FEATURE SERIES: PRACTICAL PROCESS CONTROL

8: Determining Tuning Constants for Tight and Averaging Level Control

***Myke King** continues his detailed series on process control, seeking to inspire chemical engineers to exploit untapped opportunities for improvement*

IN THE previous article we showed how to determine the parameters necessary to calculate level controller tuning. These are:

- **V = working volume of the vessel**
- **d = maximum acceptable deviation from setpoint (40% in our example)**
- **f = normally expected flow disturbance**
- **F = flow when the level controller output is 100%**
- **ts = level controller scan interval**

We now need to determine the values for controller gain (K_c), integral time (T_i) and derivative time (T_d).

TIGHT CONTROL

To design the controller to deliver tight control, we start with a proportional-only controller – where E is the deviation from setpoint (the error) and M the controller output.

$$\Delta M = K_c(E_n - E_{n-1})$$

Assuming that the process before the flow disturbance is at steady state, then E_{n-1} will be zero. After the disturbance, the error (in dimensionless form) is given by the change in liquid volume as a fraction of the working volume.

$$E_n = \frac{f.ts}{V}$$

For the level to stop moving, the controller must adjust the outlet flow by the same amount as the inlet flow. Again, in dimensionless form

$$\Delta M = \frac{f}{F}$$

For tight control, we design the level controller to correct the imbalance in the shortest possible time. This is at the end of the first scan interval, so the maximum controller gain is given by

$$\frac{f}{F} = K_{max} \times \frac{f.ts}{V} \qquad \text{or} \qquad K_{max} = \frac{V}{F.ts}$$

Care should be taken in consistency of engineering units. For example, if F is in m^3/h, then V must be in m^3 and ts in hours. Note that the result excludes f. Whatever disturbance is made to the inlet flow, one scan interval later, the controller will do the same to the outlet flow. With a proportional-only controller there will be an offset from the level setpoint. It will be negligibly small but, if required, integral action can be included. To determine this, we first calculate the vessel time constant (T). This is defined as the time taken, with no controller in place, for the level to reach the maximum deviation (d) when the inlet flow is changed by f. This is given by

$$T = \frac{Vd}{100f}$$

For tight level control, we choose a small value for d, say 1%. Empirically, setting the integral time (T_i) to $8T$ gives good control. We must again be careful with units. Depending on the control system, T (and hence T_i) should be in minutes or seconds. Because we have included integral action, we must slightly reduce the proportional action. Again, empirically, setting K_c to $0.8K_{max}$ works well. Only in unusual circumstances (that we'll cover in the next issue) does level control benefit from derivative action. Full controller tuning is therefore

$$K_c = \frac{0.8V}{F.ts} \qquad T_i = \frac{V}{12.5f} \qquad T_d = 0$$

Unlike most controllers, the tuning of tight level control is particularly sensitive to the scan interval. For example, if the current system scan interval is one second and is increased to two seconds, the gain of all tight level controllers should be halved.

There are other factors that might limit the maximum gain the controller will support – most notably noise. It may be necessary to reduce K_c to avoid excessive control valve movement.

AVERAGING CONTROL

We take much the same approach to designing an averaging level controller, starting with a proportional-only controller.

But this time we design the controller to operate as slowly as possible. We define the minimum controller gain (K_{min}) as that which will leave an offset of *d*. In other words, the level will move to the nearest alarm limit and stay there. By doing so, we use all the available surge capacity. Clearly not a practical controller, we will later add integral action to bring the level slowly back to setpoint.

As in the tight control case, the averaging controller must match the outlet flow to any change in inlet flow. But, unlike the tight controller, this takes place slowly – involving a large number of controller scans.

$$\Delta M = K_c[(E_n - E_{n-1}) + (E_{n-1} - E_{n-2}) \ldots\ldots + (E_1 - E_0)] = K_c(E_n - E_0)$$

$$E_0 = 0 \qquad E_n = \frac{d}{100}$$

$$\frac{f}{F} = \frac{K_{min} d}{100} \qquad \text{or} \qquad K_{min} = \frac{100f}{F.d}$$

We apply the same empirical method, as we did for tight control, to give full controller tuning as

$$K_c = \frac{80f}{F.d} \qquad T_i = \frac{V.d}{12.5f} \qquad T_d = 0$$

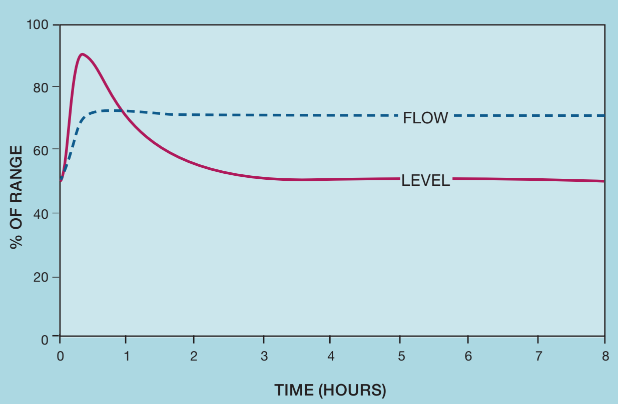
Figure 1 shows how this controller responds. In this example, avoiding the alarm at 90%, it takes around 30 minutes for the downstream flow to reach its new value, as opposed to the few seconds taken with tight control. This will substantially improve the stability of any downstream unit. It does, however, require that the level be away from its setpoint for over an hour; it is this that the process operator might find difficult to accept.

Note that, unlike tight control, this tuning is unaffected by controller scan interval. It does, however, depend on the value chosen for *f*. Of course, not all flow disturbances will be the same size; we'll address this issue next.

Less common these days due to digital control systems, the value calculated for T_i may exceed the maximum permitted. Remembering T_i is the denominator, this is telling us that the controller is close to being proportional-only. Indeed, such a controller will work well, but the permanent offset can make operator acceptance difficult. To ensure the maximum deviation is never violated

$$K_c = \frac{50}{d}$$

Figure 1: Averaging level control



ERROR-SQUARED

The *error-squared* algorithm is available in most control systems. As its name suggests. the PID algorithm uses the square of the error; actually, it uses $E|E|$ because we need to retain the sign. Remembering that the controller works in a dimensionless form, error is ranged -1 to +1, or -100% to +100%. Squaring the error doesn't affect these ranges but it does introduce non-linearity. For example, the result of squaring 10% is 1%. This means that the controller will hardly respond to this error, whereas its response to an error of 100% is unaffected.

There are many versions of the error-squared algorithm but the most common, instead of squaring each individual error in the control equation, multiplies the controller gain by $|E|$. The effective controller gain is now proportional to the error, reducing to zero as the error reduces to zero.

$$\Delta M = K_c|E|\left(E_n - E_{n-1} + \frac{ts}{T_i}E_n \ldots\right)$$

This algorithm offers no advantage for tight level control but is tuned for averaging level control using the same approach as that for the linear version. Full tuning then becomes

$$K_c = \frac{80f}{F.d}\left[\frac{200}{d}\right] \qquad T_i = \frac{V.d}{12.5f} \qquad T_d = 0$$

To understand the advantage of this algorithm, we must explore how it and the linear version handle disturbances that are different from the design. Figure 2 shows, for both controllers, the effect of a disturbance 25% larger and 25% smaller than *f*. Disturbances larger than design are handled better with less time spent violating the alarm at 90%. Those smaller are also handled better, in that more of the surge capacity is used. The algorithm is *adaptive*; it is adapting to the size of the error.

However, error-squaring has a disadvantage. As Figure 3 shows, it exhibits oscillatory behaviour as the level approaches setpoint. This is because the effective controller gain falls to zero; small deviations are largely left unchecked until

Figure 2: Comparison for non-design disturbances

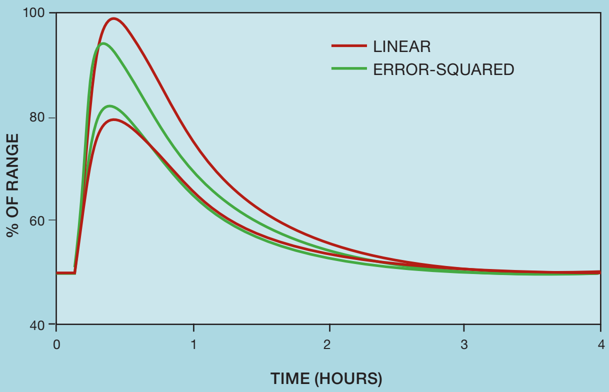
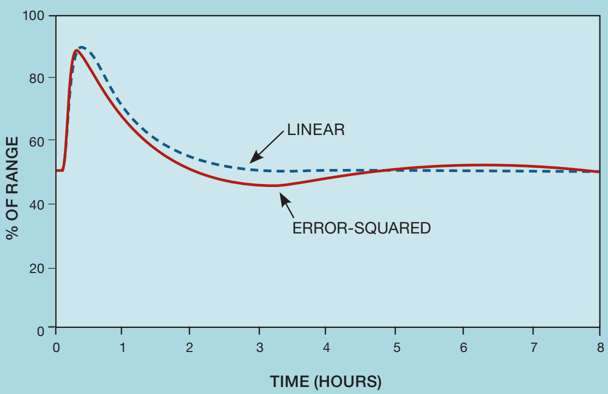


Figure 3: Comparison between error-squared and linear



they become significant. The effect on the manipulated flow is minor and has little impact on the stability of the downstream process. However, it makes persuading a sceptical process operator to accept averaging control more difficult. The solution lies in the way in which control system vendors offer the algorithm. Foxboro, for example, adds a parameter (C) to the conventional PID algorithm.

$$\Delta M = K_c(C[E_n] + 1 - C)\left(E_n - E_{n-1} + \frac{ts}{T_i}E_n \ldots.\right)$$

Setting C to 0 gives the original PID algorithm. Setting it to 1 gives error-squared. But intermediate values are permitted. By choosing a value of 0.95, we get the adaptive benefit of error-squared but without the problem of the controller gain falling to zero as the level approaches setpoint. The calculation of controller gain then changes to

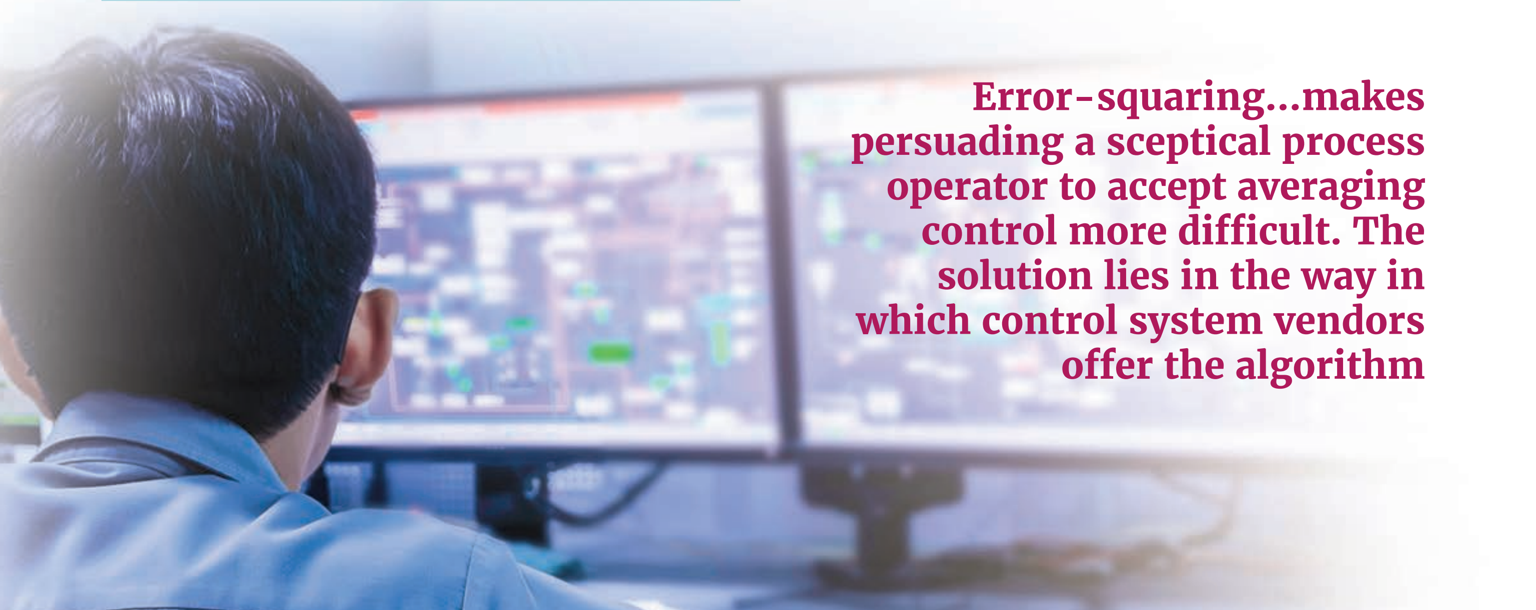
$$K_c = \frac{80f}{F.d}\left[\frac{200}{200(1-C)+Cd}\right]$$

Honeywell takes a similar approach.

$$\Delta M = (C + K_n[E_n])\left(E_n - E_{n-1} + \frac{ts}{T_i}E_n \ldots.\right)$$

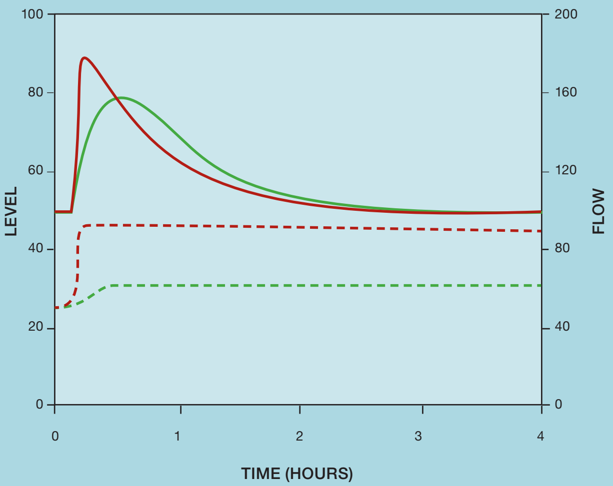
Setting C to 1 and K_n to 0 gives the linear PID; setting C to 0 and K_n to 1 gives the error-squared version. However, while K_n can have any value, intermediate values for C are not permitted. So, if we want 95% error-squared behaviour, we set C to 1 and choose C^* as 0.95.

$$K_n = \frac{C^*}{1-C^*} \qquad K_c = \frac{80f}{F.d}\left[\frac{200}{200C + K_n d}\right]$$



Error-squaring…makes persuading a sceptical process operator to accept averaging control more difficult. The solution lies in the way in which control system vendors offer the algorithm

Figure 4: Use of gap controller



GAP CONTROL

Another algorithm commonly available in control systems is the *gap controller*. In its basic form we define gaps either side of the setpoint. If the level is within these gaps, no control action is taken. We have created a *deadband* in which the controller gain is zero. To fully utilise surge capacity, the gaps, like the alarms, should be equidistant either side of the setpoint. The algorithm can be tuned to deliver effective averaging control but exhibits oscillatory behaviour similar to the error-squared algorithm. But, instead of a sinusoidal wave, it has a sawtooth shape – with amplitude equal to the deadband. While it still provides effective level control, it is likely to be even less acceptable to the operator. The solution, as before, is to avoid the controller gain falling to zero by configuring a small gain, $(K_c)_{gap}$, to be used with the gap. However, in many circumstances, it offers no advantage over the error-squared algorithm.

Its value is in a situation where flow disturbances are consistently small for most of the time but include the occasional spike. This might be caused by some relatively infrequent, but substantial change in the operation. Typically, this is associated with routine switching of equipment such as driers or reactors. With the algorithms we've covered so far, we would have to design for the worst disturbance and accept that, for most of the time, we will underutilise surge capacity. Gap control offers a better solution. The principle is to design the controller using two values for f (f_1 for the smaller frequent disturbances and f_2 for the large infrequent disturbance). The smaller disturbances are dealt with by keeping the level within the gap; larger ones use the remaining capacity. Typically, we choose the gap (G) to be around 75% of the maximum deviation (d). The calculation of integral time is unchanged; the controller gains are given by

$$K_c = \frac{80(f_2 - f_1)}{F(d - G)} \qquad (K_c)_{gap} = \frac{80 f_1}{F.G}$$

In some control systems, the gain used within the gap is configured as a multiplier (K_r) applied to K_c. This is given by

$$K_r = \frac{f_1(d - G)}{G(f_2 - f_1)}$$

Rearranging

$$G = \frac{f_1 d}{(1 - K_r)f_1 + K_r f_2}$$

As a rule of thumb, we would expect K_r to be around 0.1 and so G is determined as

$$G = \frac{f_1 d}{0.9 f_1 + 0.1 f_2}$$

Figure 4 shows its performance. The alarms are set at 10 and 90%, with the SP at 50%, so d is 40%. From the increases made to the manipulated flow (dashed lines) we can see that f_2 is 40 (from 50 to 90), while f_1 is 10 (from 50 to 60) – so G was set to 30%. For the larger disturbance, the level (solid line) stays within the 90% alarm limit. Despite being a quarter of the size, the smaller disturbance uses three-quarters of the surge capacity, with the level peaking at 80%. ■

NEXT ISSUE

While these tuning calculations will work well on most vessels, controller performance can be poor if there is a highly non-linear relationship between volume and level. In the next article we'll show how level instrumentation should be designed to avoid this. And, for vessels already in service, we'll show how to modify the controller to accommodate non-linearity.

We'll also cover level controllers that show deadtime or lag, showing how they should be tuned. And we extend the approach to cover all integrating processes.

Myke King CEng FIChemE is director of Whitehouse Consulting, an independent advisor covering all aspects of process control. The topics featured in this series are covered in greater detail in his book Process Control – A Practical Approach, published by Wiley in 2016
