Economic pressures are dispersing machine intelligence away from centralized computers toward distributed Fieldbus devices. Simultaneously, the control concept is being extended to include other operational factors such as quality assurance, process data management, just-in-time maintenance, and plant safety and availability. As credible sensing and actuation are essential prerequisites for advanced control, validation of the sensor and actuator interface is necessary. As such subsystems are linked to a range of different overall plant systems, we must consider generic validation and its reporting to the “next level up.” This article discusses validation features which are best embedded in local devices.

**Introduction**

My first intimation of the need for validation arose from industrial trials which applied self-tuning control [7] to batch chemical reactors. We achieved consistent performance over three different reactors and 11 different products (each batch involved $10^3$ liters of styrene-butadiene polymer latex—better known as carpet underlay), with a projected 18% increase of productivity. Pete Brown was in the control room during one run (see Fig. 1) where the recorded batch temperature (the principal controlled variable) had been remarkably constant for some time, and he decided to visit the plant. Nearing the reactor, he noticed that the local pressure dial was in the red danger region. Simultaneously the bursting disc ruptured and the contents of the reactor spewed over the floor, freezing as it went. What had happened?

At another company Patrick Thomas surveyed thermocouple wells at a site with about 100 batch reactors. Poor installation was found in about 85% of cases, with causes such as “well filled with rubbish,” “length of thermocouple insufficient to reach tip of well,” and “poor contact insufficient to ensure good heat transfer.” The extra thermal lag due to the poor installation caused oscillations in batch temperature; the report concluded that elimination of oscillation would lead to 6% savings in steam consumption. At a third location the self-tuning control team was working on another batch reactor. One run was going very badly, having anomalous parameter estimates. The process operator obtained a large sledgehammer and struck the control valve, whereupon effective control was resumed. A recent survey [3] indicated that 30% of all control loops in Canadian paper mills were oscillating because of valve problems.

Hence it appears that the potential benefit of much advanced control is made inconsequential because of bad actuators and sensors. Events such as the above—implying drastic changes in dynamics—invalidate the models which drive the new control designs. Unfortunately, economic pressures are de-skilling the workforce which maintains the sensors and actuators. Hence the manufacturers of those devices must provide “validation”; mechanisms by which the credibility of process interfaces can be automatically measured. That this is important is apparent from statements such as Ted Higham’s “between half and three-quarters of shut-down time is due to false alarms caused by lack of confidence of the process operators in the measurement information provided by the instrumentation” [6]. Many people have recognized the problem, though their attention has mostly been directed toward fault detection in an overall unit process.

When lecturing on digital logic systems, I refer to the “microprocessor equation” which gives the unit-cost $U$ of some logic realization in terms of the technology $\alpha$ used as $U(\alpha) = C(\alpha) + D(\alpha)/n$, where $C$ is the component cost, $D$ the development cost, and $n$ the (projected) number of sales of the product. This equation implies the use of personal computers (large $C$, small $D$) for one-off developments, field-programmable gate arrays for medium-sized runs, and application-specific integrated circuits for very large $n$. Such an equation is relevant here. In aerospace $n$ is hopefully several hundred, but often in process control $n = 1$, as the stages of process design, construction, control hardware provision, control configuration, and tuning are performed by different companies, such that rarely are two processes identical.

Hence, what is necessary is to devise a mechanism for increasing $n$. This is done by noting that sensor and valve manufacturers provide standard ranges of products, so that for them $n$ is large. Indeed companies offer “smart” devices which include internal fault detection. But if we are to consider them as part of a system we need a generic method for reporting faults suitable for the many destinations to which sensor data, say, is sent: the control loop (how is it to respond to faults?); the maintenance team (preventative or just-in-time response to impending faults?); legal accounting records (has the device/plant been over-ranged?); process management (has the product been within quality bounds?). Hence our view is that the inbuilt intelligence of the sensor system should provide information useful for all these applications and not be just for fault detection.
Enabling Technologies

Several current technologies are making this to be an appropriate time to contemplate novel validation strategies and configurations:

- **Microelectronics.** This offers great economies of scale. Embedded microcomputers enable new algorithmic options in the sensor. Sigma-Delta analog-to-digital converters allow for bandwidth/accuracy tradeoffs in data acquisition. “Intelligent instruments” can have reconfigurable “personality data.”

- **Fieldbus.** This new international standard is replacing star-wired 4-20 mA current loops and proprietary digital highways as the means for connecting sensors and actuators to a network. Its digital bidirectional communications capability provides for the transmission of data sets which are richer than the standard single 4-20 mA analog signal and allows the central control system to interrogate devices for status and other information.

FDI: Fault Detection and Identification is an important topic in control systems theory. It employs many techniques such as physical redundancy, with multiple sensors as in the nuclear and aerospace industries, and analytical redundancy, in which a set of process models, parameterized as a function of expected faults, reconciles projected output behavior and the actual measured outcome. A further possibility is data redundancy. For example, the signal data required for control, say, is often of low bandwidth; higher frequency components in the raw data can often indicate fault modes. In condition monitoring, on-line fast Fourier transforms give spectral lines which depend on the presence of faults such as the failure of a pump’s impeller blade. Another approach is to mimic the plant operator, who can often detect anomalous patterns such as stick, pulse, jump, or cycling in the time-history of the sensed data [2]. With both these data-inspection ideas, the design choice reduces that of an appropriate filter which separates out signal and fault components. For example [14,15] model the sensed data as an autoregressive process, the residuals of which (being nominally “white”) can be used as a sharp and statistically treatable detection tool for a wide set of abnormal conditions. Hence, there are a range of methods that can be brought to bear on the validation process. Our thesis is that it is best left to the device manufacturer to apply these techniques, but that there should be standards by which validation outcomes are reported via Fieldbus to the “next level up”: a term which includes process controllers, alarm generators, and data-base computers.

Sensor Validation

A detailed discussion of sensor validation is in [5]. We start with the need for the sensor to generate a validity index and dismiss the idea of simply using a single good/bad “faulty bit” as inadequate. What is needed is some measure of data quality, and we turn to the science of metrology for help. Metrology, requiring traceability of a sensor back to international standards, plays a key but intermittent role in process management—most particularly in custody transfer when a product is being metered during its hand-over from a supplier to a customer. We argue that on-line metrology should be part of the modern validated sensor. This is becoming increasingly recognized in Statistical Quality Control, where “gauge capability” is seen to be a determining factor in effective process operation.

The key idea is the use of uncertainty as part of the validity index. Uncertainty [10] has entered international standards to quantify the associated error in a measurement. The value depends on both systematic (bias) errors and randomness: repeating experiments can reduce the second but not the first. Bias errors have to be treated nonstatistically via “judgment,” but are a function of the type of sensor and its calibration history, and indeed the duty cycle to which it has been subjected. Hence adequate long-term treatment of bias is precisely one of the features that need to be handled by an intelligent sensor. With a total uncertainty $\Delta M$, for a measurement $M$ it is asserted that the true value $M^*$ lies within a range: $M - \Delta M \leq M^* \leq M + \Delta M$, at a certain level of probability such as 0.95. In practice we use relative uncertainty $\Delta M/M$.

Though it has sometimes been argued that, when combining measurements from two sensors, bias and random uncertainty should be treated separately, it is conventional to lump them into a single number. Suppose that a derived measurement $y$ depends on two data sources $u, v$ via a nonlinear function $y = f(u,v)$. The chain rule gives, approximately:

$$\Delta y = \frac{\partial f}{\partial u} \Delta u + \frac{\partial f}{\partial v} \Delta v.$$  

Interpreting the $\Delta$’s as uncertainties, assuming errors in $u,v$ to be independent, squaring and rewriting in terms of relative uncertainties, we get:

$$\left( \frac{\Delta y}{y} \right)^2 = \left( \frac{\partial f}{\partial u} \right)^2 \left( \frac{\Delta u}{u} \right)^2 + \left( \frac{\partial f}{\partial v} \right)^2 \left( \frac{\Delta v}{v} \right)^2.$$  

As an example, consider computing the derived thermal energy flow $Q$ along a pipe from primary measurements of mass-flow rate $m$ and temperature $T$ using $Q = m\Delta T$. If these primary measurements have been validated, the transmitted data will include the computed uncertainties $\Delta m$ and $\Delta T$. Then the uncertainty in $Q$ can be calculated by the next level up as:

$$\left( \frac{\Delta Q}{Q} \right)^2 = \left( \frac{\Delta m}{m} \right)^2 + \left( \frac{\Delta T}{T} \right)^2.$$  

a result easily obtained by logarithmic differentiation. The important point here is that uncertainty is a generic property which can be manipulated by the next level up for the derived measurements using the uncertainty calculus as exemplified above.

Many sensors combine measurements internally, in particular for temperature compensation. As an example, suppose a mass-flow meter uses an algorithm of the form $m^* = m + \alpha T$, where $m^*$ is the “raw” calculation and $\alpha T$ the temperature compensation term. Then:

$$\left( \frac{\Delta m^*}{m^*} \right)^2 = \left( \frac{\Delta m}{m} \right)^2 + \left( \frac{\Delta \alpha T}{\Delta T} \right)^2.$$  

IEEE Control Systems
As the correction factor $aT$ is small, the squaring of the terms makes the contributing effect of relative temperature errors on overall uncertainty smaller still.

This is where the fun begins. Suppose that a fault detector indicates that the value of $T$ is wrong. Should we simply flag that $m^*$ is wrong? Should we persist in using the wrong value of $T$, or bypass the compensation (i.e., assuming $T = 0$), or use some fudged value of $T$? We assert instead that the duty of a sensor is to provide the best estimate of the true process variable even in the presence of faults. Consider $T$. The sensor will, from historical data, have stored temperature limits encountered during the lifetime of its operation. Hence the true value is presumably somewhere between the limits, which therefore give a long-term uncertainty range. Simply taking the mean temperature (or adopting some risk-averse strategy) gives the nominal $T$ to use in the compensation. However, before the fault was flagged it was likely that $T$ and $DT$ were correct and as the maximum rate-of-change of $T$ is known, one possible algorithm is to move the nominal $T$ smoothly from the last "good" value to the long-term value, simultaneously increasing $DT$ at some appropriate rate.

The computed uncertainty given the detected fault conditions might be enough for some applications, and the diagnosed fault-flag is likely to alert the maintenance team. However, is the (validated and corrected) measurement still fit for duty? The appropriateness of the data must depend on the application, which is known by the next level up but not by the sensor. Some indicator of validity must be communicated upwards; this we call the measurement status, using a visual analogy:

- **clear**: the raw data is fine
- **dazzled**: a possibly transient abnormality (e.g., a spike or outlier is possible in the raw data but corrected for and eliminated by the validation algorithm)
- **blurred**: abnormal (e.g., noisy) raw data but believed to have some correspondence to the real measurand
- **blind**: the raw data is completely untrustworthy, such as with a confirmed and persistent dazzled condition.

Note that for all these states the sensor is still giving its best guess of the true measurand and the computed uncertainty band remains credible.

The measurement status can, in principle, be used to switch modes of the loop controller. The actual logic will depend on balancing safety and availability in the plant. In some cases only **clear** will do, but frequently a loop can be sustained under all partly-sighted conditions. If the measurement corresponds to a constraint variable as used by a predictive controller such as Quadratic Dynamic Matrix Control, the actual constraint target can be lowered to reflect the current uncertainty band. If, as is often the case, there are more sensed variables than controlled variables, it may be possible to reconfigure the controller and used only clear data.

Hence, Sensor Validation which leads to the structure of a so-called "SEVA sensor", can be represented by the intersection of the circles in Fig. 2, and is seen to be a combination of technologies.

**A Thermocouple Example**

Thermocouples, as indicated above, play a key role in process control. See [12,13] for a detailed treatment of a validation system for thermocouples, including a full uncertainty analysis.
Common faults include open-circuit cables (easily detected) and loss of thermal contact between the thermocouple and the process. This second fault is detected by a device-specific test. A current is injected into the thermocouple, inducing ohmic heating, after which the transient of sensed temperature is inspected. The idea is that the loss of heat transfer is two-way, so that "good" thermal contact will direct away the injected heat and the temperature rise during the test is small. However, poor contact results in a local increase in temperature, which takes a correspondingly long time to decay back to the surrounding value, as depicted by Fig. 3.

With an open-circuit fault, the output voltage of the input-stage operational amplifier drifts to its maximum, giving rise to a large (negative) unvalidated temperature reading. The validation process detects the fault, indicates blind, and increases the uncertainty band as shown in Fig. 4, where data from a practical trial [13] is presented. The upper plot compares the unvalidated and validated transmitted values, whereas the bottom plot indicates the time-history of the validated value, the uncertainty band, and the generated status values. It is seen that when the fault clears, the status is set to blurred and the sensor enters a recovery phase prior to resuming normal behavior. The computed values of the uncertainty are based on a detailed analysis of the sensor and the analog signal processing circuit [13].

**Actuator Validation**

Through an actuator, control is brought to bear on a process. Unfortunately, the actuator model assumed in the vast majority of control theory papers—that it is linear, instantaneous in operation, and has an unbounded output—is wrong. In practice, actuators are nonlinear, slow with direction-dependent dynamics, and limited in their action. Moreover, they deteriorate with time, thus degrading the performance of the loop. This has been recognized in a series of papers by Isermann (e.g., [8]) where actuation for mainly electromechanical systems is considered. We attempt here to model a generic actuator in the spirit of the rest of the article, but concentrate on flow valves as these are by far the most important in process control.

It is tempting to invoke duality with sensing with a statement such as: "a sensor passively captures a process variable, whereas an actuator actively affects its value." However, first consider what we call a Level 0 actuator, in which the demanded control signal \( u_d \) directly changes some internal physical property, say the valve stem position. We assume that the corresponding actual value (of position say) \( u_a \) is monitored, and hence amenable to sensor validation techniques if necessary. We define an ideal actuator to have the input/output relation shown in the left-hand diagram of Fig. 5, in which the output \( u_a \) is linearly related to the input \( u_d \) between saturation limits \( u_{min} \) and \( u_{max} \). It is convenient to scale variables so that \( u_{min} = 0 \) and \( u_{max} = 100\% \) (e.g., a valve output varies between fully closed and fully open) and such that the slope of the relation is unity.

A practical control algorithm must know the values of \( u_{min} \) and \( u_{max} \). For example, a predictive controller [4] which ignores saturation is liable to induce significant overshoots for large set-point changes, which are markedly reduced if (via Quadratic Programming) the optimization over the prediction horizon takes saturation into account. This problem is well-known in PID regulation, where a major requirement is to ensure anti-windup, such as by clamping the integral component when the actuator hits a limit. It is convenient for us to implement PID in feedback form, as shown in Fig. 6, as will become clear later.

In the diagram \( f(.) \) is the nonlinearity and \( G(s) \) represents actuator dynamics. The signal \( u_d \) is the demanded control and \( u_a \) the actual value, as measured, say, by a valve positioner. Note the positive feedback in the loop. Suppose the actuator is "perfect" so \( u_a = u_d \). Then:

\[
(1 + sT) u_d = K(1 + sT)(w - (1 + sT)y) + u_d,
\]

giving:

\[
(1 + sT) u_d = K(1 + sT)(w - (1 + sT)y) + u_d.
\]
a PID regulator in interactive form and avoiding "derivative kick." Suppose \( f = 1 \) and now \( G(s) \) has unity-gain dynamics, then:

\[
u_d = K \frac{1 + sT_i}{sT_i} \left[ w - (1 + sT_d)y \right].
\]

This has integrating action (put \( s = j\omega \) and \( \omega \to 0 \), For example, if \( G(s) = 1/(1 + sT) \), we obtain:

\[
u_d = K \frac{1 + sT_i}{s(T + T_i)} \frac{1 + sT}{1 + sT} \left[ w - (1 + sT_d)y \right].
\]

where \( T_i = T(1 + T/T_i) = \tau \) as \( T \ll T_i \) in general, so that the actuator dynamics simply reduces the effective controller gain \( K \) by a small amount to \( KT/(T + T_i) \). This is easily compensated for, so that \( G \) does not then affect the loop dynamics — one advantage of the configuration.

Suppose now that the actuation signal saturates. Clearly the feedback signal \( z \) always lies within the saturation region, and approaches the saturated value with the time-constant \( T_i \). When, after some time, the error signal \( [w - (1 + sT_d)y] \) changes sign, it is clear that \( u_d \) (and hence \( u_d \)) immediately drops from the saturated value, as required. Note in particular that the saturated value fed back via \( z \) corresponds to the actual physical value of the actuator, not some assumed value as in most desaturation strategies. Hence if for some reason the saturation level changes, the feedback PI setup will modify its behavior automatically.

One nice feature about this form of PID implementation is that it generalizes to the cascade case, as with the example of Fig. 7. The control signal of the temperature loop is \( F_{sp} \), the setpoint of the inner flow loop. Cascading has several advantages: it permits easier design from simple models/intuition, and better control commissioning (tuning the inner loop first); it is robust to outer loop failure; and the inner loop rejects pressure disturbances and linearizes the valve/flow characteristic. The disadvantages of using cascade control are the costs of the extra sensor/transmitter/controller and the need to insert an orifice plate (typically 30 pipe diameters from valve).

Consider desaturating the outer loop of a cascade structure, such at TT/TC of Fig. 7. In Fig. 8 signals \( w_0, y_0 \) are the "Level 0" actuator demand \( u_d \) and response \( y_d \). Signal \( y_1 \) is the manipulated variable of the inner loop (flow). Signal \( w_1 \), from the outer-loop PID, is the inner-loop setpoint, and \( y_1 \) is the returned achieved value. The point is the consistency in the signal structure all the way up from Level 0. A requirement for PID/desaturation in the feedback implementation is that there is unity DC gain between \( w \) and the returned \( y \); this is achieved by the inner loop PI regulator. Hence the structure is generic and allows for arbitrary levels of cascading, and so we suggest the definitions:

- **Level 0** smart actuator returns a (validated) signal corresponding to the real actuation signal (e.g., the stem position).
- **Level 1** smart actuator returns a (validated) signal corresponding to a sensed process variable (e.g., the measured flow) and so on.

### Figure 7
Controlling temperature via a cascaded loop.

An ideal controller will make \( y \) the same as \( w \) at all times. Saturation is one mechanism which prevents this; the returned \( y \) indicates this event and allows the controller to take appropriate action.

We now turn to the behavior of a non-ideal actuator, being nonlinear as shown in the right-hand plot of Fig. 5. We assume that the physical variable \( u_d \) is measured and as we have control over \( u_d \), it is possible in principle to maintain an accurate image of the nonlinear static characteristic \( f(u) \) within our compensation algorithm. The key point is that it is then possible to invert the nonlinearity so that via a table look-up any required \( u_d \) is selectable by appropriate \( u_d \). This is not possible in two principal cases: saturation (\( u_d \) exceeding \( u_{max} \) is not allowed) and where there is a jump in the input/output function such as \( A \ldots B \) in Fig. 5. Hence it is possible by using an inverse nonlinearity to provide an actuator which has the ideal saturation characteristic (as handled by our PID feedback form) but with a possible jump in the function. This jump is an essential nonideality and provokes the following "theorem":

If the setpoint or load disturbance is such that the required actuation signal lies in the unreachable range \( A \ldots B \), there will be a limit cycle. **Proof:** reductio ad absurdum.

Furthermore, the limit cycle frequency will be approximately at the bandwidth of the closed loop. Note that in effect the actuation is providing a mark-space ratio which on average will
produce the unrealizable actuation signal. We can respond to the problem of limit cycling in several ways. It might be preferable to use an internal actuator algorithm which provides the correct mark/space ratio at its own bandwidth, presumably wider than that of the main outer loop, such that the plant’s filtering action reduces the consequent ripple on the plant output. Alternatively, we could decide not to allow limit cycling and hence impose an appropriately scaled dead zone in the forward path, thus trading the inevitable offset against cycling behavior.

Hence by using internal feedback or by applying an inverse nonlinearity we attempt to make the nominal actuator approach the ideal. In Fig. 9, $f^{-1}$ represents the “best achievable” nonlinear inverse.

![Compensation for actuator nonlinearities](image)

**Fig. 9. Compensation for actuator nonlinearities.**

A validated actuator uses all means at its disposal to evaluate and compensate for internal nonidealities. Backlash, friction, deadzone type 1 (a) can be inverted if the corresponding limits are known. Some nonlinearities, such as saturation, deadzone type 2 (b), however, cannot be inverted. The “inventibility discrepancy” should be measured and reported to the next level up so that appropriate action can be taken.

Let us return to the Level 1 actuator (which is the dual of the validated sensor). Internally it contains a measurement of the actual process variable it is attempting to influence, e.g., a flow rate. Traditionally flow control involves three units: differential pressure (DP) sensing, loop PI controller, and actuating valve. But in practice do we need to follow the usual “30 diameters away from the valve” rule for installing the orifice plates which provide the differential pressure? Because flow is nearly always in cascade with other loops, its purpose is surely to maintain consistency, to linearize, and to reduce disturbance effects such as changes in feed pressure. This can just as easily be achieved by making the valve its own flow sensor using inserted pressure tappings and then to include all of the Level 1 loop locally within this “smart” valve, clearly saving on installation and other costs. This then gives us two new mechanisms for determining flow: a forward path using a calibrated stem position and a feedback path using the measured pressure drop across the valve. Having two independent sources gives a much wider scope for cross-validation.

**Loop Validation**

Given validated sensors and actuators it is natural to move toward loop validation, as in Fig. 10, where we can use all the tools available from FDI, signal processing, and parameter estimation. Unfortunately it is easy to lose economies of scale, as all loops are different and the validator needs to be “tuned” to the loop (though if the PID loop is well-tuned we can concentrate on frequencies near $1/T_1$, and the value of $T_1$ is a good guide of time scales required for data averaging, filtering, and so on).

A useful review of loop performance indicators is in [1], where it is suggested that simple step tests can be used to deduce three parameters: process gain $k_p$, apparent dead time $L$, and apparent time constant $T$. From these derived values the dimensionless parameter $\theta = L/T$ can be obtained. A simple relay test, as is often used in auto-tuning of PI regulators, obtains $k_{rel}$, the plant gain corresponding to a phase lag of $180^\circ$. A further dimensionless parameter $\kappa_p = k_p/(L/T)$ can be computed; $\theta$ and $\kappa_p$ are then used to assess the achievable performance for processes with monotone step and frequency responses. For example, [1] indicates the appropriate ranges of application of various controller structures as a function of the parameter $\kappa_p$.

Loop validation *inter alia* monitors the achieved performance of a loop, either in terms of statistics such as data means and variances or by deducing the variations in the parameters $\theta$, $k_p$. Many useful statistics would be collected automatically by validated sensors and actuators. For example, measurement noise and actuator saturation limit the achievable bandwidth and hence the loop gain: the uncertainty band computed by a validated sensor and presence of actuator clipping could both be used to indicate that the controller gain should be reduced. The algorithm for deducing the actuator’s static gain characteristic $f$ could also compute the plant’s static gain function; if this turns out to be nonlinear, the controller can be compensated so as to preserve a constant loop gain. The actuator could, on demand from the next level up, inject a relay-like signal at or near the $k_{rel}$ frequency to determine whether the ultimate period or gain had changed. Given that the input/output devices have been validated, these simple tests are used to ensure that the rest of the loop is operating under nominal conditions.

The simple indicators described above can readily be added to a Level 1 actuator and are economically viable for single loops. Multivariable processes, however, pose a problem of a different order of complexity. For example, it is doubtful whether model-based validation is cost-effective in process control applications.
because of the expense and expertise required for constructing and validating the model, and because of the management problem of ensuring that such a model is kept alive in the event of plant modifications. One interesting possibility is to use multivariable statistical methods such as projection onto latent structures [11]. Such approaches use the database constructed from normal operating records to find, within the data, those variables which provide the main contributions to plant deviations. This "influential subset" of signals can then be monitored on-line in order to provide fault diagnostics. Though powerful, the creation and maintenance of data-driven diagnostics depend on the presence of skilled personnel.

While full plant validation needs all the range of FDI tools, much can be achieved by concentrating on two extremes: excessive loop activity such as oscillations, and poor loop response caused by blockages, fouling, and so on. In practice, oscillations are more likely to be caused by poor actuators (hopefully removed by validation) or disturbances, rather than by over-tight control. Nevertheless, several researchers have produced mechanisms for detecting loop oscillation. Poor response is perhaps best detected from signature analysis: perturb the valve stem over some allowable range at a rate dependent on the bandwidth of the process and inspect the plant's response compared with some nominal value: see [9] for associated filtering operations. This test has the added advantage of providing rich data to assist in validating the actuator and sensor. But there is a lot of work to be done to produce generic, useful, and economical methods in this area.

Conclusions

It is argued that ensuring good operation of sensors and actuators is essential for process control, and that economic pressures are devolving responsibility for validation down to the individual functional unit, with rich validation data being transmitted via the interconnecting Fieldbus. At present several sensor manufacturers provide some crude form of validation (essentially simple fault detection) using microcomputers in their "smart" instruments; validation in actuators and the loop level is still, however, in its infancy. It is to be hoped that, in the future, standards for validation throughout the loop will appear, so that this important feature can be economically embedded in plant control hardware. An effective standard should improve the quality of loop control and increase the availability of process plants.

Acknowledgements

I would like to thank Manus Henry, Janice Yang, and Peter Alsop for their contributions to sensor and valve validation. Oxford's sensor validation research is funded by Foxboro and valve/loop validation by the UK EPSRC.

References


David Clarke is head of the Department of Engineering Science, Oxford University, and a Fellow of New College. With coworkers, he developed the Generalized Minimum-Variance and the Generalized Predictive Control algorithms. His current research on adaptive control is concerned with stable self-tuners based on predictive methods subject to state constraints. Industrial collaborative work is on sensor, actuator, and loop validation. He was a founding editor of IEE Proceedings, Part D, and is an associate editor of Automatica, an editor for the Int. J. of Adaptive Control and Signal Processing, and a founding editor of the new European Journal of Control. In 1989 he was elected to the Royal Academy of Engineering.

August 1995 45