28: Solutions for Long Deadtime

Myke King describes the techniques available to control processes that have long deadtimes

DEADTIME, also known as transport delay, exists in most processes. But in most it is small compared to the process lag.

Figure 1 shows how much the controller gain (K_c) must be reduced as the deadtime-to-lag ratio (θ/τ) increases. So, for example, if the deadtime is half the lag, to prevent the controller from being oscillatory we need to reduce the controller gain by about 50%. This means, of course, that it will take twice as long to respond to setpoint changes and twice as long to resolve any disturbance. This problem becomes severe as deadtime exceeds lag – requiring a 90% reduction in controller gain.

Such a situation is common in the mineral processing industry, where conveyors introduce long deadtimes. But it can occasionally occur in industries such as oil and petrochemicals. For example, on hydrodesulphurisation units, it is usual for there to be a delay of around 90 minutes between changing the setpoint of the reactor temperature controller to correct an off-grade product and the resulting change in product sulphur content. Even longer delays can exist in distillation columns – particularly those separating isomers and so requiring a very large number of trays. Lube oil treating plants have similar issues.

QUICK READ

- **Deadtime limits control performance:** As the deadtime-to-lag ratio rises, controller gain must be reduced – up to 90% when deadtime exceeds lag – slowing response to setpoint and disturbance changes
- **Compensation algorithms restore stability:** Methods such as PI with integral delay, Smith predictors and IMC counteract delay by modelling process dynamics
- **Accuracy and adaptation are vital:** These techniques outperform standard PID control but are sensitive to modelling errors; adaptive tuning helps maintain optimal performance

PI WITH INTEGRAL DELAY

Many DCS (distributed control system) include a conventional proportional plus integral (PI) algorithm with the option to delay the integral action. In addition to the conventional tuning constants, it includes θ – the time by which the integral action is delayed. Once the process dynamics have been determined, in principle, tuning the algorithm is straightforward. The controller gain (K_c) is set to the reciprocal of the process gain (K_p). The integral time (T_i) is set to the process lag (τ) and θ is set to the process deadtime. Figure 2 shows how the algorithm responds to a change in setpoint. As usual, the proportional action multiplies the change in setpoint by K_c, so therefore divides it by K_p. By definition ΔPV is $K_p \Delta MV$ so, after the process deadtime has elapsed, the PV will change by exactly the required amount, with no further action required. However, because it is

Figure 1: Need to reduce controller gain

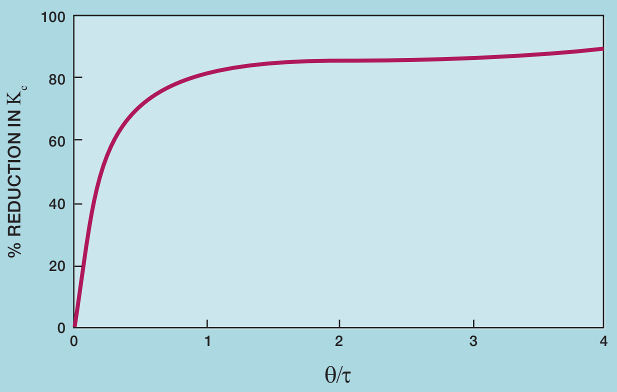


Figure 2: PI algorithm with integral delay

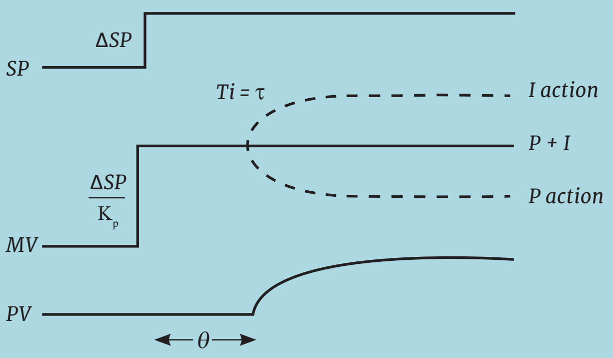


Figure 3: Smith predictor

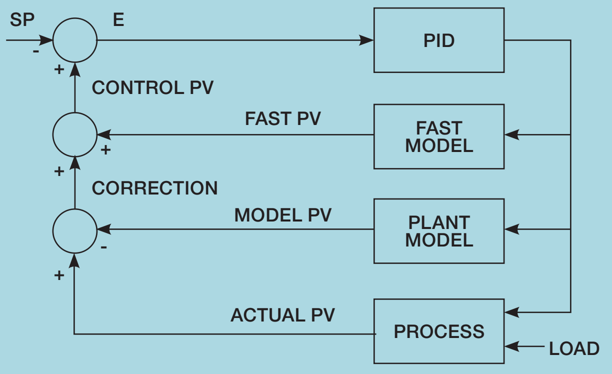
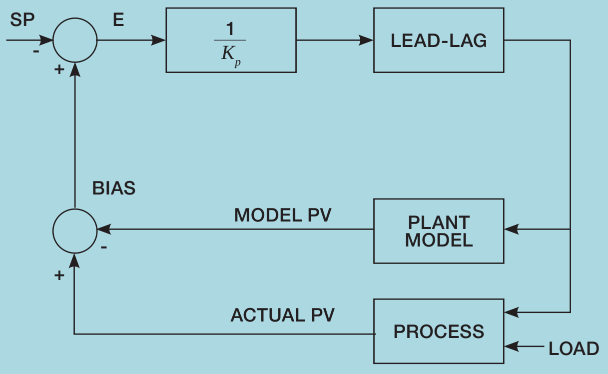


Figure 4: Internal model control



unlikely that we have perfectly estimated K_p (or there may be a process disturbance), then ongoing correction will be necessary. This, as usual, is the role of the integral action – but now delayed by θ. However, in response to the original proportional kick, the PV will begin changing at the same time. The resulting reduction in error will cause proportional action but, if our tuning is perfect, it will be cancelled out by the integral action. If not, their combination will correct any remaining error in the conventional way.

We have effectively installed a "step and wait" controller that emulates what a person would do. But it has been tuned to make exactly the right step and wait the correct time before compensating for any discrepancy in the predicted outcome. The algorithm is available within several DCS – notably those from Emerson and Foxboro. Taking Foxboro as an example, in which it is known as the PITAU algorithm, it is selected by setting the MODOPT parameter to 7 or 8. It is also included in many programmable logic controllers (PLC), such as those from Siemens and Allen Bradley.

SMITH PREDICTOR

Probably the most common textbook deadtime compensation technique is the Smith predictor. It uses a conventional PID controller to regulate a model of the process – specifically, a model with zero deadtime. Figure 3 shows the overall configuration. The fast model comprises the process gain (K_p) and the process lag (τ). As always, the model will not be perfect, so we have to correct for any deviation. So, in addition, we include the process model. This is the same as the fast model but also includes the process deadtime (θ). The output of this model is compared to the real process. The difference is the prediction error, which is added to the PID controller's PV, so that it corrects the resulting deviation from setpoint. Because the PID controller "thinks" the process deadtime is zero, it can be tuned to have a much larger controller gain and so respond more quickly to both setpoint and load changes. Theoretically the gain is limited only by any restriction on the MV overshoot although, in practice, the resulting amplification of small modelling errors may be the first to limit. It can be shown that, if set up with the same tuning constants as the PI with integral delay, the Smith predictor will give exactly the same response. However, it is likely to require custom configuration and need only be applied if the PI with integral delay is not included as standard within the DCS.

INTERNAL MODEL CONTROL

IMC includes aspects of both techniques described so far. As shown in Figure 4, it makes a step change sized to cause the PV to exactly reach the setpoint, once the deadtime expires. Model inaccuracy again is determined as the difference between the process and the plant model, with the deviation used to bias the setpoint. We covered the lead/lag algorithm in *TCE 999*. Here, its lead term (T1) is set to the process lag (τ) and so cancels it out. The engineer can the set the lag term (T2) to give the desired trade-off between a fast approach to setpoint and MV overshoot. If set equal to T1 the PV will approach the setpoint with the same trajectory as the open loop response, ie with a lag of τ. If T2 is set greater than T1 the MV overshoot will be reduced and the setpoint approached more slowly. Or T2 can be set smaller than T1 for a more rapid approach.

Probably the most common textbook deadtime compensation technique is the Smith predictor. It uses a conventional PID controller to regulate a model of the process – specifically, a model with zero dead time

DAHLIN ALGORITHM

This algorithm was developed for use when the controller scan interval is significant when compared to the process dynamics. While the scan interval is set by the DCS, certain on-stream analysers generate new measurements much less frequently. The DCS scan interval might be one or two seconds, while the update interval for, say, an on-stream chromatograph will be several minutes. There are advantages to having the controller only act when the analyser generates a new measurement. The analyser's "read-now" signal is used as the trigger. Provided the chosen tuning method takes account of scan interval, this will work well. Its main advantage is that action is only taken if the analyser is working properly. Failure would result in the measurement freezing at the last good value. If undetected the controller would then ramp its output in a vain attempt to reach the setpoint. However, the disadvantage is that the analyser interval is not necessarily constant. The analyser will run through a sequence that might include purging, cooling, heating etc. Rather that triggered by a timer, some of the stages may complete when a desired condition is met. The advantage of the Dahlin algorithm is that, provided the update time interval is recorded, it can used to modify the controller tuning constants.

We recall, from *TCE 982*, the PID controller:

$$\Delta MV = K_c\left[(E_n - E_{n-1}) + \frac{ts}{T_i}E_n + \frac{T_d}{ts}(E_n - 2E_{n-1} + E_{n-2})\right]$$

This can be rewritten as:

$$MV_n = c_0E_n + c_1E_{n-1} + c_2E_{n-2} + d_1MV_{n-1} + d_2MV_{n-2}$$

where:

$$c_0 = K_c\left[1 + \frac{ts}{T_i} + \frac{T_d}{ts}\right] \qquad c_1 = -K_c\left[1 + \frac{2T_d}{ts}\right]$$

$$c_2 = K_c\frac{T_d}{ts} \qquad d_1 = 1 \qquad d_2 = 0$$

Written in the same form, the Dahlin algorithm includes, as the last term, the value of the MV which is responsible for the current value of the PV. This is the value collected N scan intervals previously, where N is θ/ts:

$$MV_n = c_0E_n + c_1E_{n-1} + d_1MV_{n-1} + d_{N+1}MV_{n-(N+1)}$$

where:

$$c_0 = \frac{1 - e^{\frac{-ts}{\lambda}}}{K_p\left[1 - e^{\frac{-ts}{\tau}}\right]} \qquad c_1 = \frac{-e^{\frac{-ts}{\lambda}}\left[1 - e^{\frac{-ts}{\lambda}}\right]}{K_p\left[1 - e^{\frac{-ts}{\tau}}\right]}$$

$$d_1 = e^{\frac{-ts}{\lambda}} \qquad d_{N+1} = 1 - e^{\frac{-ts}{\lambda}}$$

A potential problem with all these techniques is that they are particularly sensitive to error in the estimation of deadtime. With the conventional PID algorithm we can tolerate errors of up to around 20% in the process dynamics without noticeable degradation in the performance of the controller

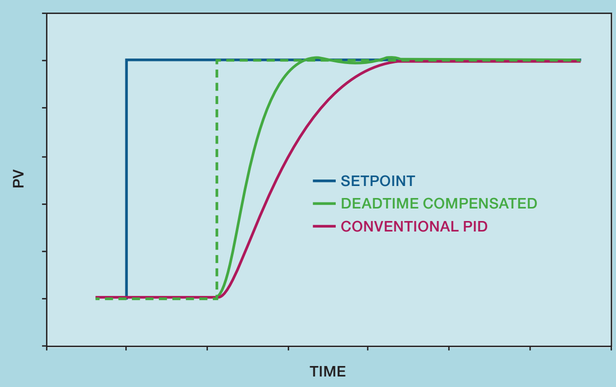
The term (λ) serves the same purpose as T2 in the IMC controller, ie it is an engineer-adjustable tuning constant that determines the trajectory of the approach to setpoint. Once set, the *c* and *d* coefficients can be automatically updated to account for any change in the controller scan interval (ts).

Unlike the techniques we covered previously this has to be custom-coded in the DCS, rather than configured using standard blocks. But, once built, can be readily cloned for multiple applications.

MODELLING ERROR

Figure 5 shows the advantage of these deadtime compensation algorithms compared to the use of the conventional PID controller. No controller can persuade to PV to change until the

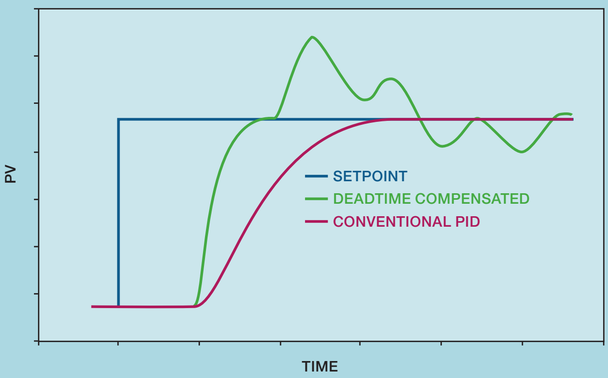
Figure 5: Advantage of deadtime compensation



deadtime has elapsed. So, the dashed line might be considered as perfect control, but this could only be achieved with excessive MV overshoot. The properly tuned compensated controller outperforms the optimally tuned conventional PID controller. The time to reach setpoint (after the deadtime has elapsed) is halved.

A potential problem with all the above techniques is that they are particularly sensitive to error in the estimation of deadtime. With the conventional PID algorithm we can tolerate errors of up to around 20% in the process dynamics without noticeable degradation in the performance of the controller. Figure 6 shows the impact of such a change. The conventional PID control is far more robust. The solution for all the deadtime compensation techniques is to apply adaptive tuning, where the value of the deadtime used by the controller is continuously updated. The actual deadtime can often be estimated from a key operating parameter. For example, in mineral processing, it could be derived from knowing the speed and length of the conveyor. For a desulphurisation unit it should be possible to derive, from plant tests, a correlation between deadtime and feed rate. ■

Figure 6: Impact of modelling error



NEXT ISSUE

An audit conducted by one of the leading control system suppliers identified that problems with the final control element were one of the most frequent causes of poor controller performance. In the next issue we'll describe the more common problems and how to diagnose them. While there is a wide range of software products that can assist with this, they are costly and include functionality that perhaps is not required, and which can be daunting to the inexperienced user. The data they collect is readily available in the data historian, now commonplace on most processes. Analysis and reporting can then be readily configured using, for example, MS Excel. The article will suggest what techniques the engineer might use – either as an alternative for an off-the-shelf product or as a benchmark against which a product can be selected.

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