Performance Evaluation of Dynamic Monitoring Systems:

The Waterfall Chart

by

George Box, Søren Bisgaard, Spencer Graves, Murat Kulahci,

Ken Marko, John James, John Van Gilder, Tom Ting, Hal Zatorski and Cuiping Wu

Abstract

Computers are increasingly employed to monitor the performance of complex systems. An important issue is how to evaluate the performance of such monitors. In this article we introduce a three-dimensional representation that we call a "waterfall chart" of the probability of an alarm as a function of time and the condition of the system. It combines and shows the conceptual relationship between the cumulative distribution function of the run length and the power function. The value of this tool is illustrated with an application to Page's one-sided Cusum algorithm. However, it can be applied in general for any monitoring system.

Introduction

It is important to surround complex systems with monitoring schemes that can indicate when a system malfunctions. This article considers the problem in relation to the auto industry's legally mandated "on-board diagnostics" (OBD), now required for all new automobiles sold in the US, Canada and Europe (see e.g. CARB, 1997) to detect malfunctions in the powertrain as precursors of potential emissions problems. However, the issues discussed have wide applicability outside this area.

In this article we first review general concepts for monitoring dynamic systems as a generalization of standard (static) monitoring schemes used for industrial quality process control, see e.g. Box et al (2000). We then discuss how to evaluate the performance of monitors, focussing particularly on the waterfall chart, which is a three-dimensional representation simultaneously showing the power and the cumulative run length distribution. The concept is illustrated by applying it to a standard one-sided Cusum chart. This is followed by a general discussion.

What is a Dynamic Monitoring System?

Static process monitoring is a well-known concept discussed extensively in the industrial quality control literature under the umbrella of Statistical Process Control (SPC). To be consistent with the engineering control literature we will call the system to be monitored the *plant*. In quality control context, it is typically assumed that the plant is normally in a state of statistical control, specifically that the plant generates data $y_t, t = 1,...,n$, that vary independently with a fixed variance σ^2 about a fixed mean μ. The purpose of monitoring is to detect any deviation from this stable state.

In our application the plant does not necessarily have a fixed mean. Thus we wish to design a monitoring system that will show whether the plant is in a satisfactory condition or is malfunctioning while taking account of its dynamic behavior. For example, consider the cooling system (plant) in a car. Despite the various feedback control systems the cooling water temperature is expected to rise as the car makes its way over a mountain. However, if the temperature suddenly increases more than should be expected, it may be a sign that the cooling system is malfunctioning. A dynamic monitor will continuously sample temperature data and compare these readings with calculated temperatures based on a model for a normally functioning car under these driving conditions. Such a dynamic monitor will therefore be able to detect subtle malfunctions of the cooling system.

In Figure 1 we exemplify the concept of a dynamic monitoring in the context of an on-board diagnostic for automobiles. However, the same principles apply to any monitoring system, dynamic or static. The plant receives inputs $\mathbf{x}_t = (x_{1t}, x_{2t}, \ldots, x_{kt})'$ and produces outputs $\mathbf{y}_t = (y_{1t}, y_{2t}, \ldots, y_{mt})'$ sampled at discrete, equi-spaced time intervals. Thus in automotive applications the plant could for example be the intake manifold including the feedback systems already in the plant.

A *model module* $f(\mathbf{x}, \boldsymbol{\theta})$ where θ is a vector of model parameters accepts data $\mathbf{x}_t$ on the input to the plant and estimates as closely as possible its expected performance when functioning properly. The model could be empirical, or it could be based on physical theory, a neural network or some combination of these. Outputs $\mathbf{y}_t$ from the plant are compared with those from the model and residual differences $\mathbf{r}_t = \mathbf{y}_t - f(\mathbf{x}_t, \theta)$ are computed. If the system functions correctly and the model of the system is an appropriate time series, the residuals will form a stationary series of random noise. A signal contained in such noise can indicate that the system is malfunctioning. To trigger an alarm, the residual noise is processed by an appropriate algorithm. For example, if the signal expected is a step change then a Cumulative Sum (Cusum) is appropriate. However, if a trend or gradual chance is expected then an Exponentially Weighted Moving Average (EWMA) algorithm should be used. By using the Cuscore concept, see Box and Luceno (1997), algorithms may be developed that are designed to detect other malfunction patterns. When the algorithm detects a signal, the operator is notified by turning on the Malfunction Indicator (MI). A decision to turn on the MI is made by the decision module, which typically contains a few adjustable parameters that can be used to fine-tune the monitor. The decision module together with the model module constitutes the monitoring system.

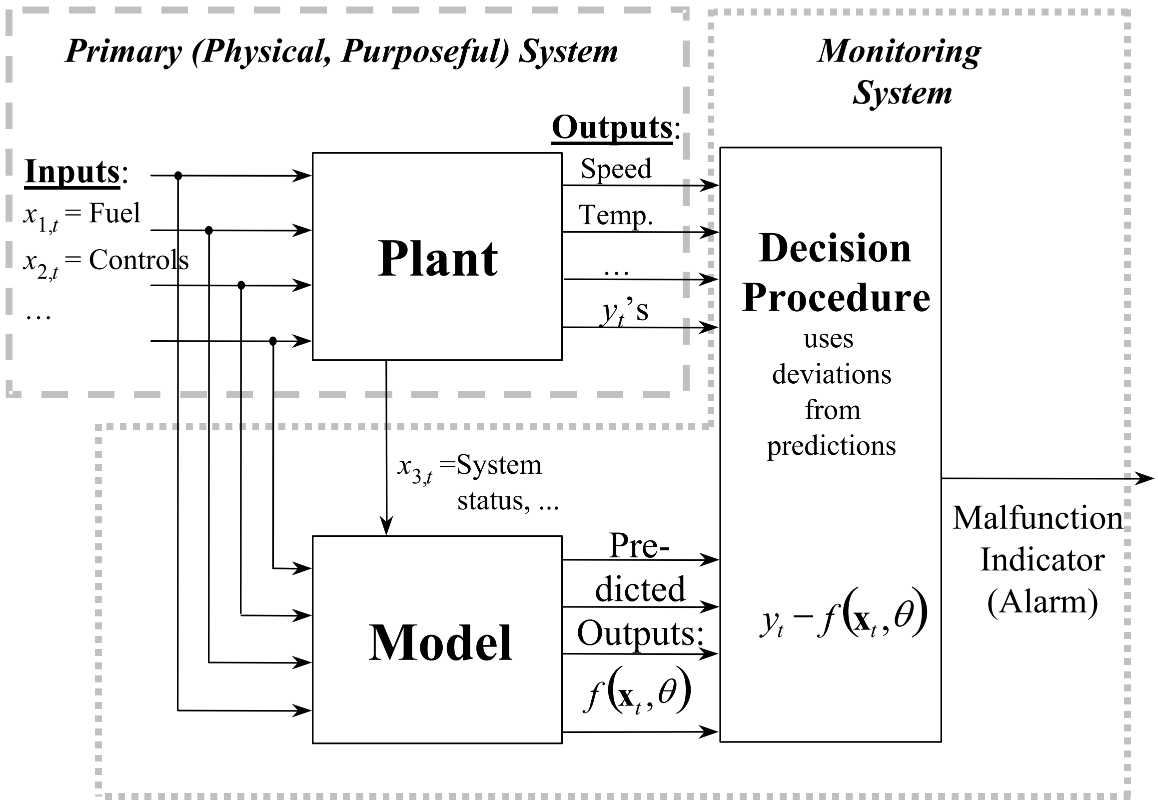

Figure 1. A conceptual model of a dynamic monitoring system.

Type I and II Decision Errors

Whether the residuals show sufficient signs of a signal to warrant a malfunction indication is complicated by the fact that the inputs and the plant itself are subject to random variations, measurement errors and model misspecification. Thus no matter how well we design the decision module, false alarms occur occasionally, turning on the MI when it shouldn't. This is called a Type I *decision error*. The possibility that the MI is not turned on when it should be is called a Type II *decision error*.

Both types of errors can reduce the practical usefulness of a monitor. In the automotive context a false alarm or a Type I error means that the owner is required to repair the car when nothing is wrong. This can lead the owner to ignore the MI even when something is wrong. Alternatively failure to detect a valid alarm implies a Type II error where the vehicle may be malfunctioning without the operator being appropriately notified.

If monitors have high error rates, it may be difficult to enforce the law. Therefore we need monitoring systems that are sufficiently quick to respond to real malfunctions, thereby avoiding a persistent Type II error, while at the same time minimizing the probability of a Type I error or false alarm. This requires good predictive models and efficient decision procedures using appropriate performance criteria. Before discussing a general approach to evaluating the performance of a given monitoring procedures we need to consider the condition of the plant.

The Condition of the Plant

The idea that a monitored systems (plant) is either "good" or "bad" is too simplistic. Cars start wearing out immediately after they come off the production line. Thus we need to consider a continuum of conditions, *s* as illustrated in Figure 2. In the automotive context it has become standard to use the following terminology. At the left hand side of the continuum, we have what we denote "as new" and at the other extreme, "hard failure". Somewhere in between "as new" and "hard failure" we define regions of "good" and "bad" separated from each other by "worst acceptable" (w.a.) and "best unacceptable” (b.u.) conditions. There may also be a “no-man’s land” in between w.a. and b.u. that is not defined as either good or bad as we will explain.

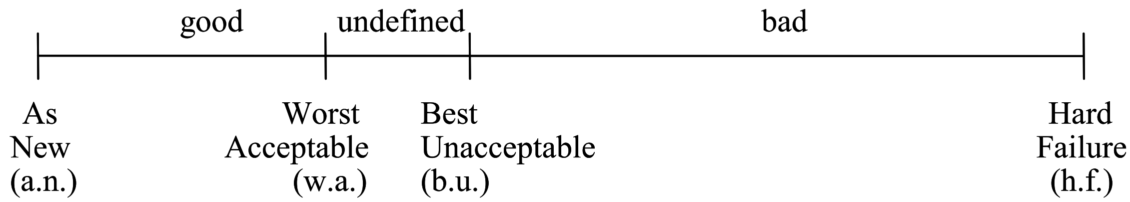

Figure 2. The Range of Possible Conditions of the System.

The Run Length Distribution

The key requirement for a monitoring system is that it is sufficiently quick to set an alarm when the system is “bad”, but hardly ever responds as long as the system remains “good.” Expressing this in terms of probabilities, when the system is bad we want a relatively high probability that the alarm will be turned on within a short time interval. However, if the system is good, we require that over a given long time interval the probability of an alarm is very small. This explains the need for the “undefined region” in Figure 2. If w.a. = b.u., that condition would both be good and bad, and a monitor would have to meet conflicting requirements for a plant in such conditions.

Now let the probability that an alarm will be triggered at least once in the time interval $(0, t]$ when the plant is in state s be $\Pr\{\text{Alarm at } T \le t | s\}$ and the intervals between samples be the unit of time. The probability is a function of t and s. Thus for a given system, we may in principle plot the probability of an alarm

$$f(t, s) = \Pr\{\text{Alarm at } T \le t \mid s\} \tag{1}$$

as a function of *t* and *s*, hence producing a three dimensional surface which is shown in Figure 3 and discussed later. Because of its appearance we call this a "waterfall chart".

Page's One-sided Cusum Algorithm

We now briefly discuss the Cusum algorithm that will be used to illustrate the waterfall chart and its value in evaluating certain properties of the run length distribution. For a series of observations $\{r_t\}$ with mean μ, a Cusum monitoring scheme in its simplest form is a cumulative sum Q_t of deviations $r_1 - \tau,\ r_2 - \tau, \ldots$ from the process target value τ

$$Q_t = \sum_{j=1}^{t} (r_j - \tau) .$$

In the dynamic monitoring context, r_t will be residuals obtained by subtracting model generated predictions from real observations.

A monitoring scheme can be constructed by plotting successive observations of Q_t. If the process has a mean equal to the target $\mu = \tau$, the Cusum will behave like a random walk fluctuating around zero. However, if μ suddenly increases even by a small amount, Q_t will contain an increasing deterministic component and plots of Q_t against *t* will increase more or less linearly with a slope of $\delta = \mu - \tau$. Similarly, if the mean decreases then Q_t will start to decrease. Thus the Cusum can be used to detect any departures of the mean from a given target value.

In some applications, it is known that an unfavorable departure from status quo will be in a particular direction. In these cases, the monitoring system is specifically aimed at detecting shifts in this one direction and it is convenient to use a Centered

Cusum. Suppose for example that μ_1 is the best unacceptable (b.u.) condition of the plant and that $d = \mu_1 - \tau$. Then a centered one-sided Cusum statistics would be defined as

$$Q_t^* = \sum_{j=1}^{t} (r_j - \tau - d/2).$$

The centered Cusum is essentially a handicapped score. When the mean of the data is on target the quantity $d/2$ will produce a downward trend in Q_t^*. Thus when Q_t^* falls below zero, it is automatically set to zero. However, when Q_t^* increases and exceeds a certain threshold, which we denote by *h*, an alarm is triggered. For computer implementation this algorithm is best put into practice by sequential calculations of

$$Q_t^+ = \max\{0, r_t - \tau - d/2 + Q_{t-1}^+\}.$$

The Cusum algorithm was invented by Page (1954) and has since been studied extensively. It is particularly well suited to monitoring processes where a malfunction corresponds to a deviation from the mean in a specific direction.

The run length is defined as the time it takes before an alarm is sounded. It has been the tradition to design a monitoring scheme by selecting the detection threshold, *h*, after considering the resulting Average Run Length (ARL) for a good system and a bad system. However, as Page (p. 101) warned, the ARL is only one of several possible measures of the performance of a monitor. To understand the behavior of a given system it is necessary to consider the entire run length distribution. In particular for a good system with a large ARL the run length distribution is approximately exponentially distributed and consequently a large percentage of false alarms can occur at times much shorter than the ARL.

The Waterfall Chart

Although under certain assumptions about the distribution of the data, the run length distribution may be computed analytically, more generally it is necessary to use computer simulation to estimate the run length distribution and the ARL and we employ this method here.

Suppose the condition, *s,* of the plant is determined by the mean $\mu = \mu(s)$ and that for $s =$ worst acceptable, $\mu = \mu_0$ and for $s =$ best unacceptable, $\mu = \mu_1$ with $\mu_0 < \mu_1$ and that the observations $\{r_t\}$ are independent normally distributed with mean μ and constant variance σ^2. For given values of *t* and *h*, we repeatedly generate random normally distributed data with variance σ^2. Based on these as input to the Cusum algorithm Q_t^+, we compute the number of alarms hence getting estimates of $f(t,\mu) = \Pr\{\text{Alarm at } T \le t \mid \mu\}$. Further repeating this for a grid of values of *t* and μ spanning a region relevant to our application, we compute estimates at a sufficient number of points to generate the surface $f(t,\mu)$. This surface is shown in Figure 3 for a system with μ_0 (worst acceptable) = 1, μ_1 (best unacceptable) = 1.5, σ = 0.15, $d/2 = (\mu_1 + \mu_0)/2$ and an alarm occurs when Q_t^+ exceeds the detection threshold *h* = 0.5.

This example is typical of the monitors used for on-board diagnostic applications. One observation is observed per trip and we require for a best unacceptable condition a high probability of an alarm set in two trips. Also for a worst acceptable condition we require a relatively low probability of alarm in the design life of the plant (say 100,000 trips).

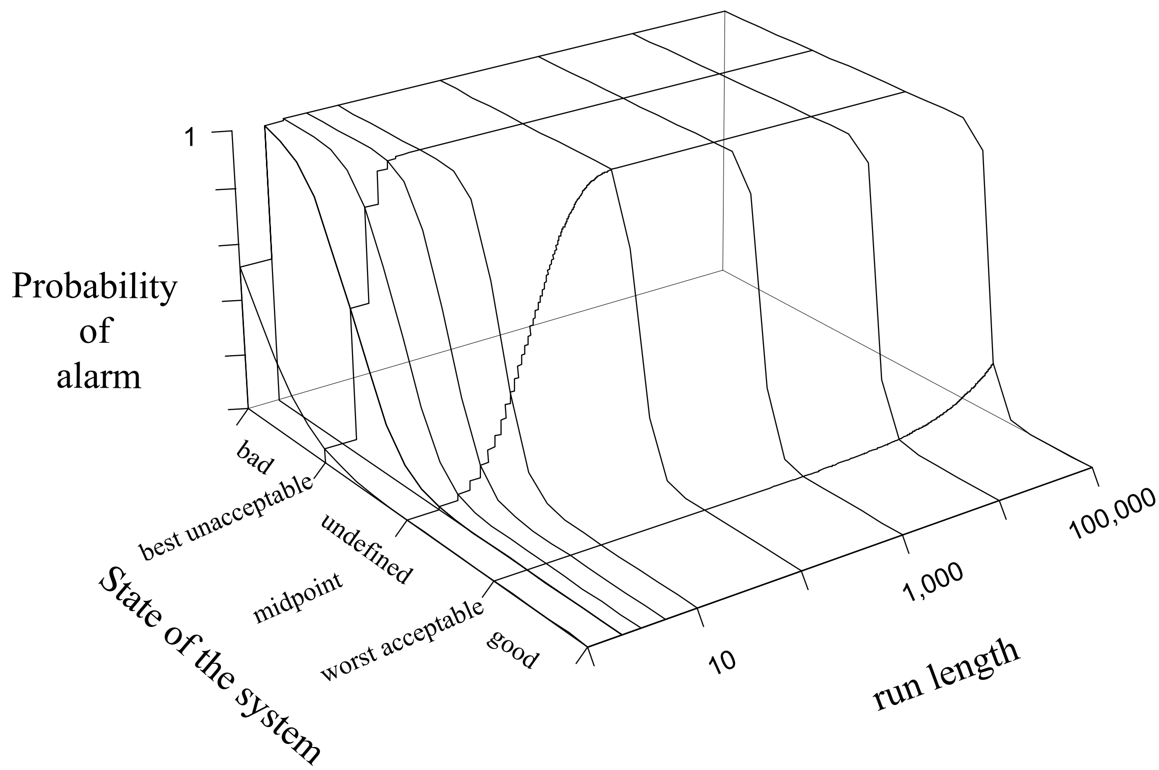

Figure 3. Probability of Indicating a Malfunction within a Given Time for a Particular Condition of the Plant.

To better appreciate the information that can be gained from Figure 3, Figure 4 shows the surface sliced for a constant condition $\mu(s)$ of the plant. This figure shows the cumulative distribution function of the run length $f(t, \mu = \text{midpoint}) = \Pr\{\text{Alarm at } T \leq t \mid \mu = (\mu_0 + \mu_1)/2.\}$. Thus for a plant that is constantly "best unacceptable (b.u.)" the probability of detecting a b.u. malfunction in two trips is more than 50 percent.

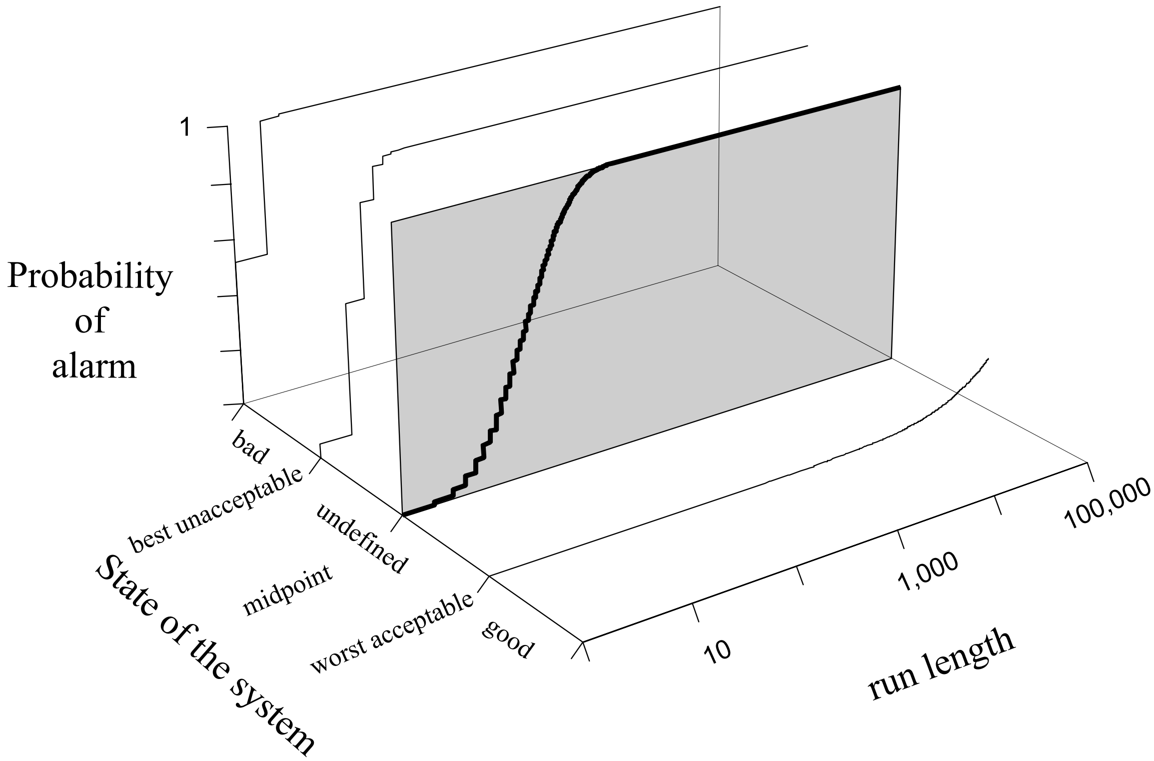

Figure 4. The Cumulative Distribution Function Is the Probability of Indicating a Malfunction at Different Run Lengths for a Particular Condition of the Plant.

Figure 5 shows the surface sliced for a specific run length t_0 so $f(t = t_0, \mu) = \Pr\{\text{Alarm at } T \le t_0 \mid \mu\}$. This graph is the *power function* for a particular run length from which we can read off the probabilities of making Type I and II errors. Thus for a given algorithm and adjustable threshold parameters, the waterfall chart combines information about the run length distribution and the power function which are both relevant for a specific application.

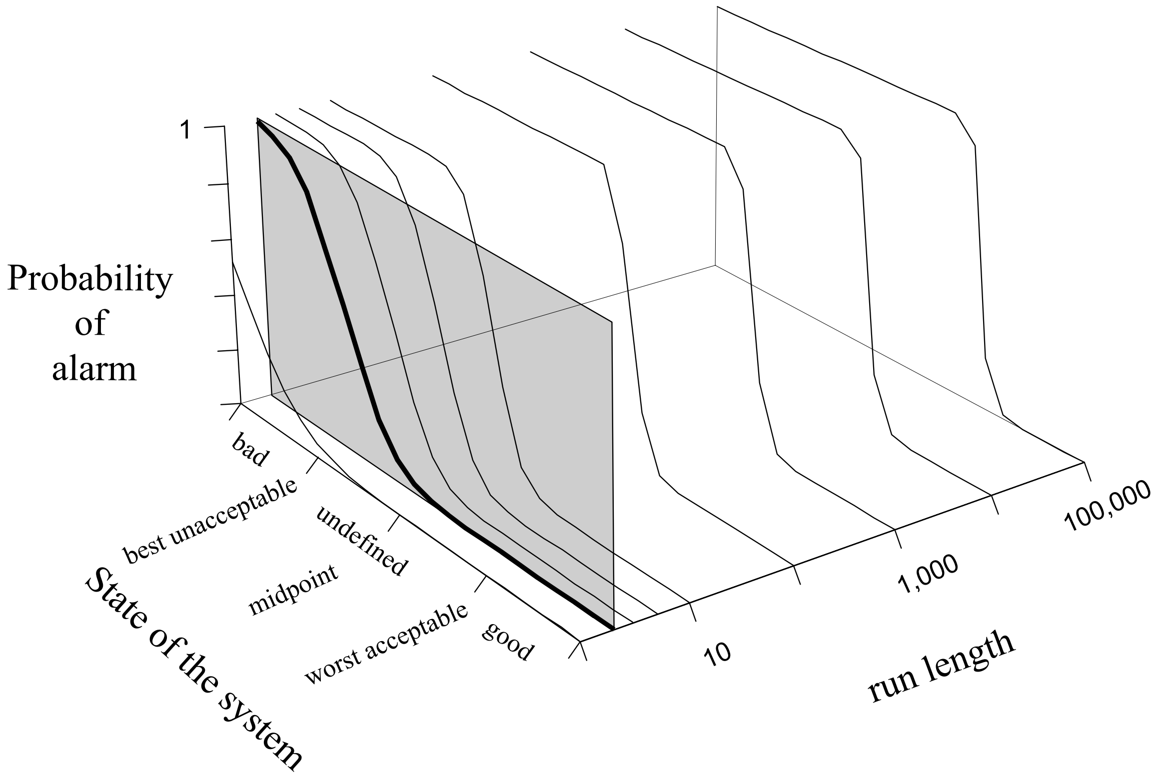

Figure 5. Power of the Monitor at a Particular Run Length.

If we slice the three-dimensional surface for a specific *probability*, we get a contour plot, which can be used for threshold selection and sensitivity analysis as discussed in the next section.

The waterfall chart shows the probability of an alarm for given fixed conditions. One of the reasons we are interested in evaluating the performance of a monitoring algorithm, however, is that we expect the plant to deteriorate. The waterfall chart is therefore not completely realistic. To be more realistic we would need to know precisely how the plant would deteriorate. Unfortunately the manner in which the plant will deteriorate is seldom known. Thus the waterfall chart is constructed with the (albeit unrealistic) assumption that the condition that the plant remains constant throughout its remaining life.

Threshold Selection and Robustness Analysis

The waterfall chart in Figure 3 is computed based on a specific detection threshold, *h*. In applications we often need to select *h* to achieve certain run length requirements as nearly as possible. One important assumption is that the process standard deviation σ is known; it is important therefore to evaluate the monitor's robustness to misspecification of σ.

The graph in Figure 6 shows five percentiles of the cumulative run length distribution of Q_t^+ (probability of alarm = 0.01, 0.1, 0.5, 0.9, and 0.99) for *h* = 0.1, 0.3, and 0.5 and for σ = 0.1, 0.15, and 0.2. We see that as the standard deviation increases, the waterfall in general becomes flatter and the probability of an alarm increases for most but not all conditions of the plant. These sub-graphs display a pattern that is general and useful for aiding the kind of intuition needed to successfully fine-tune a monitor.

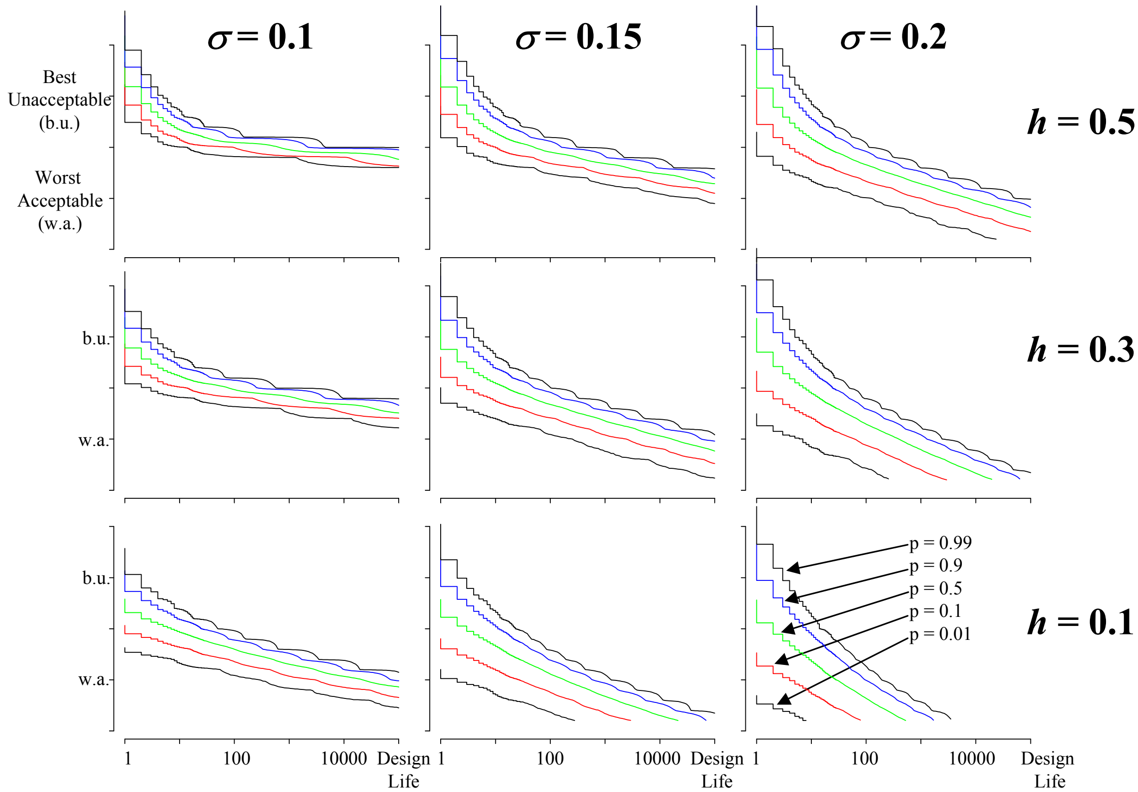

Figure 6. Threshold Selection and Robustness Evaluation
(3100 simulated runs)

From Figure 6 we also see that as the threshold increases, the entire waterfall moves back, with each percentile occurring at a higher run length or a worse condition. Increasing the threshold also makes the waterfall steeper. This is to be expected because the variance of Q_t^+ increases roughly linearly with t, the number of observations included. We might therefore expect that the variance might increase roughly linearly with the threshold, as well. This means that the threshold divided by the standard deviation of Q_t^+ should increase roughly in proportion to $\sqrt{h}$ corresponding to an increase in the statistical power of the procedure indicated by an increasing steepness of the contours as h increase in Figure 6.

Graphs like Figure 6 may be used as a tool for designing and fine-tuning monitors in a number of ways. We will illustrate with Q_t^+ but the idea can be applied to any monitor. Figure 6 can be used to select a detection threshold, *h* and also to determine if a scheme using this value of *h* would still be acceptable if the standard deviation differed from that assumed. Somewhat surprisingly, decreasing the standard deviation for a given *h* might in some cases actually decrease the probability of detecting a best unacceptable condition in two observations if the threshold *h* was held constant. Such information is important for an automobile manufacturer confronted with an on-board diagnostics compliance test by a governmental regulatory agency. Also by comparing Figure 6 with a similar analysis for different monitors, say an Exponentially Weighted Moving Average (EWMA), a deeper understanding of the comparative behavior of the two monitors could be obtained.

Discussion and Conclusion

The waterfall chart as a perspective plot has shown itself to be a useful tool for communicating the stochastic nature of a monitor. It shows the relationship between the power function and the cumulative run length distribution of a monitor and so can relate run length properties to power and Type I and II error probabilities. Equi-probable contour plots can aid in threshold selection and robustness evaluation.

Lourden (1971) and Beibel (1996) showed that Page's one-sided Cusum has the smallest maximum delay to detection of an abrupt failure for a given average run length for a constant worst acceptable condition among all possible monitors. However, since our models of monitoring situations are never completely accurate, robustness is more important than optimality (see Box and Luceño, 1998, pp. 6-10). Moreover, different optimality criteria may make other monitors preferable to a Cusum. Creative plots, such as those discussed here, can help users compare and evaluate alternative monitors. In general, the waterfall chart has shown itself to be useful for a variety of purposes -- and can in principle be generated for any monitor by Monte Carlo simulation.

Acknowledgements

This article is based on research supported by the Low Emissions Technologies Research and Development Partnership (LEP) of Daimler-Chrysler, Ford, and General Motors. Professor Box's research received additional funding from National Science Foundation Grant DMI-981239.

References

Box, G., Graves, S, Bisgaard, S., Van Gilder, J., Marko, K., James, J. Seifer, M., Poublon, M. and Fodale, F. (2000), "Detecting Malfunctions in Dynamic Systems", *Transactions of the Society of Automotive Engineers: Electronic Engine Controls 2000, Modeling, Neural Networks, OBD, and Sensors (SP-1501)*.

Box, G. E. P., and Luceño, A. (1997) *Statistical Control by Monitoring and Feedback Adjustment* (NY: Wiley)

CARB (1997) *Section 1968.1, Title 13, California Code of Regulations*, California Air Resources Board, "http://www.arb.ca.gov/regact/obdII/obdII.htm".

EC (1998) *Directive 98/69/EC*, European Commission DG III.

EPA (1998) *Fact Sheet EPA420-F-98-045*, Nov. 1998, United States Environmental Protection Agency Office of Mobile Sources, "http://www.epa.gov/orcdizux/regs/im/obd/f98045.htm".

Lourden, B. (1971) “Procedures for Reacting to a Change in Distribution”, *Annals of Mathematical Statistics*, 41, pp. 1897-1908.

Page, E. S. (1954) “Continuous Inspection Schemes”, *Biometrika*, 41, pp. 100-115.

Beibel, M. (1996) “A Note on Ritov’s Bayes Approach to the Minimax Property of the Cusum Procedure”, *Annals of Statistics*, 24, pp. 1804-1812.

Biographical footnote: George Box is Research Director at the Center for Quality and Productivity Improvement at the University of Wisconsin-Madison; he is an Honorary Member of ASQ. Søren Bisgaard (s.bisgaard@monitoring-intl.com) is Eugene M. Isenberg Professor of Technology Management at University of Massachusetts at Amherst, Professor of Industrial Statistics at University of Amsterdam, the Netherlands and a consultant; he is a Fellow of ASQ. Spencer Graves (sgraves@prodsyse.com) is a Principal with Productive Systems Engineering in San José, CA; he is a member of ASQ. Dr. Murat Kulahci (kulahci@kramer.engr.wisc.edu) is an Assistant Professor of Industrial Engineering at Arizona State University, Tempe Arizona; he is a member of ASQ. Dr. Ken Marko (kmarko@ford.com) is a Project Leader in Advanced Diagnostics, Ford Scientific Research, Dearborn, MI. Dr. John James (jjames3@ford.com) is a Senior Research Scientist in Advanced Diagnostics, Ford Scientific Research, Dearborn, MI. John Van Gilder (jvangi01@powertrain.mpg.gm.com) is a Senior OBD Engineer at General Motors Powertrain, MIford, MI; he is a member of ASQ. Dr. Tom Ting (Tom_Ting@gmrnotes3.gmr.com) is a Senior Research Engineer with General Motors Research and Development, Detroit, MI. Hal Zatorski (hz2@daimlerchrysler.com) is OBD Development Manager for Daimler-Chrysler, Pontiac, MI. Cuiping Wu

(CW59@daimlerchrysler.com) is a Quality Engineer with Chrysler Proving Grounds, Chelsea, MI.

Key Words: Run length distribution of a monitor, Power, Cumulative sum chart, Cusum