



HVAC&R Research

Publication details, including instructions for authors and subscription information:
<http://www.tandfonline.com/loi/uhvc20>

Automatically detecting faulty regulation in HVAC controls

Daniel A. Veronica^a

^a National Institute of Standards and Technology, 100 Bureau Drive, Mailstop 8631,
Gaithersburg, MD, 20899-8631, USA

Accepted author version posted online: 18 Apr 2013. Published online: 29 May 2013.

To cite this article: Daniel A. Veronica (2013) Automatically detecting faulty regulation in HVAC controls, HVAC&R Research, 19:4, 412-422

To link to this article: <http://dx.doi.org/10.1080/10789669.2013.789369>

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at <http://www.tandfonline.com/page/terms-and-conditions>

Automatically detecting faulty regulation in HVAC controls

DANIEL A. VERONICA*

National Institute of Standards and Technology, 100 Bureau Drive, Mailstop 8631, Gaithersburg, MD 20899-8631, USA

A new method is introduced to automatically detect faulty regulation of temperatures, pressures, and flow rates within HVAC systems and equipment of commercial buildings by using digital data typically available from an existing building automation system. The building automation system passes data by network to a general-purpose microprocessor executing this method. The method computerizes control charts and combines them with expert system logic to identify transients and record excursions of regulated variables beyond allowance bands set by the user. Its three separate functions monitor (1) variables regulated to a single set-point value, (2) actuating variables that drive the regulation, and (3), temperatures regulated to duplex set-points (i.e., thermostats). Faults detected include unstable, excessively oscillatory regulation and failure of regulated variable to maintain the allowed band. A brief background on control charts and expert systems is in Appendix A.

Introduction

The difficulty that any building staff of economically practical size and expertise faces in properly monitoring and maintaining the complex HVAC systems of modern commercial buildings is attested to by sources such as Westphalen et al. (2003). Digital data sampled from sensors throughout the HVAC system offer a primary resource for addressing the problem. These sensors typically—though not always comprehensively—are provided by an existing building automation system (BAS) controlling the HVAC system. Software products utilizing such data are marketed to aid staff in fault detection and diagnosis (FDD) of the HVAC system. But given the report by Summers and Hilger (2012), it is evident that current FDD software can demand tedious and costly amounts of expert time and effort from the staff or a consultant. A remedy is to develop novel *automated* FDD (AFDD) software “tools” that help the staff ensure everything works well without placing uneconomical demands on human experts.

An essential task for an AFDD tool is to determine whether closed-loop automatic controls, such as proportional-integral-derivative (PID) compensators, are properly regulating temperatures, pressures, and flow rates. That requires the tool to have an analytical component able to identify autonomously (on its own) unstable, excessively oscillatory behavior (“hunting”) in the regulated quantity as well as in the regulating device (e.g., damper or valve).

Referring at first only to the top plot panel of Figure 1, the air temperature T_{az} of a building zone served by a fan-coil unit is shown over about 39 h of a heating season as an irregular black line. The abscissa common to all the panels is time, labeled at the figure bottom as hours after midnight. Straight, horizontal upper and lower gray lines show, respectively, the cooling and heating set-point (i.e., intended) temperatures for the zone. It is seen that the unit generally regulates zone temperature acceptably, within 0.5°C (0.9°F) of the heating set-point value. However, there is subtle evidence that something is amiss, because the variation of zone temperature between Hours 03 and 18 at the left end of the plot is suspiciously large and rapid compared to that before Hour 03, while between Hours 18 and 04 on the following day, the variation is much slower, sluggishly persisting below the heating set-point for longer periods. From Hour 04, the zone temperature is again regulated well as at the beginning of the plot, until a hidden event near Hour 13 causes regulation to jump to the cooling set-point.

So, the top panel of Figure 1 shows temperature regulation of a zone affected, at times adversely, by factors not identifiable in the plotted data alone. It is possible these hidden factors degrade the energy efficiency of the HVAC system or put unnecessary wear on components. Further information is needed to determine if that is true. However, the only information building staff would have are complaints from occupants during the sluggish excursions of zone temperature below set-point. In buildings having centralized plants serving tens to hundreds of distributed zones, the staff typically would not have time to create plots such as Figure 1 to examine for suspicious features. Even less plausible would be the further time and effort the staff needs to gather evidence about observed features and come to diagnoses resolving whether or not faults exist and, if they do, where they are and what is the cost of their impacts. It is instead the job of an AFDD tool to resolve all those things for the staff as automatically and autonomously as practicable.

This article not subject to US copyright law.

Received October 12, 2012; accepted March 9, 2013

Daniel A. Veronica, PhD, PE, ASHRAE Member, is Mechanical Engineer.

*Corresponding author e-mail: daniel.veronica@nist.gov

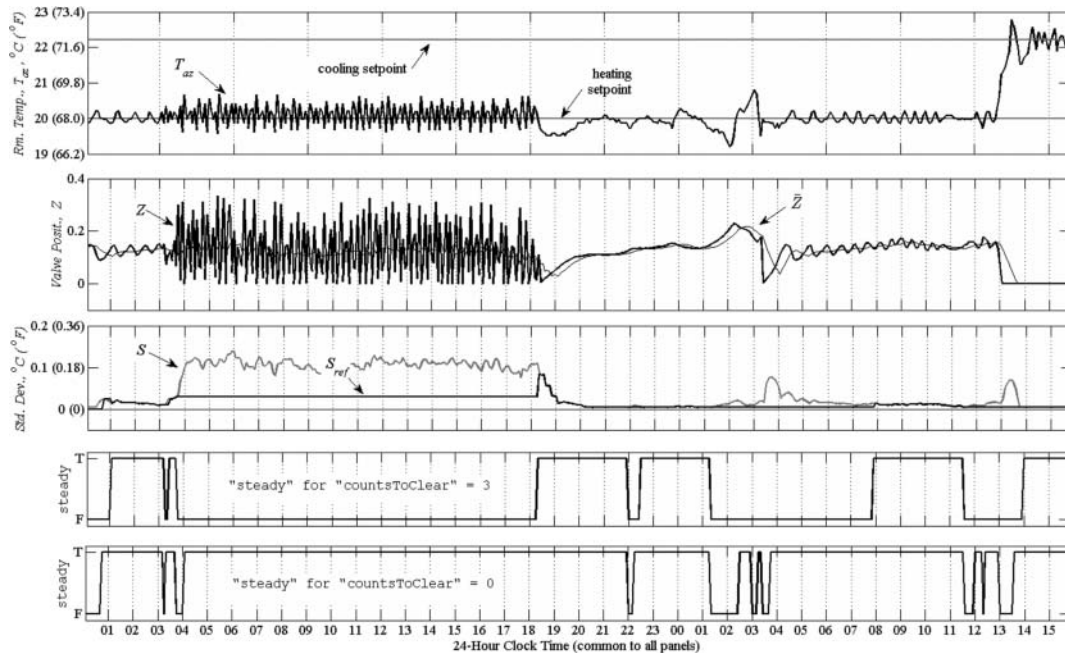


Fig. 1. Logic analysis sequence of erratic valve activity.

Primary capabilities for detecting and diagnosing HVAC control faults

To fully diagnose situations as those just discussed, the AFDD tool must perform some functions beyond the scope of this article. All those functions rely, however, upon the tool first having at least four primary capabilities.

- Capability (a): It should reliably discriminate dynamics caused by faults from dynamics reflecting normal, fault-free operation.
- Capability (b): It should identify excessive variation or drift in a regulated quantity, that is, a dependent quantity, such as duct air temperature or pressure, having a single set-point value.
- Capability (c): Some quantities such as zone temperature at a thermostat are regulated by duplex set-points (e.g., a higher value activating cooling and a lower one activating heating with an unregulated band between). Hunting and drift should be identified in this case without implicating free dynamics in the unregulated band.
- Capability (d): Faulty dynamics, such as hunting or drift of a regulated variable, should be associated to the devices exerting the regulation, such as specific valves, dampers, fan motors, or control modules. Correlations can then infer whether the problem is really a more primary fault in one of those devices.

These four primary capabilities can be accomplished by combining control charts with expert system logic.

Background and prior research

Appendix A provides a brief background on two types of control charts, the Shewhart chart and cumulative sum (CUSUM)

chart, and discusses issues facing their use in AFDD of HVAC systems. Appendix A also discusses prior research dealing with those issues and defines an “expert system” as previously referenced.

Approaches much more elaborate than control charts have been considered for AFDD of controllers. For example, Fasolo and Seborg (1995) evaluated controller performance using an index based upon regressions into a time-series model, Miao and Seborg (1999) used an index constructed from autocorrelation functions, and Tudoroiu et al. (2009) examined techniques from state estimation and spectral analysis. These more sophisticated approaches can in theory claim some esoteric diagnostics, such as harmonic decomposition, that control charts alone cannot. It is shown here, however, that satisfactory AFDD of closed-loop regulated HVAC devices can be achieved more simply by coupling control charts to expert system logic.

At least one prior HVAC research effort with control charts (Schein and House 2003) has been implemented by industry. There, CUSUM charting is embedded within the controllers of variable air volume (VAV) units by appending logic to the units’ usual control programs, demonstrating online (i.e., real-time) detection of faults in the units. To keep that added logic, called the VAV performance assessment control chart (VPACC), small and simple enough for the 1990s-era microcontrollers and interface software it targeted, it does not estimate the two statistics CUSUM charts typically need: data mean and standard deviation. VPACC instead replaces those statistics with exogenous a priori parameters obtained offline through controlled testing of VAV units.

During operating transients in HVAC systems, autocorrelation appears between samples taken of a single variable over time. Because it can be mistaken as fault evidence, autocorrelation was viewed by prior research as an obstacle to using control charts for HVAC, as detailed in Appendix A. To

reduce false alarms from autocorrelation, VPACC suspends its surveillance for fixed periods after shifts in zone schedule from “unoccupied” to “occupied” and clears its registers every 6 h. Appendix A also describes how autocorrelation often cross-correlates to well-known factors reflecting normal, acceptable system operation. Because it is not an expert system, VPACC cannot isolate normal transient behavior from faulty behavior, nor can it attribute faulty behavior to an external cause (i.e., attribute a VAV unit fault to trouble in the air handler upstream). Referring to the preceding capabilities list, VPACC performs Capability (b), but not Capabilities (a), (c), or (d). Schein and House (2003) thus acknowledged that VPACC generates more false or misleading alarms given the realistically cross-correlated data of field sites than when given relatively simplistic data from computer simulations or laboratory tests. What is needed is presented next: a way for an AFDD tool to expertly and autonomously distinguish changes in a quantity due to faults from changes due to normal dynamic operation—Capability (a).

Distinguishing controller faults from normal dynamics

A “normal” transient in a variable is one caused by dynamics in an accepted influence, such as a valid control signal, an occupancy or plug load change, weather, or hour of day. Of those four, the first presents a greater challenge because it has the potential to drive the most rapid transients, and a primary concern is a normal transient occurring fast enough to fill control chart registers to an alarm limit when in fact no alarm is warranted. Slower normal transients are much less a problem, because the charting algorithm can be designed to maintain its own autoregressive mean of the variable as a moving datum to key the chart registers. This mean will follow the dynamics of slow-acting influences and thus help avert false alarms. While occupancy or plug load changes could also conceivably produce fast transients, most BASs do not evaluate (at least not numerically) those factors, and so data on them is generally sparse or absent.

An expert system infrastructure supports the task

In many HVAC processes, such as heating and cooling coils, the controlled quantities (e.g., leaving air temperature [LAT]) normally follow regulation (e.g., valve motion) with significant and varying first-order lags. Such lags make it more difficult to use numerical techniques such as correlation, regression, or hypothesis testing, to infer causes to the effects observed in data. Instead of a numerical technique, correlating logically between binary (i.e., true or false) “states” is much simpler, yet fully sufficient to provide the four capabilities listed earlier. This is illustrated by the following simplified case.

A control chart monitors each instrumented quantity—such as controller set-point, airflow rate, and chilled water (CHW) valve position—normally driving the values of each regulated quantity, such as the LAT from a cooling coil. A binary variable, one of the many logical states in this expert system, is output by a Shewhart chart function, $\text{ShewChart}(\dots)$,

operating on the current CHW valve position, Z_{vc} :

$$Z_{vc}\text{Steady} = \text{ShewChart}(Z_{vc}, Z_{vc}\text{Mean}, Z_{vc}\text{StdDev}), \quad (1)$$

where $Z_{vc}\text{Mean}$ and $Z_{vc}\text{StdDev}$ are the running mean and standard deviation of the valve position, calculated for the chart from a circular data buffer. After all control charts have operated upon the current round of sampled data, their output states can be used in logical expressions downstream either alone (as “primitives”) or combined to form derived states, such as

$$\text{LATfactorsSteady} = \text{setptSteady} \& \text{airflowSteady} \& Z_{vc}\text{Steady}. \quad (2)$$

The symbol $\&$ is the logical “and” operator, and LATfactorsSteady is a binary state characterizing the steadiness of the factors that an expert knows normally affect a coil’s LAT, being in this example, respectively, its set-point, airflow, and valve position. Downstream of derived states, the AFDD tool program steps through arrays of paired if–then “rule” statements testing that the states correlate in ways reflecting normal HVAC system operation. For example, given a control chart has generated the primitive logical state LATsteady , two rule pairs (numbered “13” and “14” in this example) use it with a derived state in

$$\text{ruleIf}(13) = \text{LATfactorsSteady}, \quad (3)$$

$$\text{ruleThen}(13) = \text{LATsteady}, \quad (4)$$

$$\text{ruleIf}(14) = \text{not}(\text{LATsteady}), \quad (5)$$

$$\text{ruleThen}(14) = \text{not}(\text{LATfactorsSteady}), \quad (6)$$

where the $\text{not}(\dots)$ function negates (i.e., flips to opposite) the state in its argument. The $\text{ruleIf}(n)$ and $\text{ruleThen}(n)$ terms are not programmer’s if–then syntax, but they are another form of binary variables derived from previously evaluated states. The key distinction is that while states alone do not necessarily have “normal” or “faulty” values, rule statements explicitly use states so that a “false” value for any $\text{ruleThen}(n)$ when its $\text{ruleIf}(n)$ is “true” is evidence of a fault in the HVAC system. In the example, it is normally expected that any “true” value for LATfactorsSteady correlates to a “true” for LATsteady . Conversely, a “false” for LATsteady is normally expected only when LATfactorsSteady is “false.” A fault in the LAT controller is suspected when either is not the case, as tested by

$$\text{ruleTest}(n) = (\text{ruleIf}(n) == \text{false}) | ((\text{ruleIf}(n) == \text{true}) \& (\text{ruleThen}(n) == \text{true})), \quad (7)$$

where the operator “==” tests for logical equality, and “|” (i.e., the “verbar” character) is the logical “or” operator. Each rule index n , where ruleTest is “false,” causes the AFDD tool to initialize a fault detection “case” for interactive diagnosis later using tool components beyond the scope of this article. Note that the ostensibly simpler test of merely checking for logical equality between LATfactorsSteady and LATsteady would not express an equally full knowledge about feedback

control. So, authoring effective states and rules for an expert system is an expertise in itself, a topic also beyond the scope here.

The essential points for now are that control charts and expert system logic are two complementary parts in an infrastructure of computer programming engineered to produce autonomous capabilities in the AFDD tool, and that those capabilities begin with the control charts generating primitive states sequentially from the data they are fed. The remainder of this article focuses on that last point.

Identifying transients in variables automatically

Some prior research on FDD of HVAC equipment, such as Li and Braun (2003), employs a priori models of acceptable steady-state process behavior to detect faults. These models can be computational (e.g., first-principles equations), statistical distributions, or tabulated performance data. The AFDD tool then needs a “steady-state detector,” such as that described by Kim et al. (2008), to distinguish the periods of data compatible with such models from periods it must ignore. If the tool does not use a steady-state model, there is no reason to bar valid dynamic data from it, so filtering data down to overall steady states alone is potentially unnecessary. What expert logic does need instead is automatic detection of when specified quantities begin and end transients. This feeds states and rules logically correlating, for example, that an excursion in duct air temperature is due not to some fault, but to a normal transient in CHW valve position instigated by a scheduled reset of CHW temperature. Transients are identified automatically by adapting the Shewhart chart as follows.

Automating the Shewhart chart involves calculating the z -score, z_n , of each sample x_n from the sample set mean $\bar{x}$ and standard deviation s . Equation A3 of Appendix A shows a z -score in general terms, but to accomplish the purpose here, two customizations are necessary. Given that the HVAC process has continuing duration, $\bar{x}$ and s must be computed from a circular “rainfall” buffer of running sample values extending back a defined time span. Forty minutes (eight samples) produced the results seen here. Also, unlike Equation A3, the standard deviation used to calculate the z -score at sample time n , z_n ,

$$z_n = \frac{(x_n - \bar{x})}{s_{ref,n}}, \quad (8)$$

is not always from the current contents of the rainfall. Instead, the standard deviation used is that from the instance of the rainfall when the quantity was last automatically classified as being steady: a “running reference” standard deviation s_{ref} . It can differ from the current value s , as expressed for sample n by the following pseudocode:

$$s_{ref,n} = (s_n * \text{isTrue?}(\text{steady})) + (s_{ref,n-1} * \text{isFalse?}(\text{steady})). \quad (9)$$

The casting functions $\text{isTrue?}(\text{steady})$ and $\text{isFalse?}(\text{steady})$ both test whether a binary logical variable gener-

ically named steady (such as LATsteady in Equation 4) is, respectively, “true” or “false.” The casting functions return the number “1” if their test (not the variable) proves true and “0” (zero) if it does not. As shown, the numbers returned are multiplied (the “*” operator) by, and thus select between, the standard deviation of all samples currently in the rainfall buffer s_n and the reference standard deviation brought forward from the previous iteration, $s_{ref,n-1}$.

Equations 8 and 9 are computed for each sample x_n of each quantity x (e.g., a temperature, flow rate, or actuator position) monitored by the Shewhart chart for transients. The Shewhart chart in automated form is a function $\text{ShewChart}(\dots)$, which assigns a “true” value to a normally “false” binary logical state shewTrip (meaning a sample has “tripped” a chart threshold) for any z_n outside the three-sigma band defined by $\pm 3s_{ref}$. The chart thus takes that sample as evidence that x is now, as explained in Appendix A, affected by a special-cause factor.

No more is asked each sample period of a Shewhart chart than to revalue its shewTrip output state for the variable it monitors. It is up to expert logic downstream of all the charts to determine what any “true” value for a shewTrip means in the context of the prevailing HVAC system operation. For example, correlating a special-cause factor found by a chart to a valid change in a controller set-point infers a normal event. The method is to first pass each shewTrip to a state characterizing whether the monitored variable is “steady” (i.e., within the band allowed as steady-state variation):

$$\text{steady} = (\text{not}(\text{shewTrip}) \& \text{isEqual?}(\text{counter}, \text{countsToClear})), \quad (10)$$

where counter and countsToClear are both positive-valued integers. The functions $\text{not}(\dots)$ and $\text{isEqual?}(\dots)$ apply, respectively, a negation and a numerical equality test. Initially, shewTrip , steady , and counter are, respectively, “true”, “false”, and 0. Each sample period that shewTrip is “false” increments counter , up to a maximum called countsToClear . Each period that shewTrip revalues as “true” decrements counter , down to a minimum of 0. Either update occurs after Equation 10 executes. Changing steady from “true” to “false” is thus immediate, but going from “false” to “true” is delayed by countsToClear . This reduces chattering and spurious true values of steady , as seen by comparing the plot panels of Figure 1, collectively called a *logic analysis sequence* because it shows changes in specified states of the expert logic over time alongside the analog quantities driving those changes.

The bottom two panels Figure 1 show steady , which refers to the behavior of the heating coil water valve position Z , seen in the second panel from the top. The only two values of steady possible are “true” (T) or “false” (F) on the vertical axis. In the bottom-most panel, countsToClear had been set to 0 (i.e., no delay action), resulting in steady being incorrectly true during an interval of wide, obvious hunting of the valve. In the panel second from bottom, where countsToClear is set to 3, the same hunting interval is correctly identified by steady being “false”. Only when steady is “true” is s_{ref} updated to equal s .

The benefit of combining control charts with expert logic is seen by comparing the top four plot panels of Figure 1. The second panel from the top shows hunting of the hot water valve position Z as the obvious immediate cause of the zone temperature oscillation seen in the top panel and discussed in the Introduction. The expert logic infers the same automatically without human graphical cognition. A crucial factor is the proper update of the reference standard deviation s_{ref} by the standard deviation s from the current 40-min rainfall. To prevent divide-by-zero errors, rainfalls are binned so that 0 values of s do not occur even for constant data.

It is seen that erratic valve activity just before Hour 04 trips the Shewhart chart, assigning “false” to steady, which in turn saves away the current standard deviation s as s_{ref} . The valve action is not again considered steady until after Hour 18, when it calms back to a variance that repeatedly (recall the role of countsToClear) yields “false” for shewTrip. At that point, the current value s of standard deviation becomes the “new” s_{ref} used by the Shewhart chart (via Equation 9). This new s_{ref} begins larger than its previous value, the valve activity held in the rainfall buffer only just satisfying the chart, but s_{ref} settles adaptively to lower values thereafter.

Values of countsToClear greater than 0 put the switch of steady from “false” to “true” later than a human expert reading the valve plot would more accurately put it. But it is evident that 0 for countsToClear yields misidentifications that a non-zero value prevents. In the bottom panel of Figure 1, much of the Day 1 period of erratic action by the valve between Hours 04 and 18 is misidentified as being steady, because a single sample met the three-sigma chart criterion (with s_{ref} as sigma), switching steady to “true”, which jumped s_{ref} to a very large value. That subsequently allowed a factually erratic period to be marked as steady, an error the same routine avoids in the above panel where countsToClear is non-zero.

The Shewhart chart thus proves useful to identify and demarcate transients in the operation of HVAC equipment. Since dynamic data from the equipment can be due to many factors, some totally acceptable, it is expert logic downstream that determines whether or not any transient so observed indicates a fault.

Identifying faults in devices under regulation

Besides identifying transients, control charts can detect faults directly. Automated functions based upon the CUSUM chart described in Appendix A put out binary logical states characterizing regulation explicitly, such as “tracking too high,” “tracking too low,” or “hunting.” These states are “primitive” in the expert system because analytical (e.g., chart) functions generate them directly from data with no other logic intervening. That action constitutes Capabilities (b) and (c).

The example of Figure 1 concerns temperature regulation, so discussion continues here in terms of a generic temperature T , although pressure or flow regulation could apply as well. Two incremental quantities based upon the set-point temperature T_{set} are updated upon each sample T_n of the regulated

temperature at time step n :

$$\Delta U = (T_n - (T_{set} + b))\Delta t_{ddc}, \quad (11)$$

$$\Delta L = (T_n - (T_{set} - b))\Delta t_{ddc}, \quad (12)$$

where b is a parameter chosen by the user to reflect the amount of dispersion in T_n that the chart algorithm is to allow. The BAS sampling period is $\Delta t_{ddc} = 1/12$ h (5 min) was used here—meaning the units of ΔU and ΔL are degree-hours per sample. These increments then tally into two chart registers A and C as follows:

$$A_n = A_{n-1} + q(\Delta U > 0)\Delta U, \quad (13)$$

$$C_n = C_{n-1} + q(\Delta L < 0)\Delta L, \quad (14)$$

where the function $q(\dots)$ converts the true and false results of logical expressions into the numbers 1 and 0, respectively. Non-zero A is always positive, and non-zero C is negative, as is the case with P and Q used by CUSUM. Unlike CUSUM, here each data sample affects at most only one register, affording a capability to identify hunting that CUSUM lacks.

Units of A and C are degree-hours, analogous to watt-hours in electric meters. The units are relevant because the user sets a warning parameter K_{warn} in them by which the chart algorithm judges quality of controller regulation via the following:

$$\text{tracksHigh} = ((A_n + C_n) > K_{warn}), \quad (15)$$

$$\text{tracksLow} = ((A_n + C_n) < -K_{warn}), \quad (16)$$

$$\text{hunts} = ((A_n > K_{warn}) \wedge (C_n < -K_{warn})), \quad (17)$$

where “ $\wedge$ ” is the logical “and” operator. The three outputs (left of each equal sign) are primitive states that indicate regulation of the quantity is satisfactory when all are valued “false” and faulty when any is valued “true”.

In the case of a quantity regulated to duplexed set-points, such as the zone thermostat of Figure 1, the band parameter in Equations 11 and 12 is here made redundant by the unregulated band between the two set-points. Band b is thus eliminated, and the increments become

$$\Delta U = (T_n - T_{set,cooling})\Delta t_{ddc}, \quad (18)$$

$$\Delta L = (T_n - T_{set,heating})\Delta t_{ddc}. \quad (19)$$

These increments then tally into four chart registers, A through D , as follows:

$$A_n = A_{n-1} + q(\Delta U > 0)\Delta U, \quad (20)$$

$$B_n = B_{n-1} + q((\Delta U < 0) \wedge (A_n > -B_{n-1}))\Delta U, \quad (21)$$

$$C_n = C_{n-1} + q(\Delta L < 0)\Delta L, \quad (22)$$

$$D_n = D_{n-1} + q((\Delta L > 0) \wedge (C_n > -D_{n-1}))\Delta L. \quad (23)$$

Non-zero B is always negative and non-zero D positive, and B and D are limited to no more than the additive inverse of A and C , respectively. Introducing registers B and D is better than treating duplexed set-points as two separate instances of the

single-set-point case. With separate single set-points the tool would need logic selecting in which register sums are relevant over any given period so other register sums can be disregarded to prevent false alarms. Instead, the three primitive logic states are determined without selective logic:

$$\text{tracksHigh} = ((A_n + B_n) > K_{\text{warn}}), \quad (24)$$

$$\text{tracksLow} = ((C_n + D_n) < -K_{\text{warn}}), \quad (25)$$

$$\begin{aligned} \text{hunts} = & ((A_n > K_{\text{warn}}) \wedge (B_n < -K_{\text{warn}})) \\ & \vee ((C_n < -K_{\text{warn}}) \wedge (D_n > K_{\text{warn}})), \end{aligned} \quad (26)$$

where “ $\vee$ ” is the logical “or” operator.

The three states defined above explicitly identify faults associated with regulation of a quantity. They also serve expert system logic downstream with evidence to diagnose more primary faults or faults having no explicit test of their own.

Identifying faults in devices exerting regulation

Capability (d) requires identifying faulty dynamics in quantities (e.g., position of valves or dampers or fan speed) exerting regulation on other quantities. For example, a hot water valve position Z regulates heating of the air that a fan-coil unit supplies to the zone whose temperature is plotted in Figure 1. There is no set-point for Z to track, thus no “tracks high” or “tracks low” fault states to consider. Instead, Z modulates as needed to maintain a set temperature in the air discharged from the unit. Various other factors external to this modulation loop influence Z , notably the fan, outdoor air damper position, and air temperatures in the zone and outdoors. So, Z cannot be expected to remain steady even if discharge air temperature remains steady. However, those external influences are typically slow relative to hunting, the principal fault mode. Thus, it is suitable to base diagnostics upon the mean of Z from a rainfall of past samples going back a specified number of steps in time. The register increments now become

$$\Delta U = (Z_n - (\bar{Z}_n + b))\Delta t_{ddc}, \quad (27)$$

$$\Delta L = (Z_n - (\bar{Z}_n - b))\Delta t_{ddc}, \quad (28)$$

where $\bar{Z}_n$ is the mean of Z as obtained from the rainfall buffer at the current time step n . As was done for regulated variables, these increments tally into two chart registers, A and C , according to Equations 13 and 14, and the registers then generate two logic states, swing and hunts:

$$\text{swing} = ((A_n + C_n) > K_{\text{warn}}) \vee ((A_n + C_n) < -K_{\text{warn}}), \quad (29)$$

$$\text{hunts} = ((A_n > K_{\text{warn}}) \wedge (C_n < -K_{\text{warn}})), \quad (30)$$

where “ $\vee$ ” is the exclusive “or” operator.

A “true” for swing indicates that the regulating quantity has changed in the same direction over several samples, something notable but not necessarily a problem. The margin implied by those samples depends upon K_{warn} . Given reliable values in swing and hunts, subsequent logic can check whether any “true” value obtained correlates with steady being “false”

from any acceptable causal factor, such as valid change of set-point, fan speed, or hot water supply temperature. If that check fails, a fault is suspected, triggering further diagnostic logic including user interaction.

Results from laboratory testing

The top panel of Figure 2 repeats that of Figure 1, so accumulations in registers A through D , seen in the second panel, can be compared directly to developments driving them. Winter weather causes zone temperature to modulate at the heating set-point for most of the plotted period, where only registers C and D accumulate value. These registers toggle zone temperature state hunts from “false” to “true” and back near Hour 14, 11 h after hunting visibly begins. A change to parameter K_{warn} could toggle the state sooner, but it is also desired that hunts not falsely toggle “true” during the acceptable modulation seen between Hours 05 and 12 on the second day. Since acceptable modulation can go on indefinitely, a balanced choice of K_{warn} and a register reset period is needed to prevent false alarms. Such a reset occurs at Hour 12 of Day 2. In the bottom panel, the state tracksLow toggles through “true” during the period of sluggish response between Hour 19 on the first day and Hour 03 on the second day. Subsequent logic can check for any correlation with a “true” hunts from charts of the hot water valve, fan, and outdoor damper. A coincidence infers that primary cause lies in a specific control loop, averting redundant alarms. The second panel of Figure 1 shows that the intervals of hunting and sluggish regulation of zone temperature correlate with similar behavior of the hot water valve. In fact, during those intervals, faulty valve controller gains had been applied deliberately as a test.

Figure 3 shows a fault in regulation of mixing box air temperature T_{am} by the outdoor air damper position Z_{dm} . Gray dashed lines in the top panel show the band b , by which the user sets an acceptable variation for T_{am} , here 0.5°C (0.9°F). In this case, regulation is satisfactory between Hours 08 and 12, where Z_{dm} is seen in the second panel to slowly trend open (outdoor air is shut off at 0 Z_{dm}) as daytime outdoor temperature rises. After Hour 12, however, a change occurs that affects regulation of T_{am} very adversely. The subsequent large swings in T_{am} above set-point are captured as repeated “true” values of the TamTracksHigh state seen in the bottom panel. Coincident hunting of the damper is captured by “true” values for the ZdmHunts state. Subsequent logical statements then combine these primitive states with other states, such as TamSetptSteady, to localize the cause of the change and determine whether a fault exists.

Additional aspects of the method

Managing chart registers

It is necessary to ensure sums accumulating in chart registers do not result in numerical overflows. Also, fault evidence can pass undetected if sums accumulating in registers grow large and “stale,” meaning much of the data contributing to them occurred so long ago as to be irrelevant or even deleterious

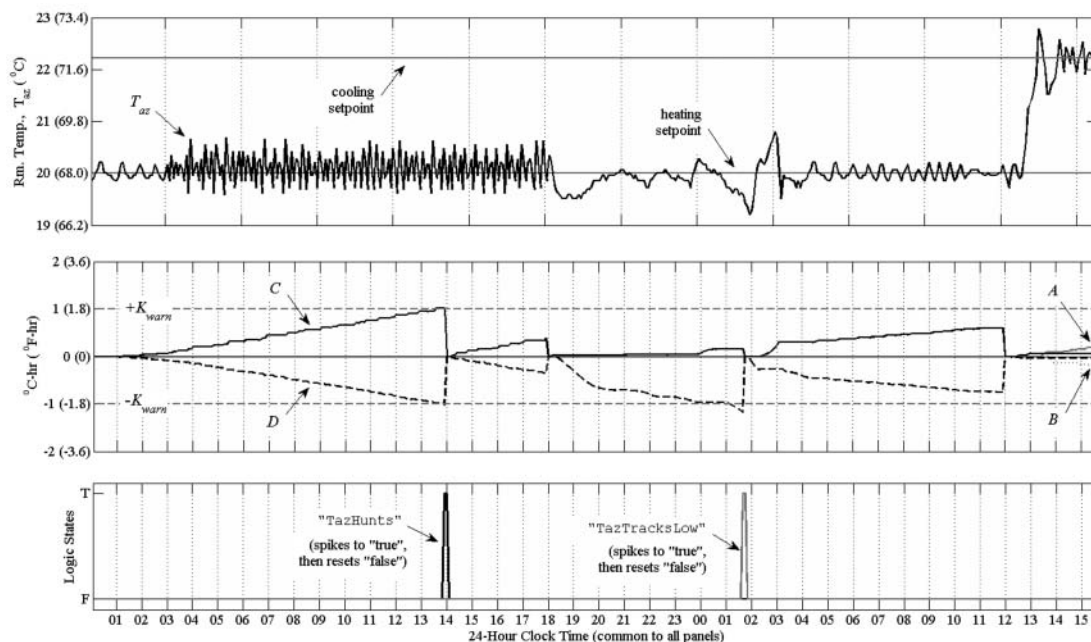


Fig. 2. Logic analysis sequence of hunting and drift in a regulated variable.

to current surveillance. Proper selection of allowance bands, such as b in Equation 11, permits fault-free operation to make only small additions to chart registers over time, although getting no addition is unlikely. Further, fault-free operation will always add large values to some registers in duplex set-point charts. One provision is resetting all registers to 0 when any of the three fault states test as “true”, a practice used in CUSUM. Unlike CUSUM however, here each data sample outside the accepted band does not tally to both positive and negative registers, making hunting detectable but also removing a feature that intrinsically helps keep registers bounded.

Another way to manage registers is through so-called “forgetting factors” applied to all register equations, such as F_f in this example for A :

$$A_n = A_{n-1} + q(\Delta U > 0)\Delta U - (F_f A_{n-1})q(\Delta U \leq 0). \quad (31)$$

Trials showed the value of F_f must be chosen carefully to not inhibit detection of hunting, so its use adds to the parameters needing expert attention. Since satisfactory results were obtained from periodic resets alone, forgetting factors were not tried further.

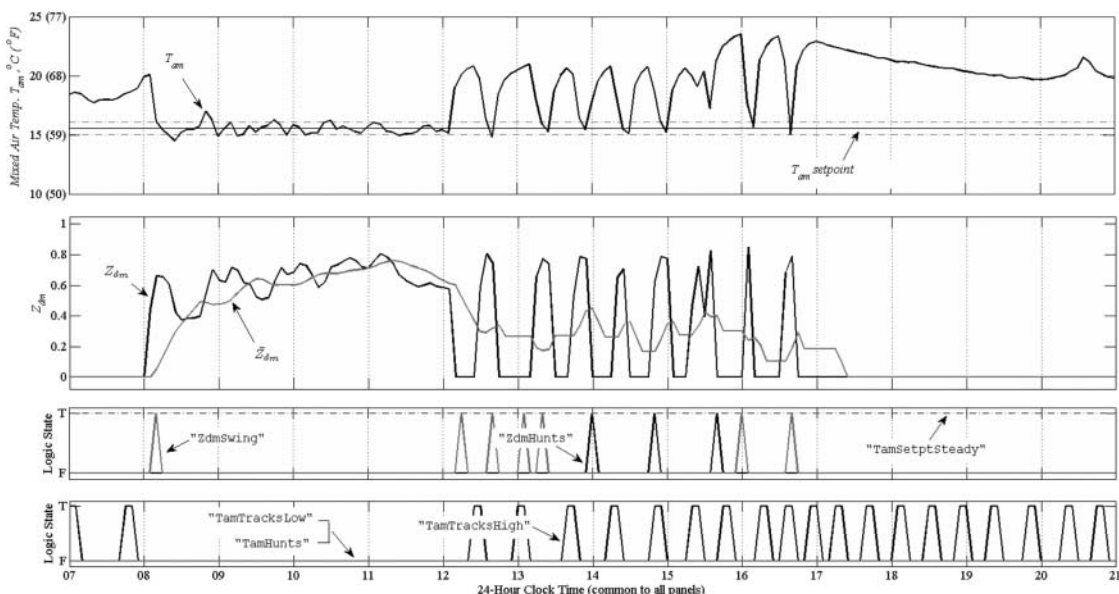


Fig. 3. Logic analysis sequence of hunting and drift in a regulating variable.

The solution taken here simply reset all registers to 0 at a specific period regardless of other factors. It was found here that reset periods longer than 18 h left the registers too stale to exploit recent data features, while periods below 10 h inhibited detecting slow-evolving faults. The heuristic demand to manually find robust settings for field deployments offers a future opportunity for fuller automation.

Field implementation

This method is being implemented in the field as one part of a larger, self-contained AFDD program running on a generic, general-purpose computer having open-protocol real-time network access to present value data from the BAS controllers existing in the building. This is in contrast to prior AFDD approaches, such as VPACC, intended to be embedded in the existing BAS products themselves, using the same, typically very constrictive, programming package the BAS vendor supplies its customers to set up conventional controls. That approach was reasonable in its era, when the hardware supporting the automation of buildings was much more limited and expensive than it is now. It took into account an environment lacking the comprehensive open-protocol networking and more capable hardware and programming interfaces that are available today at reasonable cost. Real-time data from the BAS network is not mandatory—the method described here also works on archived “trend logs”—but a real-time action lets the user answer diagnostic queries, asked by a suitably designed AFDD tool, with contemporaneous information not available from the BAS but beneficial to the tool’s effectiveness.

Rethinking the AFDD tool as a program networking autonomously from its own hardware and software frees it from the constraints of existing BAS products tailored primarily to enable technicians of little programming skill to set up and maintain simple, conventional HVAC control sequences. That orientation, lacking the usual capabilities of general-purpose computing, fully excludes programs of the sophistication described in this article. In buildings where the BAS can supply an autonomous AFDD tool with sensor data by open-protocol networking, the tool along with its hardware and maintenance could easily be an integrated product offered by a third-party specializing in HVAC AFDD and the having the advanced skills needed to create and service such tools for use by building staff. Recognizing that potential, no specific way to implement the chart or expert logic structures is given here. Researchers able to build upon the method already have the skill in general-purpose programming to implement it in laboratories, perhaps better than could be detailed here. The crucial issue instead is developing such methods into user-oriented commercial products of the building automation industry. Future collaborative development and field testing projects with HVAC industry partners are essential to build within that industry the new, cross-discipline competencies they need to make effective AFDD products a marketed reality.

Conclusions

Control charts combined with expert system logic offer a viable way to automatically detect and diagnose hunting and

other faults in HVAC system controllers. Historically involving recordkeeping and calculations done manually on paper, control charts now take computerized form as numerical registers “charting” the behavior of a monitored process by accumulating specified data, particularly by summing excursions of designated quantities beyond allowed bands. Autocorrelation in the data, traditionally a bane to control charts given continuous dynamic processes such as HVAC, becomes a diagnostic asset when expert system logic downstream of the charts is used to test for the cross-correlations normally expected between quantities having known physical relationships.

Equations 11 through 30 describe the “first generation” of charting functions keeping control loops under surveillance in a far more comprehensive AFDD tool for commercial HVAC systems, called “AFDD Expert Assistant,” being developed by the U.S. National Institute of Standards and Technology (NIST) for field testing in spring 2013. With the viability of control charts proven for that role, further research and field trials will aim for many improvements, for example, to eliminate delayed return of the logic state steady to true in Figure 1 and to eliminate the occasional spurious transitions of that state. Broader areas for future work include estimating critical parameters such as the band and alarm limits autonomously, and charting variation of the mean of a quantity as a way to extract frequency-domain information that could improve diagnostics.

References

- Betteley, G., N. Metrick, E. Sweeney, D. Wilson. 1994. *Using Statistics in Industry*. Hertfordshire, UK: Prentice Hall International.
- Fasolo, P.S., and D.E. Seborg. 1994. An SQC approach to monitoring and fault detection in HVAC control systems. *Proceedings of the American Control Conference, Baltimore, MD, June 1994*.
- Fasolo, P.S., and D.E. Seborg. 1995. Monitoring and fault detection for a HVAC control system. *HVAC&R Research* 1(3):177–93.
- Forsyth, R. 1984. Expert Systems, Principles and Case Studies, pp. 9–17, *The architecture of expert systems*. R. Forsyth, ed. London, UK: Chapman and Hall.
- Kim, M., S.H. Yoon, P.A. Domanski, and W.V. Payne. 2008. Design of a steady-state detector for fault detection and diagnosis of a residential air conditioner. *International Journal of Refrigeration* 31:790–9.
- Li, H., and J.E. Braun. 2003. An improved method for fault detection and diagnosis applied to packaged air conditioners. *ASHRAE Transactions* 109(2):683–92.
- Miao, T., and D.E. Seborg. 1999. Automatic detection of excessively oscillatory feedback control loops. *Proceedings of the IEEE International Conference on Control Applications, Kohala Coast Island, Hawaii, August 22–27*.
- Page, E.S. 1954. Continuous inspection schemes. *Biometrika* 41(1):100–15.
- Ryan, T.P. 2000. *Statistical Methods for Quality Improvement*, 2nd Ed. New York: Wiley & Sons.
- Schein, J., and J.M. House. 2003. Application of control charts for detecting faults in variable-air-volume boxes. *ASHRAE Transactions* 109(2):671–82.
- Shewhart, W.A. 1931. *Economic Control of Quality Manufactured Product*. New York: D. Van Nostrand.
- Shiffler, R.E. 1988. Maximum z-scores and outliers. *The American Statistician* 42(1):79–80.
- Summers, H., and C. Hilger. 2012. *Fault Detection and Diagnostic Software*. Emerging Technologies Program Report ET11PGE3131, Pacific Gas and Electric Company, San Francisco, CA.

- Tudoroiu, N., M. Zaheeruddin, C. Chiru, M. Grigore, and E. Tudoroiu. 2009. Unscented Kalman filter and frequency analysis techniques used for fault detection, diagnosis and isolation in heating, ventilation air conditioning systems—comparison results. *Proceedings of the IEEE International Symposium on Industrial Electronics, Catania, Italy, May 21–23*.
- Westphalen, D., K.W. Roth, and J. Brodrick. 2003. System & component diagnostics. *ASHRAE Journal* 45(4):58–9.
- Wise, S.A., and D.C. Fair. 1998. *Innovative Control Charting*. Milwaukee, WI: ASQ Quality Press.
- Young, T.M., and P.M. Winistorfer. 2001. The effects of autocorrelation on real-time statistical process control with solutions for forest products manufacturers. *Forest Products Journal* 51(11/12):70–7.

Appendix A: Control charts and expert systems

Certain fundamentals of control charts or expert systems may be unfamiliar to many in the HVAC research community but are important for AFDD of HVAC systems. This appendix summarizes those fundamentals as needed to serve this article.

Origin of control charts

With the advent in the early 20th century of large-scale manufacturing involving standardization and interchangeable parts, control charts began as the practice of periodically plotting specific metrics related to product quality as time series on paper graphs. The earlier practice of simply testing finished parts to discard rejects had become uneconomical, and in control charts, producers found a way to track key quality indicators as data sampled over time, responding proactively to avoid waste if warning limits are approached. Not only could the charts track expected devolutions, such as tooling wear, but by charting multiple production metrics, correlations could be drawn to reveal unanticipated factors influencing product quality. Products such as electrical fuses and light bulbs emerged, whose quality needed to be quantified consistently, verifiably, and economically for large lots while allowing for metrics requiring that individual examples be tested to destruction. Those issues were thoroughly studied in the 1920s by a statistician at Western Electric Company, Walter Shewhart, leading to his seminal book (Shewhart 1931) establishing control charts as analysis tools within a larger, more comprehensive framework now known equivalently as statistical quality control (SQC) or statistical process control (SPC). In this context, “control” has a broader connotation than the closed-loop feedback regulation associated with that word in HVAC, where the timescale between measurement and automated compensating action is typically only a few seconds. In SPC, the compensating action is generally not automatic, its timescale can range from minutes to days, and it considers both defined and hidden factors, suggesting something more akin to a researcher’s notion of “experimental control.” Further, SPC takes “process” to encompass all the factors in an item’s manufacture whether physically coupled or not, again a broader notion than usually seen in HVAC literature.

Within SPC, many types of control charts have been established for specific industrial uses, and a common practice to aid diagnostics is to track the sample mean and dispersion

on separate control charts having distinct warning limits and action procedures (Wise and Fair 1998; Betteley et al. 1994). Two control chart types are considered here for AFDD of HVAC systems: the Shewhart chart and the CUSUM chart.

The Shewhart chart

The general category of control charts bearing Shewhart’s name are distinguished by their decisive use of the IID (independent, identically distributed) concept from statistics. To be IID, the value of any one sample of the observed quantity can have no relation to the value of any other, apart from the presumption that they all come from a common generating process. Further, IID means the dispersion that this process exhibits in generating samples under normal (i.e., natural, varied only by an accepted degree of random “chance”) conditions can be characterized over time by one statistical distribution of constant parameters. It is typical but not a necessity that this actual “normal” distribution is presumed a Gaussian normal distribution, so the term “nominal” is used for it here. As long as its samples continue to exhibit the nominal distribution, the process is considered statistically “under control.”

Shewhart (1931) considered the variation in the same characteristic measured (a metric) on successive samples of a manufactured item as having two components. One component is the variation assignable to deterministic causes whether identified or not, meaning all the dynamic features in the data that are understood well enough to at least know they are not random. The second component is all the variation remaining. That is, all variation remaining in the samples is provisionally “chance” by default. These two variation components are known, respectively, as *special-cause* and *common-cause*. Shewhart’s idea was to use the chart to move avoidable variation from the default, common-cause component to the special-cause component and then out of the process entirely.

Given a series of IID samples, a Shewhart chart reveals when there is a special-cause component in their variation, enabling current or impending process trouble to be assigned to a causal factor and fixed. Tighter parameters can then be applied to the chart iteratively, revealing successively deeper special-cause factors, the common-cause variation being progressively whittled away as more of it is assigned to fixable causes and eliminated. Upon reaching a practical minimum in common-cause variation, charting of the metric is then continued as surveillance against emergent special-cause factors.

Setting parameters for the nominal distribution also implicitly defines which factors influencing the process are common-cause factors. As an example in parts manufacturing, the imprecision accepted in a caliper measurement becomes a common-cause factor, and wear of a cutting tool detectable as a special-cause factor, depending on how much caliper imprecision the nominal distribution allows. Intended or not, some degree of tool wear ends up as common-cause if the nominal distribution allows for more caliper imprecision than is typically the case. Perhaps two milling stations both feed a single caliper test station, with one milling station consistently having less precision in placing its cutter. This creates an assignable special-cause factor that can be lost within the common-cause

variation charted at the caliper, since any disparity between the milling stations would be unknown when the caliper chart parameters are first set. The fault at the errant milling station can be revealed by the caliper chart, however, if variance changes are caught as special cause and associated to periods when either mill is offline. Automating for HVAC systems that same schema of detection and logical association is the goal of teaming control charts to expert logic as previously discussed.

Thus, surveillance of a process by a Shewhart chart traditionally requires addressing application details carefully, and often heuristically, for each metric charted. Parameters for the nominal distribution must allow emerging special-cause factors to be identified promptly, while minimizing false warnings on factors properly common cause. The parameters are implemented on a paper Shewhart chart by way of one or more pairs of parallel lines—commonly termed “limits,” but in fact acting as the thresholds for sample counting bins—beyond which a sample counts toward the tally expected by the nominal distribution. A paper chart shows by graphs and tables the number of samples counted beyond its threshold lines over a moving interval of time. The length of that time interval, the *run length*, results implicitly from the distribution chosen as nominal, as do the location of threshold lines and the counts expected in the overlapping bins they delineate. Any instance of those counts exceeding what is nominally expected is taken as evidence that an assignable special-cause factor has entered the process.

For example, presuming a Gaussian nominal distribution having standard deviation σ , no more than 31 discrete counts beyond threshold lines at $\pm\sigma$ about the mean are theoretically expected (given the samples are IID) over a moving run length of 100 samples. A thirty-second count could in practice be taken as evidence that a problem has developed, although in theory, that diagnosis is not certain because the nominal distribution reflects probability only. It could be asked which of the 32 samples over the run time is to be regarded as the discrepant one. Supporting evidence is gained by binning other thresholds, for example at $\pm 2\sigma$, $\pm 3\sigma$, or unilaterally. Beyond the $\pm 3\sigma$ threshold, only two counts are nominally expected over a 1000-sample run length, an average of 1 in 500 samples, so a second count in 500 could help isolate the trouble in time, as would any graphically observable clustering of samples. Clearly, challenges arise in automating traditional, graphically based Shewhart charts, notably in the potentially long run lengths needed and in replicating the human graphical cognition of sample clustering over time.

The CUSUM chart

The Shewhart chart is suited to detect swift variance changes in a metric, such as that caused by cracked tool bits in manufacturing or by actuators suddenly departing steady operation in HVAC. Detecting a small or slow drift in the metric mean, like tool wear in the example above, is problematic for a Shewhart chart because the drift may go undetected for a long time if the common-cause variance is relatively wide. A complementary SPC tool for such situations is the CUSUM chart, first introduced by Page (1954). Of the variants now developed,

Ryan (2000) defined a CUSUM chart having a *slack value*, k , used in conjunction with the z -score z (see Equation A3) of each sample at time n . Samples observed either above or below a band of $\pm k$ about the expected mean are accumulated in separate numerical registers:

$$P_n = \max[0, ((z_n - k) + P_{n-1})], \quad (\text{A1})$$

$$Q_n = \min[0, ((z_n + k) + Q_{n-1})]. \quad (\text{A2})$$

Positive z_n greater than k moves sum P further above 0 and sum Q toward 0. Negative z_n less than $-k$ moves sum Q further below 0 and sum P toward 0. Drift in the metric mean is signaled by exceeding alarm thresholds placed upon P and Q .

Automating control charts for HVAC systems

Automating control charts for computer implementation means, among other issues, reducing the need for human graphical cognition as much as is practical. The AFDD method presented here does not exploit sample clustering, although automating its recognition could help diagnostics. Of the traditional paper chart features, only threshold lines are automated here, done by the sample z -score. Parameters defining a normal distribution are the mean $\bar{x}$ and standard deviation s of a sample set. A z -score z_n is calculated for each sample x_n :

$$z_n = \frac{(x_n - \bar{x})}{s}. \quad (\text{A3})$$

Common statistical practice is to standardize the z -score by sample size, which here is the run length used by any particular threshold line. Shewhart (1931) rigorously considered sample size because a common use of his charts is to infer the quality of a large number of products by testing small sampled sets. However, the parameters of populations are not being estimated here, so standardizing by sample size is not necessary. Further, it is not wanted, because Shiffler (1988) showed that if z -score is standardized to sample size, no z -score beyond 3 is even possible unless at least 11 samples are considered. That would be problematic if a long sampling period of several minutes is used to minimize traffic in a real-time data network, because it would force longer run lengths that inhibit detecting faults early.

Unlike processes in manufacturing that generate discrete items having metrics amenable to the IID assumption. HVAC processes involve continuous time series of quantities sampled from fluid streams, such as temperature and velocity, where physical couplings often exist between successive samples. These couplings result from inherent properties such as the specific heats of fluids and heat transfer media, and arise in the sampled data series mathematically as autocorrelation. Time series from HVAC processes, such as air temperature leaving a cooling coil, also normally exhibit frequent dynamics that are the acceptable results of exogenous influences, like control intervention, occupancy, weather, or time of day. If the time series is viewed in isolation, these dynamics also appear as autocorrelation. But expert knowledge of the

system can readily point to cross-correlation with other dynamic variables, typically with appreciable lag, such as CHW valve position in the case of LAT. Either form of correlation violates the IID premise and, thus, theoretically undermines control charts as SPC traditionally employs them to explicitly isolate instances of special-cause variation as faults to be fixed.

Fasolo and Seborg (1994) used Shewhart and CUSUM charts for AFDD of HVAC equipment, dealing with autocorrelation by charting only residuals between sample data and a first-principles dynamic process model of the particular device tested. Schein and House (2003) similarly approached AFDD of VAV units, charting residuals derived from prior dynamic tests of the same units in a laboratory. Young and Winistorfer (2001) surmised that autocorrelation puts an assignable special-cause variation into the samples that traditional control chart methods cannot separate from common-cause variation. Thus, the usual capability to iteratively isolate and remove special-cause factors becomes stuck at some unacceptably high level of common-cause variation. They addressed the problem by digital filtering to cancel autocorrelation out of the data upstream of their computerized charting algorithms. These past approaches share the disadvantage of needing laborious human expert a priori information peculiar to the size, configuration, and dynamic characteristics of each specific unit put under AFDD surveillance. That modeling, bench-test data, or filtering must then remain valid enough through all operational contingencies to maintain the surveillance with minimal false and missed alerts. Otherwise, charting (and, hence, AFDD surveillance) can be done only during those relatively rare periods of unchanging static operation, during which autocorrelation becomes irrelevant. This latter, more restrictive, course was the one taken by Schein and House (2003), as described in the main body of this article. All these disadvantages are significant enough to motivate exploring a different approach.

The method described here adapts automated versions of the Shewhart and CUSUM charts to be analysis tools that future work will more fully integrate within a larger, more comprehensive AFDD tool of expert system architecture. The Shewhart chart, instead of discriminating faults explicitly, is used only to signal the starts and stops of dynamics in quantities that the expert logic can then check for any normal correlation to other similarly charted quantities. The CUSUM chart is adapted for a diagnostic role not inhibited by correlation, due to the availability of complementary expert system logic that the Shewhart charts make possible.

Expert systems

An “expert system” is a computer program that replicates or enhances the reasoning and judgment that a human, having a specific expertise, exercises on data held for evaluation and action using that expertise. Expert systems are a major field within AI, thus rating exhaustive treatments in foundational AI texts. Forsyth (1984) provided a brief, practical primer defining expert systems as all distinctively having at their core two complementary parts: (1) a “knowledge base” that typically is the body of facts, rules, and relationships (e.g., a model or lookup table) formulating what the expert knows and (2) an “inference engine” that moves the information given to the program through the knowledge base in a manner expressing expert reasoning and judgment. For the purpose of this article, it is sufficient to know the following points. The programmed expressions defining rules and relating states, such as LATregSteady, form some parts of the knowledge base. The inference engine is a probabilistic algorithm evaluating those states and rules given BAS data and supplemental information the user provides, transforming the data into advice the user can act upon. This article only addresses knowledge base parts in an expert system.