Statistical Detection of Alarm Conditions in Building Automation Systems

Brian Sallans
Information Technologies
ARC Seibersdorf research GmbH
1220 Vienna, AUSTRIA
brian.sallans@arcs.ac.at

Dietmar Bruckner
Institute of Computer Technology
Vienna University of Technology
1040 Vienna, AUSTRIA
bruckner@ict.tuwien.ac.at

Gerhard Russ
Information Technologies
ARC Seibersdorf research GmbH
1220 Vienna, AUSTRIA
gerhard.russ@arcs.ac.at

Abstract— **A method for the automatic detection of abnormal behavior in a building automation system is compared to a standard system for problem detection. The automated method is based on statistical models of sensor behavior. A model of normal behavior is automatically constructed. Model parameters are optimized using an on-line maximum-likelihood algorithm. Incoming sensor values are then compared to the model, and an alarm is generated when the sensor value has a low probability under the model. The alarms generated by the automated system are compared to alarms generated by pre-defined rules in a standard automation system. The performance, strengths and weaknesses of the automated detection system are discussed.**

I. INTRODUCTION

Automation systems have seen widespread deployment in modern buildings, and include systems for environmental control, energy management, safety, security, access control, and remote monitoring. As the cost of automation systems falls, and the technology converges towards standardized protocols, we can expect automation to move from the office into the home. It will also encompass not just building management technology, but also entertainment, kitchen appliances and communications devices.

Today's building sensor and control systems are primarily based upon the processing of sensor information using predefined rules. The user or operator defines, for example, the range of valid temperatures for a room by a rule – when the temperature value in that room is out of range (e.g. caused by a defect), the system reacts (for example, with an error message). More complicated diagnostics require an experienced operator who can observe and interpret real-time sensor values. However, as systems become larger, are deployed in a wider variety of environments, and are targeted at technically less-sophisticated users, both possibilities (rule-based systems and expert users) become problematic. The control system would require comprehensive prior knowledge of possible operating conditions, ranges of values and error conditions. This knowledge may not be readily available, and will be difficult for an unsophisticated user to input. It is impractical for experienced operators to directly observe large systems, and naive users can not interpret sensor values. The goal of this work is to automatically recognize error conditions specific to a given sensor, actuator or system without the need of pre-programmed error conditions, user-entered parameters, or experienced operators. The system observes sensor and actuator data over time, constructs a model of "normality", and issues error alerts when sensor or actuator values vary from normal. The result is a system that can recognize sensor errors or abnormal sensor or actuator readings, with minimal manual configuration of the system. Further, if sensor readings vary or drift over time, the system can automatically adapt itself to the new "normal" conditions, adjusting its error criteria accordingly.

The system, called BASE (Building Automation system for Safety and Energy efficiency) [1], was tested in a building automation system consisting of 248 sensors spread across four systems (a heating and three ventilation systems). The data was collected over a time period of five months. This paper presents the results of this trial, highlights the strengths and weaknesses of the automated system, and suggests future areas of improvement.

II. BACKGROUND

In this trial the diagnostic system BASE is compared to a standard building automation system. The building automation system consists of a number of sensors and actuators connected by the LonWorks fieldbus (LON) [2]. It offers a visual interface using a Management Information Base (MIB) for retrieving and manipulating system parameters and for the visualization of system malfunctions.

The diagnostic system BASE is based on statistical "generative" models (SGMs). Statistical generative models attempt to reproduce the statistical distribution of observed data. The model can then be used to determine how likely a new data value is or to "generate" new data (i.e. draw samples from the model).[1]

Recent work in generative models has focused on non-Gaussian models (see for example [3], [4]). The diagnostic system BASE uses a number of different SGMs to capture the distribution of sensor data, including histograms, Gaussian mixture models [5], and hidden Markov models [6].

Sensor and control data poses several challenges. The data can be anything from very low level temperature

[1] For probability density models, which are functions of real-valued variables, the probability of the input value is not directly computed from the model. Rather the probability of either exceeding the given value, or of generating a value within a small neighborhood of the given value is computed.

1-4244-9701-0/06/$20.00 ©2006 IEEE.

readings to higher level “fault” detectors or occupancy data from building entry systems. This very different data must be fused into a single system. The volume of data requires fast algorithms [7], [8], and algorithms that can work with on-line data as it arrives [9]. The data values are time-dependent, so a model must (explicitly or implicitly) take time into account [10], [6], [11].

Previous work in applying statistical methods for fault detection includes the use of methods from statistical quality control (SQC) (see [12] and [13], for example), and statistical process monitoring and control [14]. Statistical quality control methods compare sensor values to prepared statistical ”charts”. If the sensor values vary from their expected values over a prolonged period of time, the charts can detect this variation. These methods are appropriate when it is sufficient to reliably detect a small variation after collecting a large number of samples, but are inappropriate for detecting abnormalities in real time. Methods from statistical process monitoring have been applied to a number of systems including chemical processes monitoring [15], monitoring of engines [16], and monitoring of communications networks [17]. Typical statistical methods used include principal components analysis and partial least squares models fit to historical batch [15] or online data [14]. Deviations from normality are then detected using standard statistical tests.

There are several other approaches to fault detection. In classical model-based detection, detailed domain knowledge is used to build a model of the system. Deviations between model predictions and system behavior are flagged as faults (see [18] for a survey). In pattern matching detection, faults are induced in a system, and the resulting sensor values are recorded. A classifier, such as a neural network, is trained using this data set of normal and abnormal behavior to detect failures (see [19] for example). These methods require either a working system for experimentation, or an in-depth knowledge of the system in question, both of which are lacking for large building automation systems.

Despite their success in other domains, SGMs have not been applied to error detection for building automation sensor and control data. There are two reasons for this. First, it has only recently become possible (and economical) to collect a wide range of cross-vendor sensor data at a central location. Second, most algorithms to optimize the parameters of SGMs are quite compute-intensive. Many algorithms have been of only theoretical interest, or are restricted to small toy problems. Only recently have powerful approximation algorithms and powerful computers become available that can handle large quantities of data in real-time. To our knowledge, this paper is the first case study of the use of statistical process monitoring in a building automation system.

III. THE DIAGNOSTIC SYSTEM

The goal of the diagnostic system BASE is to automatically detect sensor errors in a running automation system. It does this by learning about the behavior of the automation system by observing data flowing through the system. The diagnostic system builds a model of the sensor data in the underlying automation system, based on the data flow. From the optimized model, the diagnostic system can identify abnormal sensor and actuator values. The diagnostic system can either analyze historical data, or directly access live data.

We use a set of statistical generative models to represent knowledge about the automation system. A statistical generative model takes as input a sensor value, status indicator, time of day, etc., and returns a probability between zero and one.

Using SGMs has several advantages. First, because the model encodes the probability of a sensor value occurring, it provides a quantitative measure of “normality”, which can be monitored to detect abnormal events. Second, the model can be queried as to what the “normal” state of the system would be, given an arbitrary subset of sensor readings. In other words, the model can “fill in” or predict sensor values, which can help to identify the source of abnormal system behavior. Third, the model can be continuously updated to adapt to sensor drift.

A. Error Detection

Given an SGM, implementation of this functionality is straight-forward. The system assigns to each newly-observed data value a probability. When this probability is high, the system returns that the new data value is a “normal” value. When the probability falls below a specific threshold, the system rates the value as “abnormal”. The SGM system generates alarm events when it observes abnormal sensor values. This leaves open the question of how to assign the threshold for normality. In practice, the user sets the threshold using a graphical interface. Initially, before the system has learned normal system behavior, many alarms are generated, and the user may decide to set the threshold to a value near zero. As the system acquires a better model of the sensor system, the threshold can be raised. In any case, the threshold parameter tells us how improbable an event should be to raise an alarm. The system can also use a log-probability scale, so that the threshold can easily be set to only register extremely unlikely events.

B. Statistical Generative Models

The BASE system implements a number of SGMs (see Table I).

TABLE I
STATISTICAL GENERATIVE MODELS

Model	Variable Type	Parameters
Gaussian	Real	μ,σ^2
Histogram	Discrete, Real	Bin counts
Mixture of Gaussians	Real	μ_i,σ_i^2,π_i
Hidden Markov model	Real	T_{ij},μ_i,σ_i^2
Hidden Markov model	Discrete	T_{ij},Bin counts

The more complex models add additional capabilities, or relax assumptions in comparison to a simple Gaussian model.

The diagnostic system uses SGMs of automation data points. For any given data value x, model M assigns a probability to x: $P_M(x) \rightarrow [0,1]$.

Note that, for discrete distributions such as a histogram, the value assigned to x by the model $P_M(x)$ is a well-defined probability, since the set of possible assignments to x is finite. For a probability density, such as a Gaussian or mixture of Gaussians, the probability value assigned to x by the model is the probability *density* at that value. In order to convert this density to a probability, the probability of generating a value within a neighborhood $\pm\delta$ around x is computed as $\int_{-\delta}^{\delta} P_M(x+\phi)d\phi$, and approximated as $2\delta P_M(x)$ for small δ. Alternatively the probability under the model of equaling or exceeding the observed value can be computed: $P_M(x' \geq x) = \int_{\phi}^{\phi+\infty} P_M(x+\phi)d\phi$.

The data x can be a sensor reading such as an air pressure sensor, contact sensor, temperature sensor, and so on. Given a new data value, the system assigns a probability to this value. When the probability is above a given threshold, the system concludes that this data value is “normal”.

Given a sequence of sensor readings $\mathbf{x} = \{x_1, ..., x_T\}$ from times 1 to T, the system must create a model of “normal” sensor readings. The system uses an on-line version of the expectation maximization algorithm for maximum-likelihood parameter estimation. Given a model M with parameters θ, the log-likelihood of the model parameters given the data $\mathbf{x}$ is given by:

$$\mathcal{L}(\theta) = \log P_M(\mathbf{x}|\theta) \tag{1}$$

where the notation $P(\mathbf{x}|\theta)$ denotes the conditional probability of $\mathbf{x}$ given the current values of the parameters θ.

The maximum-likelihood parameters are defined as the parameter values which maximize the log-likelihood over the observed data:

$$\theta_{ML} = \underset{\theta}{\operatorname{argmax}} \{\log P_M(\mathbf{x}|\theta)\}$$

C. On-line Parameter Updates

In order for the system to continually adapt the model parameters, the parameter update algorithm must incrementally change the parameters based on newly observed sensor values. Such “on-line” updates have the advantage that there is no time during which the system is in an “optimization” phase, and unavailable for diagnostics.

For the tests described in this paper, mixture of Gaussians models were used. For the mixture of Gaussians model, the BASE system uses a simple stochastic estimation method, based on an expectation-maximization algorithm. As each new data value x_i is observed, the parameters are adjusted in a two-step process. First, the posterior probability of each element of the mixture given the data value is computed. Second, the parameters are adjusted so as to increase the expected joint log-probability of the data and the Gaussian mixture component. See section 2.6 of [5] for details.

IV. RESULTS

The diagnostics system was used to analyze sensor and actuator values from a building automation system consisting of a heating system and three ventilation systems. The BASE diagnostic system monitored the values of 248 sensors and actuators. These included forced air temperatures, room temperatures, air pressures, the status of control valves, and so on. The data was collected over a period of five months (mid December to mid May), thus including the seasonal transition from winter to summer.

The BASE system was allowed to adapt models to each of the 248 sensors. Each sensor model consisted of 12 mixtures of Gaussians models, one model for every two hours of a 24 hour day. During this time period, the alarm messages from the standard building automation system were also recorded. Because of the relatively simple nature of the individual models, we were able to simultaneously fit a large number of models in real time. In our case, in any particular 2 hour period, 248 sensor models were being fit simultaneously, and the system optimized $248 \times 12 = 2976$ sensor models in total.

In the following sections we describe and show examples of model parameter optimization, optimized models, and compare alarms delivered by BASE and by the traditional building automation system.

A. Parameter Optimization

The model log-likelihood, given by Eq.(1), is a measure of model quality. As the parameter values are optimized on-line, the log-likelihood of the sensor models increases. Fig. 1 shows the average log-likelihood during parameter optimization, on average, for the sensor models in the test system. The log-likelihood increases with time, indicating that the models improve over time. The log-likelihood does not consistently increase, however, due to the on-line fitting of parameters simultaneously with reporting of abnormal sensor values. If sensors receive abnormal values, the log-likelihood decreases, until the values return to their normal range or the sensor model adapts to the new range of values.

Fig. 2 shows an example from a single sensor. The upper figure shows the sensor value, and the lower shows the corresponding log-likelihood as a function of time. The large disturbance in the center (a power fluctuation) registers an alarm. So do the two small ”spikes” near the end of the graph.

B. Comparison to Standard System

During the test period, each sensor model’s log-likelihood was computed for each new sensor value. This log-likelihood was compared to a threshold. If the log-liklihood fell under threshold, an alarm was emitted. These alarms were then analyzed to discover if they corresponded to alarms emitted by the standard system.

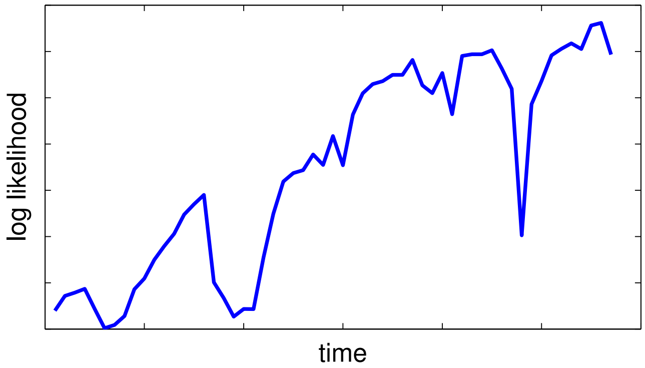

Fig. 1. Average learning curve for sensor models. On average the log-likelihood for the model improves over time. The three large drops in average log-likelihood correspond to large modifications of the system (addition of equipment, changes in system parameters). Smaller drops correspond to system-wide disturbances (such as a power outage).

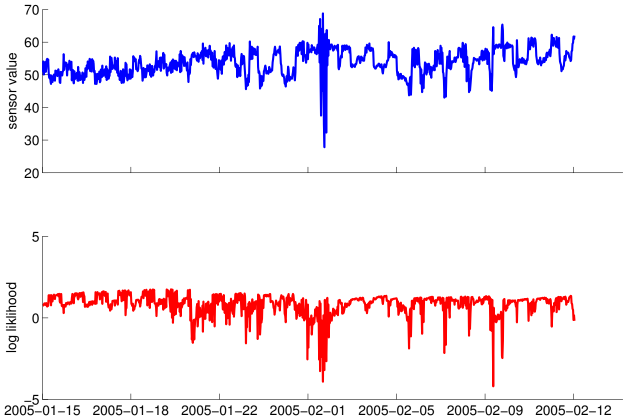

Fig. 2. Sensor value and log-likelihood for a single sensor from the system. Unusual sensor values register as drops in the log-likelihood, causing alarms.

Over the 5 month test period, the standard system emitted 2521 Alarms (excluding the lowest priority informational messages). The number of alarms delivered by BASE depends on the alarm threshold selected by the user. The number of BASE alarms ranged from 198 at the lowest alarm threshold to 1599 at the highest tested alarm threshold.

The BASE system, or any automatic system for alarm detection, can deliver "false" alarms, that is, alarms that are considered unimportant by the operator. This can occur if the sensor value makes a statistically significant but in reality unimportant deviation from normality, or if the sensor model does not sufficiently capture the true variation in sensor values. With the standard system, alarms are by definition not "false", because they occur for exactly defined situations. However, alarms from standard systems can also be "false" in the sense that they are considered unimportant by the operator. In order to compare the BASE and standard alarms, an equivalent to "false" alarms must be defined for the standard building automation system. In this paper, an alarm from the standard system is considered unimportant if it is cancelled by the operator within five minutes of its occurrence. We call these alarms "quick" alarms. We also label BASE alarms considered unimportant by the operator "quick" alarms.

Of the BASE alarms, a subset of approximately 10% of alarms was classified by hand into "quick" or "normal" alarms. In order to compare to the standard alarms, a subset of approximately 10% of the standard alarms were also classified into "quick" and "normal", using timing information from the alarm log. In both cases, the subsampled alarms were uniformly randomly selected, and it was assumed that the subsample of classified alarms were representative of the set of all alarms.

Fig. 3 shows an analysis of alarms emitted by both systems. The bar graph shows clusters of bars, one cluster for each of 9 different alarm thresholds. The error bars indicate confidence values, taking the subsampling of hand-analyzed alarms into account.

As the threshold increases, the number of alarms delivered by BASE increases. After a threshold value of -4.2, the number of BASE alarms does not increase, indicating that the data from the automation system is either classified by BASE as quite unlikely (probability less than 0.05) or likely (probability greater than 0.15) with little in between. This suggests that the BASE models describe the observed data well. Also, the number of BASE alarms which correspond to alarms from the standard system, the number of unique good BASE alarms, and the number of BASE false alarms all increase as threshold increases.

In general, the standard system delivers a surprising number of alarms that are quickly dismissed, much more so than by BASE. This may simply be because of greater familiarity with the standard system, so the operator can quickly decide between alarms that they often see and new, problematic situations. The number of BASE false alarms is low in comparison, indicating that the BASE system does deliver useful information. The large but not complete overlap between BASE and standard alarms suggest that true alarms can be sorted into three categories: Alarms where a value crosses a pre-defined threshold, without deviating from historical values (for example, an alarm indicating that a fuel tank is nearly empty); alarms that deviate from normality but do not cross threshold (see the alarms in Fig. 2 or Fig. 4 for example); and alarms that do both. The BASE and standard systems largely complement one another and give the operator additional information which they would not otherwise have.

C. User Comfort

One case in which BASE can deliver useful information is to enhance user comfort. In the test environment, users receive heated air from the central heating system, and are then able to increase air temperature with an additional local system. It is therefore difficult to set global thresholds indicating a local heating problem. The BASE system detects local problems quickly, since each

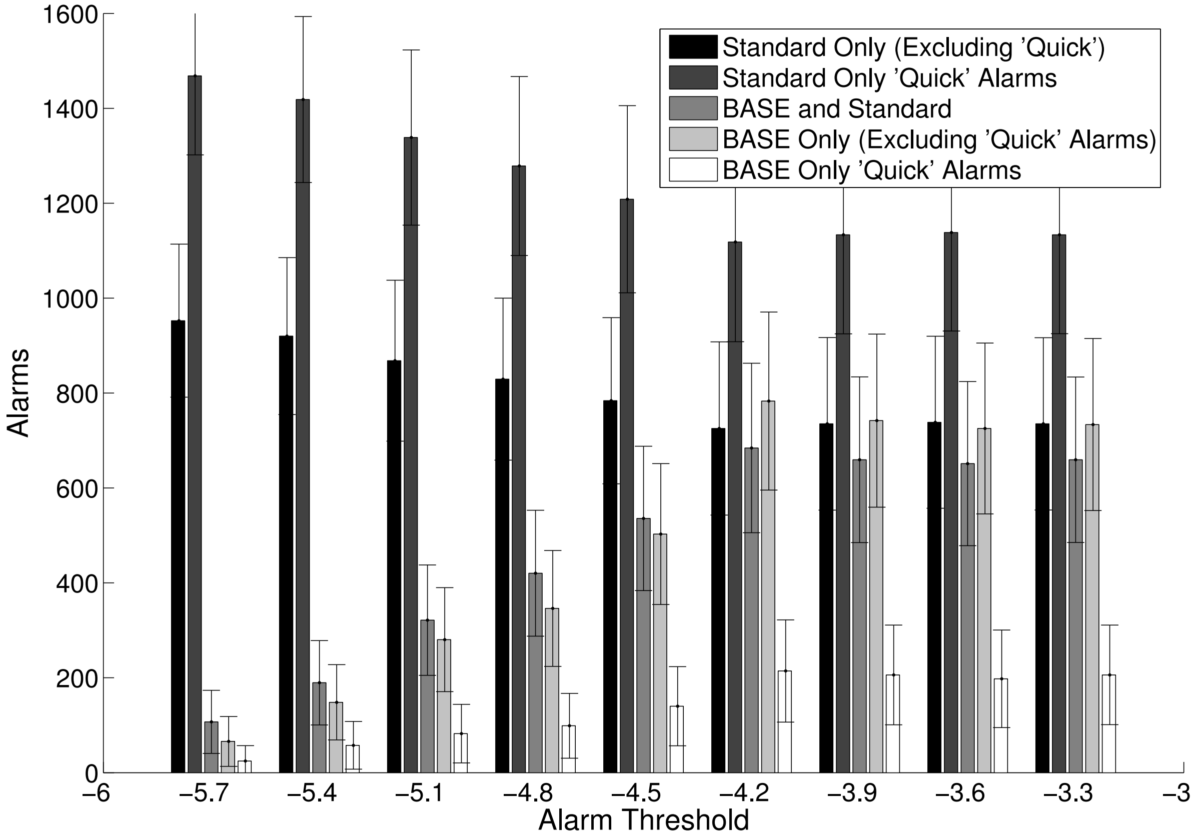

Fig. 3. Analysis of BASE alarms and alarms from the Standard building automation system. Both systems deliver alarms that are non-critical (those labeled "Quick" alarms) as well as alarms that are considered important by the user. As the threshold increases, the number of good alarms, but also the number of false alarms delivered by BASE increases.

sensor model is adapted to the local environment. Fig. 4 shows an example of local problem detection. The upper figure shows outside temperature, and the lower figure shows local room temperature. The dark stripes indicate normal working hours, and the number indicates average temperature over the 12 hour period. On the third day the indoor temperature drops to an abnormal (and uncomfortable) 18C, triggering a BASE alarm (vertical line). Without this alarm, the occupants of the room must alert the building managers themselves.

D. System Adaptation

The BASE system adapts model parameters to changes in the environment. That is, when there are repeated examples of an abnormal situation, BASE will adapt the sensor models such that the new situation is no longer considered abnormal. Fig. 5 shows an example of this behavior. When the temperature in a room is abnormally high (on date 11.02), this is detected by the system, and an alarm is delivered (vertical line). However, the next time that this temperature is reached (on 14.02), the system has already adapted to this circumstance, and does not deliver an alarm. How many repetitions of a situation are required before an alarm is no longer generated depends on the system learning rate, and how unusual the situation was. More unusual situations require more repetitions before the system adapts.

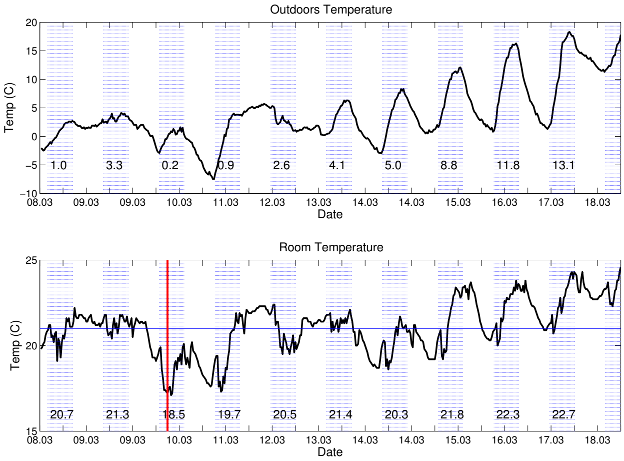

Fig. 4. An example of detecting a change in local conditions. On the third day (third dark stripe) average indoor temperature drops, triggering an alarm.

V. Conclusion

This paper describes a test of statistical methods for the automatic detection of abnormal sensor values. The BASE system can automatically build a model of normal sensor behavior. It does this by optimizing model parameters using an on-line maximum-likelihood algorithm. Incoming sensor values are then compared to the model,

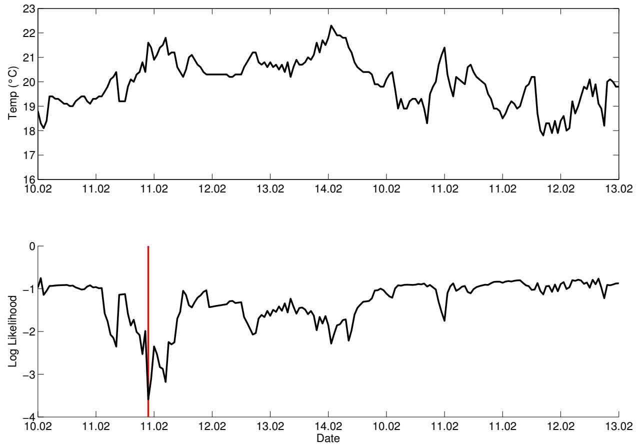

Fig. 5. An example of system adaptation. On 11.02 the temperature is abnormally high, and an alarm is generated. On 14.02 a similar temperature is reached, and no alarm is generated.

and an alarm is generated when the sensor value has a low probability under the model. The model parameters are continuously adapted on-line.

We found that the traditional system and the BASE system complement one another. While the traditional system was able to detect statistically insignificant but important deviations using thresholds, BASE was able to detect deviations that were within the thresholds used by the traditional system, but were nevertheless significant deviations from normality. These alarms have implications for system reliability, safety, efficiency and user comfort.

VI. REFERENCES

[1] B. Sallans, D. Bruckner, and G. Russ, "Statistical model-based sensor diagnostics for automation systems," in *Proceedings of the 6th IFAC International Conference on Fieldbus Systems and their Applications*, 2005, pp. 239–246.

[2] D. Loy, D. Dietrich, and H. Scheinzer, Eds., *Open Control Networks: LonWorks/EIA 709 Technology*. Boston, Dordrecht, London: Kluwer Academic Publishers Boston, 2001.

[3] G. E. Hinton, B. Sallans, and Z. Ghahramani, "A hierarchical community of experts," in *Learning in Graphical Models*, M. I. Jordan, Ed. Kluwer Academic Publishers, 1998, pp. 479–494.

[4] T.-W. Lee, M. Girolami, and T. Sejnowski, "Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources," *Neural Computation*, vol. 11, no. 2, pp. 417–441, 1999.

[5] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York NY.: Oxford University Press Inc., 1995.

[6] L. R. Rabiner and B.-H. Juang, "An introduction to hidden Markov models," *IEEE ASSAP Magazine*, vol. 3, pp. 4–16, January 1986.

[7] T. S. Jaakkola, *Variational Methods for Inference and Estimation in Graphical Models*. Cambridge, MA: Department of Brain and Cognitive Sciences, MIT, 1997, Ph.D. thesis.

[8] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, p. 183:233, 1999.

[9] R. M. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. I. Jordan, Ed. Kluwer Academic Publishers, 1998, pp. 355–368.

[10] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, Series D, Journal of Basis Engineering*, vol. 82, pp. 35–45, March 1960.

[11] B. Sallans, "Learning factored representations for partially observable Markov decision processes," in *Advances in Neural Information Processing Systems*, S. A. Solla, T. K. Leen, and K.-R. Müller, Eds., vol. 12. The MIT Press, Cambridge, 2000, pp. 1050–1056.

[12] P. Fasolo and D. E. Seborg, "An SQC approach to monitoring and fault detection in HVAC control systems," in *Proceedings of the American Control Conference*, Baltimore, Maryland, 1994.

[13] J. Schein and J. House, "Application of control charts for detecting faults in variable-air-volume boxes," in *ASHRAE Transactions*, vol. 109, no. 2, 2003, pp. 671–682.

[14] P. Goulding, B. Lennox, D. Sandoz, K. Smith, and O. Marjanovic, "Fault detection in continuous processes using multivariate statistical methods," *International Journal of Systems Science*, vol. 31, no. 11, pp. 1459–1471, 2000.

[15] B. Wise and N. Gallagher, "The process chemometrics approach to chemical process fault detection and supervision," *Journal of Process Control*, vol. 6, no. 6, pp. 329–348, 1996.

[16] N. Pontoppidan and J. Larsen, "Unsupervised condition change detection in large diesel engines," in *IEEE Workshop on Neural Networks for Signal Processing*, C. Molina, T. Adali, J. Larsen, M. V. Hulle, S. Douoglas, and J. Rouat, Eds. Piscataway, New Jersey: IEEE Press, 2003, pp. 565–574.

[17] C. S. Hood and C. Ji, "Proactive network fault detection," in *INFOCOM (3)*, 1997, pp. 1147–1155.

[18] R. Isermann, "Process fault detection based on modeling and estimation methods – a survey," *Automatica*, vol. 20, no. 4, pp. 387–404, 1984.

[19] J. House, W. Y. Lee, and D. R. Shin, "Classification techniques for fault detection and diagnosis of an air-handling unit," *ASHRAE Transactions*, vol. 105, no. 1, pp. 1987–1997, 1999.