FEATURE SERIES: PRACTICAL PROCESS CONTROL

12: Filters

Myke King explains filters and the benefit of moving away from the standard technique

MANY process measurements are subject to noise. Liquid levels can be turbulent; orifice flow meters will show noise if the flow is mixed phase and pressures may reflect vibration from compressors. Ideally, good process and instrumentation design should have eliminated noise at source. Failing this, filtering can be applied to the measurement prior to its use in a controller. Filtering is an example of signal conditioning.

Implementing a filter, that is a standard feature of the distributed control system (DCS), is a compromise between noise reduction and measurement distortion. Strong filters are effective at noise reduction but can increase both the apparent deadtime and the overall process lag. If these changes are significant, then we must change to slower controller tuning. Typically, this would be required if the deadtime increases by more than 10% or the lag by more than 20%.

On many processes, the degree of filtering typically employed is excessive. Filters are added (wrongly) to make measurement trends look smooth. The criterion that should be used is the amplitude of the signal sent to the actuator – usually a control valve, where excessive valve travel can cause mechanical problems. For example, if the controller gain (K_c) is less than 1 and there is no derivative action (T_d = 0), then the integral action will reduce the noise, potentially to an acceptable amplitude, without the need for filtering.

FIRST ORDER EXPONENTIAL

All control systems have the option to filter any measurement – usually by applying the *first order exponential* filter. First order, because it introduces a single lag (τ_f) and exponential because of the digital approximation. In general, it takes the form

$$Y_n = P.Y_{n-1} + (1-P)X_n \qquad \text{where } P = e^{-ts/\tau_f}$$

X_n is the current raw measurement, while Y_n is the current filtered value and *ts* the scan interval. The filter is *recursive*, in that it uses previous filtered values (in this case just Y_{n-1}), to determine the current value. Setting *P* to 0 disables the filter, setting it to 1 blocks any change in the raw measurement. *P* is selected to give the required level of noise reduction. Figure 1 shows that its effect is approximately linear.

In many systems the engineer will set *P* directly and is usually free to do so over its whole range. Yokogawa, however, limits the selection to 0, 0.5, 0.75 and 0.85. In other systems the engineer sets τ_f. In Honeywell's DCS it is the parameter *TF* which is entered in minutes. In ABB's it is T_{fil} – entered in seconds. The issue with this approach is that *P*, and hence the level of noise reduction, then depends on the controller scan interval (*ts*). There are many examples of this being altered, usually associated with a control system upgrade, and the resulting change in noise reduction surprising the engineer. Figure 2 illustrates this. For example, upgrading Honeywell's TDC2000 to one of its

Figure 1: Performance of exponential filter

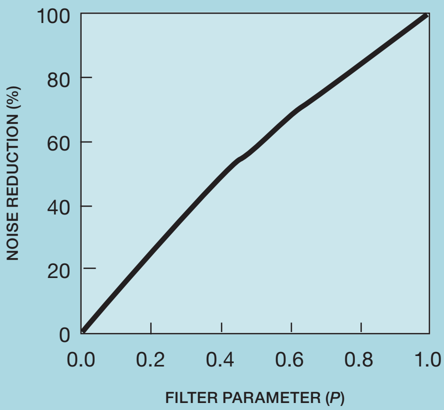
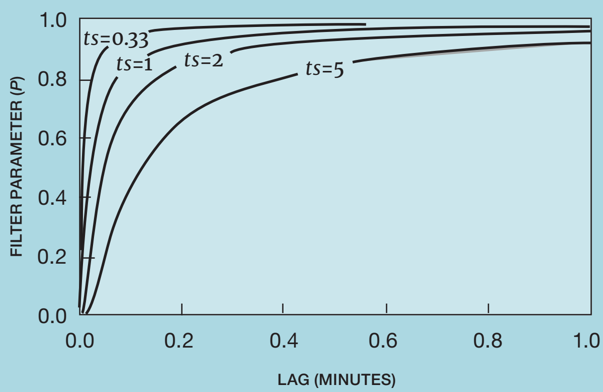


Figure 2: Effect of scan interval on filter performance



later systems increases the scan interval from 0.33 seconds to 2 seconds. For a 90% noise reduction, *TF* will have been set to 0.05. The change in scan interval reduces noise reduction to around 50%. To compensate, *TF* must be increased to 0.3. In Foxboro's *analog input block* (AIN), τ_f is defined by *FTIM*, which is entered in minutes. But Foxboro offers three options, selected by defining the parameter *FLOP*. Setting this to 1 selects the conventional exponential filter, except that its formulation is slightly different

$$Y_n = \frac{\tau_f}{\tau_f + ts} Y_{n-1} + \frac{ts}{\tau_f + ts} X_n \qquad \because\ e^{-\frac{ts}{\tau_f}} \approx \frac{1}{1 + \frac{ts}{\tau_f}} = \frac{\tau_f}{\tau_f + ts}$$

Filtering has been around for much longer than digital control. Most analog instrumentation includes the equivalent of the exponential filter in the field transmitter (for example, this would be a RC network in an electronic analog transmitter). The problem is that this might be adjusted locally, by the instrument technician, unbeknown to the engineer in the control room. This will alter the apparent process dynamics and, if a significant change, will cause control problems. Most sites have procedures that permit such changes only in the control system.

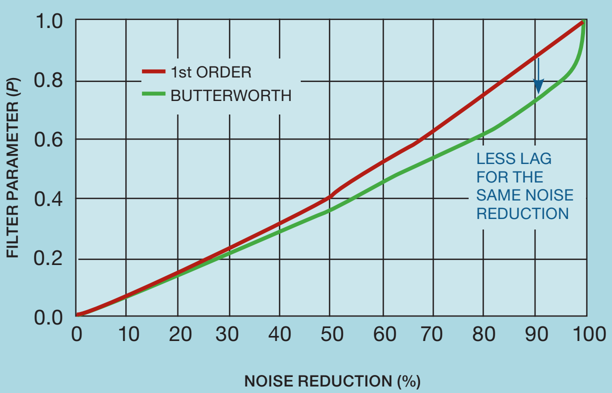
BUTTERWORTH

In the Foxboro DCS, setting *FLOP* = 2 selects the *Butterworth* filter. In general, Butterworth filters can have an order much greater than 1. That in the Foxboro DCS is second order. The approximation used is

$$Y_n = \frac{2\tau_f}{\tau_f + ts} Y_{n-1} - \frac{\tau_f}{\tau_f + 2ts} Y_{n-2} + \frac{2ts^2}{(\tau_f + ts)(\tau_f + 2ts)} X_n$$

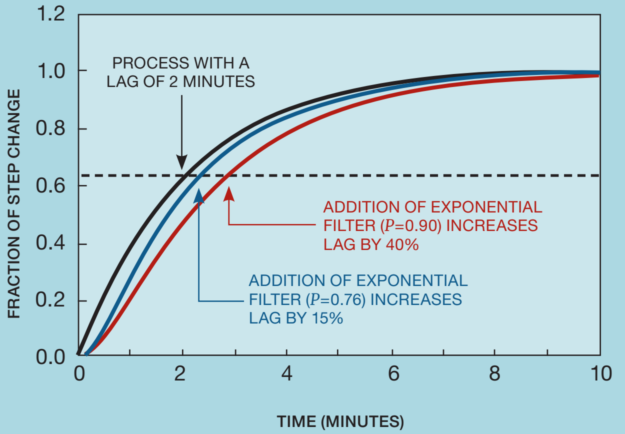
Figure 3 illustrates the advantage over the first order filter. To achieve a noise reduction of 90%, the first order filter requires a value of about 0.9 for *P*. The Butterworth requires a value of

Figure 3: Relative filter performance



On many processes, the degree of filtering typically employed is excessive. Filters are added (wrongly) to make measurement trends look smooth

Figure 4: Advantage of Butterworth



about 0.76. This introduces less additional lag. Figure 4 shows the effect of adding these filters to a process that has a lag of two minutes. Using the 63% response time as a measure of lag, the first order filter increases this by about 40% while, for Butterworth, the increase is much less, at 15%. So, if the exponential filter were added, the controller would require retuning while, with the Butterworth, it would not.

MOVING AVERAGE

Setting *FLOP* = 3 in the Foxboro DCS selects the simplest moving average filter, using just the last two raw measurements.

$$Y_n = \frac{X_n + X_{n-1}}{2}$$

Although not a standard feature in other DCS, it is possible to code a moving average filter relatively easily. Its tuning parameter (*N*) is simply the number of historical values in calculating the average. So

$$Y_n = \frac{1}{N}\sum_{r=1}^{N} X_{N-r+1} = B_1 X_n + B_2 X_{n-1} \ldots\ldots + B_N X_1 \qquad \text{where all } B_r = \frac{1}{N}$$

Figure 5 shows its effectiveness. Even with *N* set to the minimum value of 2, noise is halved. In terms of its effect on

Figure 5: Effect of N

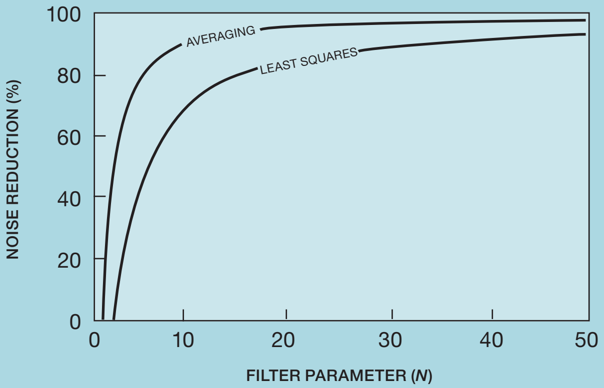
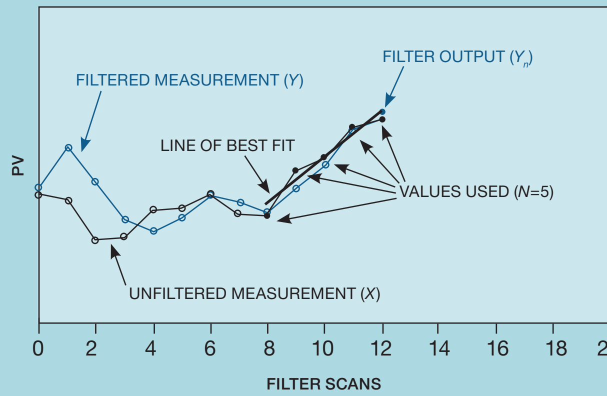


Figure 6: Least squares filter



process dynamics (following a step change in input), rather than an exponential approach to steady state, the moving average filter produces a ramp of duration $N \times ts$. However, when added to the process lag, the result is indistinguishable from that with the exponential filter. Curve fitting shows we can approximate the filter lag time constant as $0.53N \times ts$. Given that it is not usually a standard feature of most DCS, the moving average filter offers no advantage over the exponential filter.

LEAST SQUARES

Least squares regression is a technique usually applied to curve fitting. In this case, we deduce the line of best fit, to the last N raw measurements, as a means to predict what the next measurement will be. Also known as LOESS (locally estimated scatterplot smoothing), Figure 6 illustrates the principle. Although the arithmetic of least squares regression is too complex for use in real time, for this application, it reduces to the equation that we used for the averaging filter (shown above). But, instead of each coefficient being the same (as $1/N$), they are calculated from

$$B_r = \frac{4(N+1) - 6r}{N(N+1)}$$

Note that N must be larger than 2 for filtering to take place. With only two historical values, the line of best fit will pass through both and the predicted measurement will be the same as the current measurement ($B_1 = 1$ and $B_2 = 0$).

Figure 5 shows that, to achieve the same level of noise reduction, N needs to be larger than that for the moving average filter. One might think that this would increase the filter lag. But the predictive nature of the filter adds *lead*. We will cover lead in a future article on feedforward control, but it can be thought of as a means of cancelling out lag. So, using this filter can actually reduce the apparent process lag. Figure 7 illustrates this, again considering a process with lag of two minutes. With N set to 25, the filtered response is almost identical to that of the raw measurement. As we increase N (say to 75) the filtered measurement reaches steady state ahead of the raw measurement.

NUMBER OF HISTORICAL VALUES

One might be concerned, when dealing with particularly noisy measurements, about implementing custom filters that use measurements that are maybe around 100 scan intervals old. However, we can show that the standard exponential filter does the same. If we consider the last two scans

$$Y_n = P.Y_{n-1} + (1-P)X_n$$

$$Y_{n-1} = P.Y_{n-2} + (1-P)X_{n-1}$$

Combining gives

$$Y_n = P^2.Y_{n-2} + (1-P)X_n + P(1-P)X_{n-1}$$

Going back one more scan

$$Y_{n-2} = P.Y_{n-3} + (1-P)X_{n-2}$$

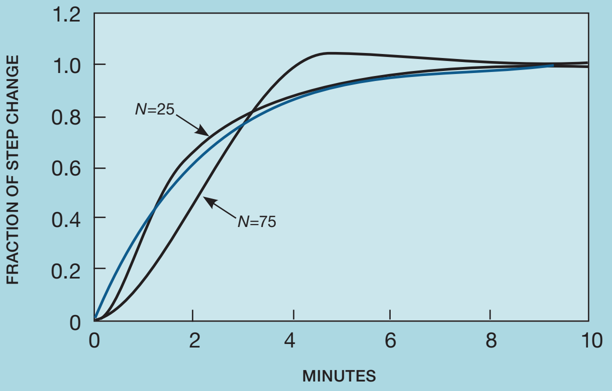
Combining gives

$$Y_n = P^3.Y_{n-3} + (1-P)X_n + P(1-P)X_{n-1} + P^2(1-P)X_{n-2}$$

As we extend this expansion, since P is less than 1, the first term will become negligible. The filter will then comprise the same formula as that used for the moving average and least squares filters, where

$$B_r = P^{r-1}(1-P)$$

Figure 7: Advantage of least squares



This formula is used to develop Table 1. For comparison, choosing N as 100, the table also shows the coefficients for the moving average and least squares filters. These coefficients must sum to 1, otherwise the filter would change the process gain. The table shows that, to achieve this, the exponential filter requires measurements older than those used by the others.

DERIVATIVE ACTION

Measurement noise is often used as an excuse not to include derivative action in the controller. Noise can always be sufficiently attenuated for derivative action to be used. However, the commonly used exponential filter can prove counter-productive, because its adverse impact on the process dynamics requires the controller gain to be reduced. Applying the least squares filter should not only resolve this but likely allow derivative action to be increased.

We recall that the benefit of derivative action is that it allows the controller gain to be substantially increased. So, to properly tune the controller, we must first obtain the process dynamics with the filter in place and use these to determine the tuning parameters. This can be achieved using tuning software mentioned in *TCE 983*. This also includes the facility to show the effect of measurement noise. ■

NEXT ISSUE

Our next article will cover other applications of signal conditioning. In particular, we'll describe a number of linearisation techniques. In situations where the process gain can change by more than the acceptable 20%, these can much improve the performance of regulatory controls.

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

Table 1: Filter coefficients

EXPONENTIAL (P=0.95)			AVERAGING (N=100)		LEAST SQUARES (N=100)	
r	B_r	$\sum_{r=1}^{r} B_r$	B_r	$\sum_{r=1}^{r} B_r$	B_r	$\sum_{r=1}^{r} B_r$
1	0.05000	0.05000	0.0100	0.0100	0.0394	0.0394
2	0.04750	0.09750	0.0100	0.0200	0.0388	0.0782
3	0.04513	0.14263	0.0100	0.0300	0.0382	0.1164
4	0.04287	0.18549	0.0100	0.0400	0.0376	0.1541
5	0.04073	0.22622	0.0100	0.0500	0.0370	0.1911
6	0.03869	0.26491	0.0100	0.0600	0.0364	0.2275
7	0.03675	0.30166	0.0100	0.0700	0.0358	0.2634
.	.	.	.	.	.	.
96	0.00038	0.99273	0.0100	0.9600	−0.0170	1.0741
97	0.00036	0.99309	0.0100	0.9700	−0.0176	1.0564
98	0.00035	0.99344	0.0100	0.9800	−0.0182	1.0382
99	0.00033	0.99377	0.0100	0.9900	−0.0188	1.0194
100	0.00031	0.99408	0.0100	1.0000	−0.0194	1.0000