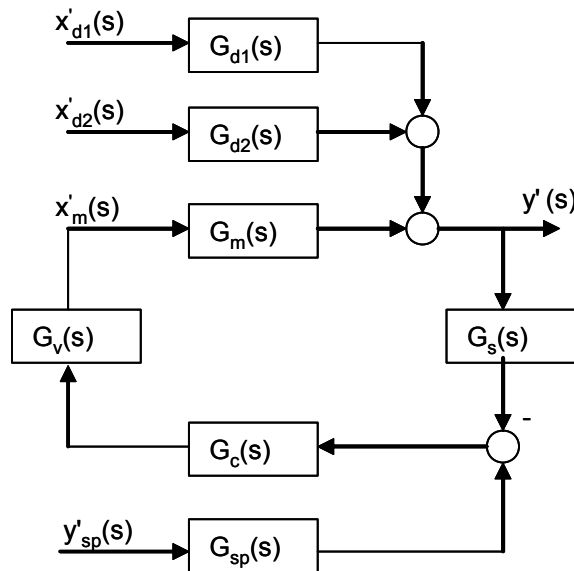
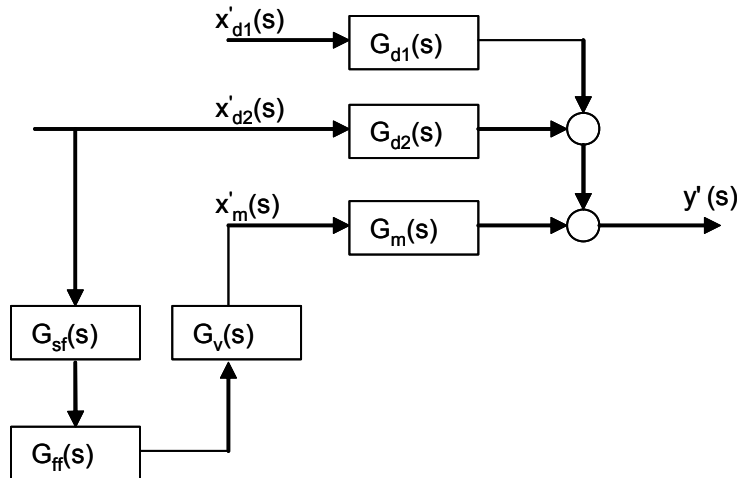


Figure 8.6-1. Feedback control diagram

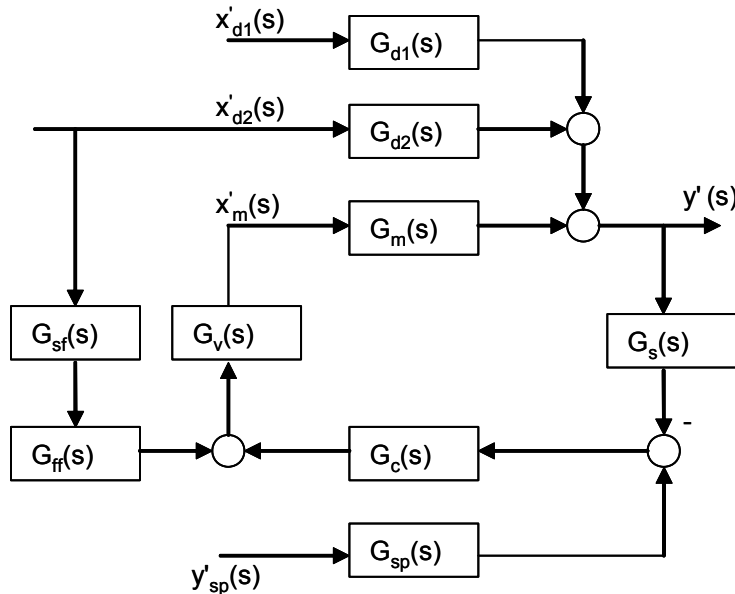
In feedback control, disturbance  $x_{d2}$  proceeds through the process ( $G_{d2}$ ) to affect controlled variable  $y$ . The controller reacts to the resulting error and adjusts the manipulated variable; the change in manipulated variable proceeds through the process ( $G_m$ ) after the fact to reduce the error.



**Figure 8.6-2. Feedforward control diagram**

In feedforward control, disturbance  $x_{d2}$  proceeds in parallel through the process ( $G_{d2}$ ) and through the feedforward controller ( $G_{ff}$ ). The controller adjusts the manipulated variable to counteract the disturbance, so that disturbance and manipulated variable affect output variable  $y$  together. In the very best of cases, the manipulated variable would compensate the disturbance step for step, so that the controlled variable would never be affected!

We also contrast feedforward with the cascade structure of Section 8.5. Feedforward control also adds another sensor and controller. However, the concept differs from that of cascade in that the disturbance is measured, but the manipulated variable does not affect it - there is no feedback. Feedforward is thus more specific than cascade control: it is designed to head off a particular disturbance. However, it cannot measure how well it did, nor can it respond to other disturbances, such as  $x_{d1}$ , that might affect the controlled variable. Hence feedforward is to be applied in conjunction with a conventional feedback loop. Both the feedback and feedforward controllers adjust the manipulated variable in Figure 8.6-3.



**Figure 8.6-3. Feedforward/feedback control diagram**

The feedforward algorithm is not conventional PID. Rather it is specific to the process and the disturbance. We wish to specify the feedforward controller transfer function  $G_{ff}$  to minimize the effect of  $x_{d2}$  on  $y$ . Ideally, we want  $y'(s) = 0$ , and from Figure 8.6-2 or 8.6-3 this requires that

$$G_{ff} = \frac{-G_{d2}}{G_{sf}G_vG_m} \quad (8.6-1)$$

The parallel path through the feedforward controller makes use of advance warning about the disturbance. Given perfect process models, plus the ability to render those in transfer function  $G_{ff}$ , the compensation can completely negate the effect of  $x_{d2}$ . Of course, perfection is unlikely, as we will see later.

### 8.7 summary comparison between cascade and feedforward

On first encounter, one is apt to confuse cascade and feedforward with one another. Table 8.7-1 shows a side-by-side comparison: assume an existing process with a feedback controller, such as that in Figure 8.6-1. Call this controller the primary controller, and compare adding either a secondary feedback controller in a cascade scheme (Figure 8.5-1), or a feedforward controller in a feedforward/feedback scheme (Figure 8.6-3). The table highlights the similarities and differences of the two schemes.

We should conclude the comparison with a clarification of concept: think of “cascade” and “feedforward” as ways to arrange controllers, not confined to the specific arrangements given above. That is, it is possible

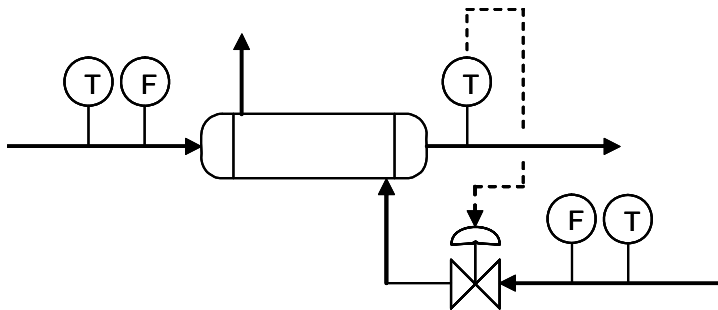
to have a feedforward controller that adjusts the set point of a feedback controller. In this circumstance, we have both feedforward control and a cascade structure; we might say that the feedforward controller “cascades” to a secondary controller. We will show an example in the next section.

**Table 8.7-1. Comparison between cascade and feedforward control**

| similarities   |   |   |
|--|---|---|
| primary controller attempts to correct deviations from set point |   |   |
| primary controller set point unchanged                           |   |   |
| manipulated variable unchanged                                   |   |   |
| add extra sensor and controller                                  |   |   |
| differences  | cascade (Figure 8.5-1)  | feedforward (Figure 8.6-3)  |
| extra measurement:   | intermediate variable   | disturbance variable  |
| characteristics of the new measured variable:                    | intermediate variable <ul style="list-style-type: none"> <li>• is affected by disturbances and manipulated variable</li> <li>• affects controlled variable</li> </ul> | measured disturbance <ul style="list-style-type: none"> <li>• is not affected by manipulated variable</li> <li>• affects controlled variable</li> </ul> |
| algorithm of added controller:                                   | PID   | specific to process model   |
| manipulated variable:  | directed by secondary controller  | directed by both feedforward and feedback controllers   |
| primary controller action:                                       | varies set point of secondary controller  | varies manipulated variable, as before  |
| helps by:  | reducing the degree of disturbance that reaches the controlled variable   | anticipating effects of a particular disturbance and responding to compensate   |

### 8.8 example

A feed stream is pre-heated using a heating oil in a shell-and-tube exchanger. The outlet temperature is controlled by manipulating the flow rate of the oil.

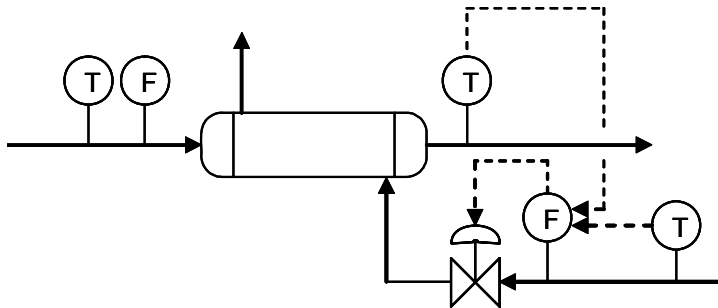


**Figure 8.8-1 Single-loop control of outlet temperature**

The process is subject to several disturbances: the flow rate and inlet temperature of both process (feed) and service (heating oil) streams may vary. The latter disturbances are particularly troublesome; because the heating oil is supplied from a header that feeds other, larger users, swings



in the supply pressure and temperature are frequent. We propose employing additional measurements and control loops, arranged as shown in Figure 8.8-2.



**Figure 8.8-2 Multi-loop control scheme for outlet temperature**

Flow control of the service stream is now effected by a secondary loop. Two primary controllers cascade to this new secondary loop. The first is the original feedback controller for the process temperature. The second is a feedforward controller from the oil temperature.

Should the oil supply pressure decrease, the secondary flow controller will respond by opening the supply valve. This will allow the oil flow rate to return to its desired value. The cascade structure responds to the pressure disturbance before the controlled variable is affected.

Should the oil supply temperature decrease, the feedforward controller will respond by directing the secondary controller to increase the flow rate. The higher flow rate of heating oil will tend to maintain the heat duty in the heat exchanger even as the supply temperature falls.

The temperature controller on the oil requires a feedforward algorithm because the manipulated variable (heating oil flow) does not affect the measured variable (heating oil temperature). The temperature controller on the process stream is a feedback controller that responds to any error in that temperature, such as may arise from disturbances in the process stream flow and temperature. Both controllers cascade to a secondary controller.

## EQUIPMENT

### 8.9 practical feedforward controllers

Of course there is an overwhelming variety of process models, but if we recall that a large class of processes may be successfully represented by FODT, we can derive a feedforward controller with some claim to generality (Marlin, 2000). Lumping the valve and sensor into the process description, the transfer function (8.6-1) becomes

$$G_{ff} = -\frac{K_d}{K_m} \frac{\tau_m s + 1}{\tau_d s + 1} e^{-(\theta_d - \theta_m)} \quad (8.9-1)$$

We generalize this form by associating a controller gain with the ratio of process gains, a lead time with the  $x_m$  process path, a lag time with the  $x_{d2}$  process path, and a dead time with the difference between disturbance and manipulated process dead times. These four parameters may then be tuned, as with PID parameters, to improve controlled variable response to  $x_{d2}$ .

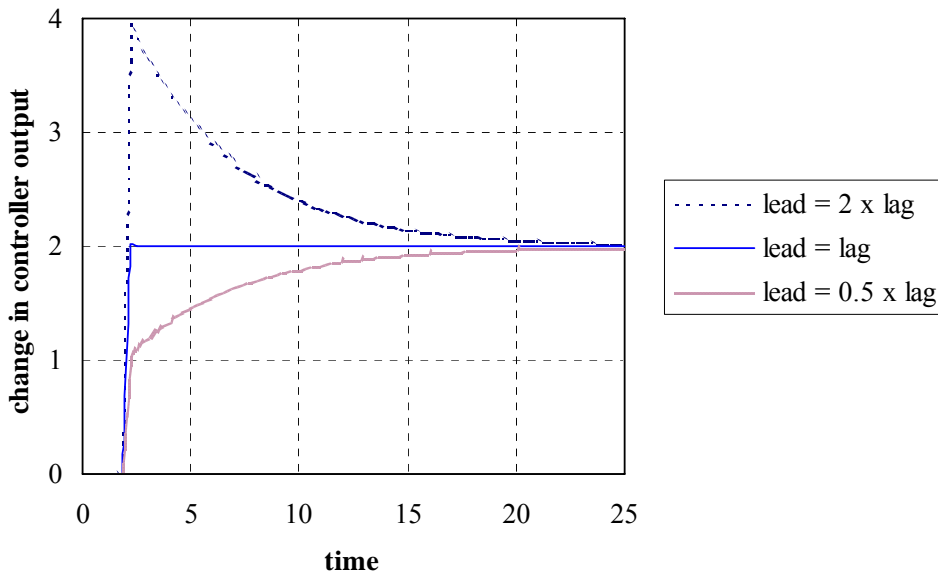
$$G_{ff} = K_{ff} \frac{T_{lead} s + 1}{T_{lag} s + 1} e^{-\theta_{ff}} \quad (8.9-2)$$

The effect of gain is to amplify the controller response to an input. The gain satisfies the steady state relationship between disturbance and manipulated variables. The effect of dead time is to delay the controller's response, so that it will not affect the controlled variable prematurely. Such a step is appropriate if the disturbance dead time exceeds that of the process. However, if the disturbance dead time is less, perfect control requires a negative  $\theta_{ff}$ , which implies predicting the onset of future disturbances! Such a time-machine would be very useful, for many purposes, but we are not likely to find one - thus the parameter  $\theta_{ff}$  would be set to zero.

We can illustrate (8.9-2) by calculating the output of the controller upon receiving a step input of magnitude A. Applying the step change and inverting, we find

$$x_{co}' = AK_{ff} \left[ 1 + \frac{T_{lead} - T_{lag}}{T_{lag}} e^{-(t - \theta_{ff})/T_{lag}} \right] \quad (8.9-3)$$

The lead/lag elements shape the development of the response, as shown in Figure 8.9-1. For example, if  $T_{lag}$  is set to be greater than  $T_{lead}$ , the controller output increases with time. This is appropriate if the disturbance propagates more slowly through the process than does the manipulated variable. If the reverse is true, then a more vigorous manipulated variable response is in order from the start, and so  $T_{lead}$  is set larger than  $T_{lag}$ .



**Figure 8.9-1** Response of controller  $G_{ff}$  to a step input

### 8.10 summing feedforward and feedback signals at one valve

Each controller specifies, at any time, how much the valve should be moved from its previous position. These two outputs are summed and then sent to the valve transducer.

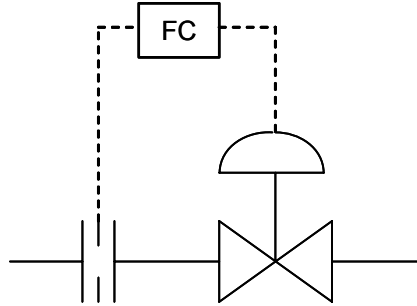
It may be that the two controllers will oppose each other, so that the sum of the outputs is a smaller movement than each individual controller would have directed. This is not necessarily a problem: each controller responds in its own area of expertise (feedback responding to present error; feedforward forecasting the effects of present disturbance) and the sum of the outputs addresses both concerns.

Of course, the sum of the controller outputs could fall outside the physical 0 - 100% range of the valve. In this case, the valve can move only to its limit in response. If this is a frequent occurrence, the manipulated variable may be insufficiently strong to overcome disturbances.

### 8.11 flow control

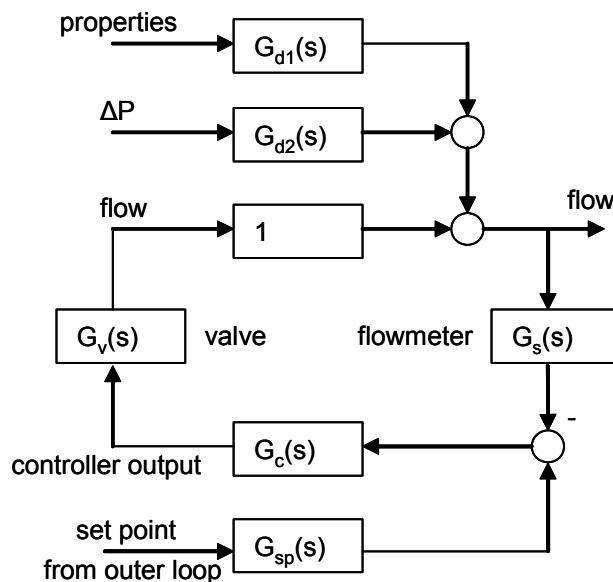
As we have already seen in examples, a very common application of a cascade scheme is to use a flow controller as an inner cascade loop. This allows the flow to be less affected by, for example, pressure differences, so that it can be a more reliable manipulated variable in the outer loops.

We examine the flow control loop. The equipment comprises a valve, a flowmeter, and connecting pipe.



**Figure 8.11-1** Equipment to measure and manipulate flow

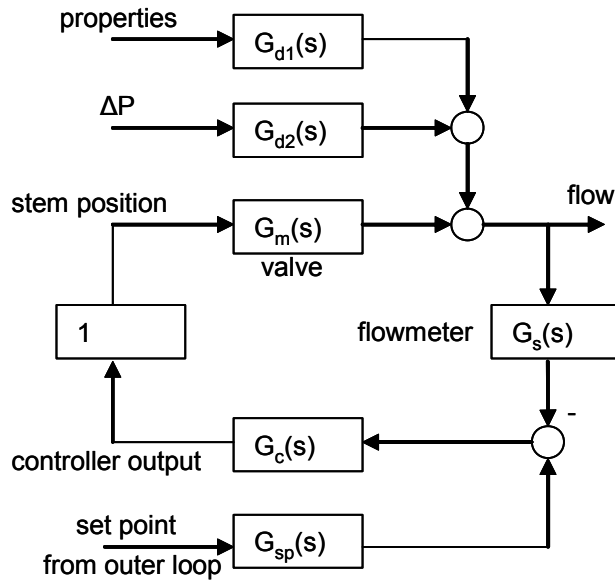
If we regard the valve as the final control element and the flowmeter as the sensor, there is really no process left. Hence a block diagram will show the process as a unity transfer function: the controlled variable and manipulated variable are identical:



**Figure 8.11-2** Block diagram of flow control loop

The valve affects the flow through the stem position, as described in transfer function  $G_v$ . The diagram also identifies two disturbances that affect flow: a change in the pressure difference across the valve (as might result from variations in sources and sink conditions) and changes in physical properties of the flowing fluid. All these transfer functions come, as in other examples, from a linearized model of flow through a valve.

After our emphasis on distinguishing manipulated and controlled variables, having the manipulated variable be identical to the controlled variable may seem peculiar. An alternative point of view is to consider the manipulated variable to be the valve stem position, as shown in Figure 8.11-3.



**Figure 8.11-3 Alternative block diagram of flow control loop**

In effect, the transfer function that describes how valve stem position affects flow has moved from  $G_v$  in Figure 8.11-2 to  $G_m$  in Figure 8.11-3. In this series of Lessons, we have preferred the point of view of Figure 8.11-2; that is, a controller acts through a final control element to produce a flow, which may then be used to manipulate a process output.

**CLOSED LOOP BEHAVIOR**

**8.12 cascade performance**

From Figure 8.5-1, we derive the transfer function for the closed loop cascade structure. With nested loops in block diagrams, it is best to begin with the inner loop.

$$\begin{aligned}
 x'_i(s) &= G_{d2}x'_{d2}(s) + G_{m2}x'_m(s) \\
 &= G_{d2}x'_{d2}(s) + G_{m2}G_vG_{c2}(x'_{col}(s) - G_{s2}x'_i(s))
 \end{aligned}
 \tag{8.12-1}$$

so that

$$x'_i(s) = \frac{G_{d2}}{1 + G_{L2}} x'_{d2}(s) + \frac{G_{m2}G_vG_{c2}}{1 + G_{L2}} x'_{col}(s)
 \tag{8.12-2}$$

where the transfer function around the secondary loop is

$$G_{L2} = G_{m2}G_vG_{c2}G_{s2}
 \tag{8.12-3}$$

With  $x_i'$  known, we continue in the outer loop.

$$\begin{aligned}
 y'(s) &= G_{d1}x'_{d1}(s) + G_{m1}x'_i(s) \\
 &= G_{d1}x'_{d1}(s) + \frac{G_{m1}G_{d2}}{1+G_{L2}}x'_{d2}(s) + \frac{G_{m1}G_{m2}G_vG_{c2}}{1+G_{L2}}x'_{co1}(s) \\
 &= G_{d1}x'_{d1}(s) + \frac{G_{m1}G_{d2}}{1+G_{L2}}x'_{d2}(s) + \frac{G_{m1}G_{m2}G_vG_{c2}}{1+G_{L2}}G_{c1}(G_{sp}y'_{sp}(s) - G_{s1}y'(s))
 \end{aligned} \tag{8.12-4}$$

Solving for  $y'$

$$y'(s) = \frac{G_{d1}(1+G_{L2})}{1+G_{L2}+G_{L1}}x'_{d1}(s) + \frac{G_{m1}G_{d2}}{1+G_{L2}+G_{L1}}x'_{d2}(s) + \frac{G_{m1}G_{m2}G_vG_{c2}G_{c1}G_{sp}}{1+G_{L2}+G_{L1}}y'_{sp}(s) \tag{8.12-5}$$

where the transfer function around the primary loop is

$$G_{L1} = G_{m1}G_{m2}G_vG_{c2}G_{c1}G_{s1} \tag{8.12-6}$$

Equation (8.12-5) is general for a two-loop cascade structure with the disturbances arranged as in Figure 8.5-1. It can be specialized by substituting particular transfer functions for the components in the cascade.

### 8.13 cascade tuning

Marlin (2000) suggests tuning the inner loop first, with the outer loop in manual setting. Then tune the outer loop with the inner loop in automatic setting.

### 8.14 cascade closed loop stability

The characteristic equation for the cascade structure, obtained from (8.12-5), contains the transfer functions for both controllers. Hence both controllers affect the poles and thus stability of the cascade.

### 8.15 feedforward performance and stability

From Figure 8.6-3, we derive the transfer function for the closed loop feedback/feedforward structure.

$$\begin{aligned}
 y'(s) &= G_{d1}x'_{d1}(s) + G_{d2}x'_{d2}(s) + G_mx'_m(s) \\
 &= G_{d1}x'_{d1}(s) + G_{d2}x'_{d2}(s) + G_mG_v(x'_{cof}(s) + x'_{co}(s)) \\
 &= G_{d1}x'_{d1}(s) + G_{d2}x'_{d2}(s) + G_mG_vG_{ff}G_{sf}x'_{d2}(s) + G_mG_vG_c(G_{sp}y'_{sp}(s) - G_{sp}y'(s))
 \end{aligned} \tag{8.15-1}$$

Solving for controlled variable  $y'$

$$y'(s) = \frac{G_{d1}}{1+G_L} x'_{d1}(s) + \frac{G_{d2} + G_m G_v G_{ff} G_{sf}}{1+G_L} x'_{d2}(s) + \frac{G_m G_v G_c G_{sp}}{1+G_L} y'_{sp}(s) \quad (8.15-2)$$

where the transfer function around the feedback loop is

$$G_L = G_m G_v G_c G_s \quad (8.15-3)$$

Notice that the feedforward controller affects only the transfer function for disturbance  $x_{d2}$ ; other disturbances and the set point have the usual feedback loop transfer functions. By (8.6-1) we try to make the  $x_{d2}$  transfer function zero. However, if it is not, the feedback controller is also available, through the transfer function denominator, to respond to  $x_{d2}$ , as it does to other disturbances.

Because the characteristic equation obtained from (8.15-2) does not depend on the feedforward controller, adding a feedforward loop has no effect on tuning the feedback controller for closed loop stability. Even so, if the feedforward controller itself is unstable, the ensemble is likely to be inoperable.

### 8.16 feedforward tuning

Ideally all the other disturbances would subside so that the feedback loop could be put in manual and the feedforward loop tuned to respond to its particular disturbance. One may not be so fortunate, however.

### 8.17 conclusion

By going to the trouble and expense of extra measurements, and obtaining deeper knowledge of the process, we are enabled to improve on the performance of the single-loop PID controller. This is not to say that cascade or feedforward enhancements are always to be recommended - they must be technically feasible (the measurements and character of the new variables being appropriate) and economically justified. This of course, is a familiar story to engineers.

Feedforward and cascade point the way to further control schemes that make use of process models. Various forms of model-based control can offer advantages in single control loops, but really come into their own when we consider how individual control loops may interact - that is, when realistic process models are MIMO - multiple input/multiple output.

### 8.18 reference

Marlin, Thomas E. *Process Control*. 2nd ed. Boston, MA: McGraw-Hill, 2000. ISBN: 0070393621.