

Automated Diagnosis for Facility HVAC Systems

Rajaram Ganeshan & Lewis Johnson

USC Information Sciences Institute
4676 Admiralty Way Suite 1001
Marina Del Rey, California 90292
{rajaram,johnson}@isi.edu

Roger Henderson

Facilities Management, University of Southern California

From: AAAI Technical Report SS-99-04. Compilation copyright © 1999, AAAI (www.aaai.org). All rights reserved.

Abstract

Energy Management and Control Systems (EMCS) are used to manage Heating, Ventilation, Air Conditioning (HVAC), sometimes also lighting and security systems in single or multiple facilities. Although EMCS control algorithms are designed to be energy efficient, inefficiencies can result from: (1) equipment or sensor or controller malfunction, (2) operator error (building operators can override control settings without understanding the implications), and, (3) facility use inconsistent with design assumptions (e.g., the maximum design cooling load is exceeded in everyday use). Such problems can go unnoticed until they result in a significant deterioration in performance resulting in either an alarm condition or complaints from occupants. Identifying the problems well in advance of the alarm or complaint stage can result in substantial savings and occupant satisfaction. This paper proposes a model-based approach to detect and diagnose these problems.

Introduction

Energy Management and Control Systems (EMCS) are used to manage Heating, Ventilation, Air Conditioning (HVAC), sometimes also lighting and security systems in single or multiple facilities. With direct digital controls (DDC), the EMCS continuously monitors and controls almost all aspects of HVAC components and systems. Although EMCS control algorithms are designed to be energy efficient, inefficiencies can result from: (1) equipment or sensor or controller malfunction, (2) operator error (building operators can override control settings without understanding the implications), and, (3) facility use inconsistent with design assumptions (e.g., the design cooling load is exceeded in everyday use; or, a computer monitor is placed close to the room thermostat thereby throwing it off). Such problems can go unnoticed until they result in a significant deterioration in performance resulting in either an alarm condition or complaints from occupants. Identifying the problems well in advance of the alarm or complaint stage can result in substantial savings

and occupant satisfaction.

Equipment malfunction. Diagnosing equipment malfunction has been extensively researched (Hamscher, Console & De Kleer, 1993). Approaches to automated diagnosis can be classified into two general categories: (1) pattern-matching based on predetermined set of faults (Lee et al. 1997), and, (2) model-based approach (De Kleer & Williams, 1987). In the pattern-matching approach, current symptoms are matched against a database of expected symptom values (or fault signatures) for various kinds of faults. A match determines the fault causing the problem. In the model-based approach, models of correct and/or faulty behavior are used to identify faulty components based on discrepancies between observed and predicted behaviors. The model-based approach is attractive because unlike the pattern-matching approach, all of the symptoms-faults do not have to be explicitly catalogued.

Operator error. Dealing with operator error requires reasoning about efficient behavior. For example, suppose that the economizer control settings in an air-handling unit have been overridden by the operator and the economizer dampers are fully open (100%). When the outside temperature changes such that the economical damper setting is less than 100%, this situation is not recognized as a fault because the system behavior is consistent with the settings. A general solution for this problem that is consistent with the model-based framework, is to add an objective function (e.g. minimize cost) to the model which will allow it to predict a lower cost solution when compared with the observed solution.

Facility usage inconsistent with design assumptions. Facility usage in a manner inconsistent with design assumptions is quite common. For example, when computers are introduced into a space that was not originally designed for that purpose, it can result in a large cooling load which the air-handling unit was not designed to handle. In such a situation the cooling valve even at its highest setting cannot bring the actual supply temperature close to the desired setpoint. This is illustrated by sensor measurements for a 24-hr period obtained from an air-handling unit on one of the buildings on the USC campus (Figures 1 & 2). Notice that the variation of chilled water valve position over the 10 am-9pm period (Figure 1 - stays

at 100%) and the difference between setpoint and actual supply temperatures which seem to diverge over the same period (Figure 2).

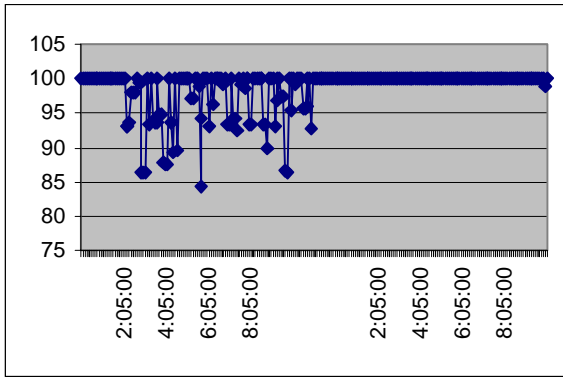


Figure 1 Chilled water valve setting wrt time

If not modeled carefully, such an over-capacity situation will be incorrectly interpreted as a fault either in the cooling valve controller and/or the cooling coil.

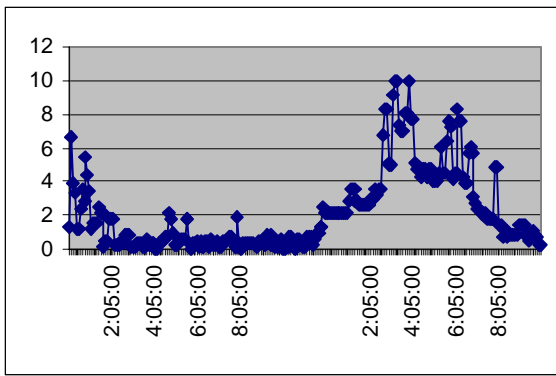


Figure 2 Difference between setpoint and supply temperatures wrt time

There are two approaches to dealing with this problem: (1) assumptions about the maximum cooling load are made part of the model - then such assumptions can be identified as one of the reasons for discrepancies between observed and predicted behavior; or, (2) structure the state-space into an abstract situation-based model of behavior like in the PROBES system (Marsella & Johnson, 97). Each situation represents a mode of behavior (steady-state, overcapacity, etc.) and determines the model that applies. The latter approach might allow for simpler models in a given situation that require less input or can even work with missing or incorrect inputs.

Modeling HVAC Components and Behavior

The typical components in a HVAC system include (McQuiston & Parker, 94): air handlers, fans, heating

sources, refrigeration (or chillers), pumps, piping, and controls. Figure 3 shows an abstract model of a variable-air-volume HVAC system (chillers & associated piping

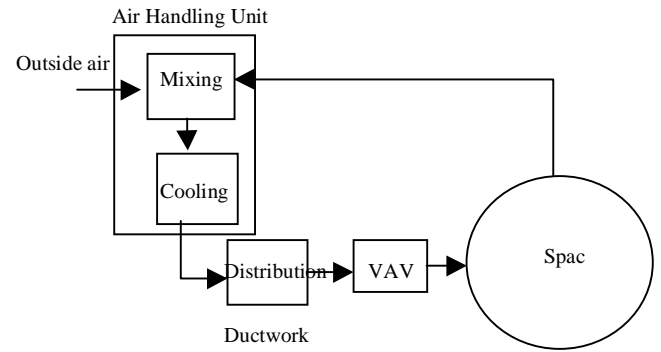


Figure 3 Abstract model of a VAV HVAC system

and pumps are excluded). In a cooling mode, cool air is supplied to the space which absorbs the heat from the space and is then re-circulated, mixed with outside air and cooled and returned to the space. The mixing and cooling processes occur in the air-handling unit (AHU). Insulated ductwork supplies the cooled air from AHU to the space and returns circulated air from the space to the AHU. The variable air volume (VAV) boxes supply air to individual zones within the space. In this paper, we focus on the air-handling unit and show how it is modeled for the diagnosis process. Figure 4 shows the air-handling unit which includes the return fan, dampers for exhaust, recirculated and outside air, air filter, cooling coil, supply fan and the sensors (for return, outside and mix air temperatures, and static pressures). The static pressure sensor SP1 is located a little downstream (dotted line in Figure 4) of the supply fan.

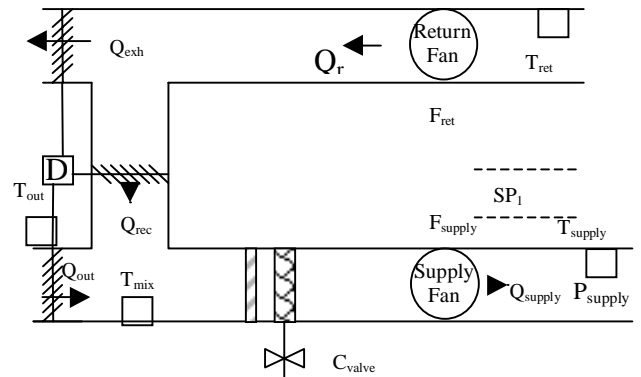


Figure 4 Air-handling unit

The two key processes in the AHU are mixing and cooling. The mixing process mixes return air and outside air in the AHU. The mix proportion is determined by how much the outside air-dampers are open which is in turn controlled by the difference between return air and outside air

temperatures. There is a minimum ventilation limit (usually 10%) which implies the outside air dampers can never be completely closed. The mixing is assumed to happen under adiabatic, steady flow conditions (Figure 5).

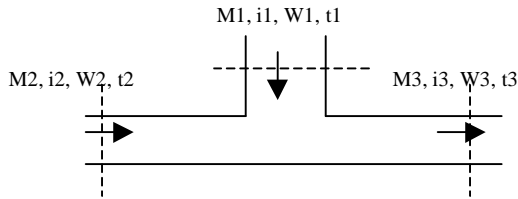


Figure 5 Mixing of 2 air streams

In these discussions, the parameters that describe the air stream include: mass flow rate (m), enthalpy (i), humidity ratio (w) and temperature (t). Based on energy and mass balance (both mass of dry air and water vapor), the following relationship can be derived:

$$I_3 = ((m_1/m_2) \cdot i_1 + i_2) / (1 + m_1/m_2)$$

$$W_3 = ((m_1/m_2) \cdot W_1 + W_2) / (1 + m_1/m_2)$$

Assuming dry air and constant specific heat the temperature of the mixture:

$$t_3 = (m_1/(m_1+m_2)) \cdot t_1 + t_2 \cdot (m_2/(m_1+m_2)) \dots (1)$$

The cooling process in the AHU also results in dehumidification of air because it loses some water vapor.

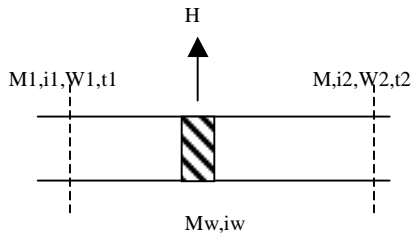


Figure 6 Cooling of an air stream

This results in both sensible heat loss as well as latent heat loss (due to dehumidification). Based on energy balance and mass balance considerations, we can derive the following:

$$H = m(i_1 - i_2) - m(W_1 - W_2)iw \dots (3)$$

where H is the rate of heat transfer

The rate of heat transfer is a function of the cooling valve setting which is based on the difference between the setpoint and actual the supply air temperatures. The setpoint supply air temperature is determined based on zone temperatures in the spaces being cooled.

The supply fan speed is based on difference between the static pressure setpoint and actual static pressure differences. The static pressure setpoint (desired) is determined by the VAV damper settings (e.g., if any damper is below 20% then static pressure setting 0.8; if any damper is above 90% then static pressure is 1.1.) The return fan speed is usually set to lag behind the supply fan so that more air is pumped into the spaces than is taken out to avoid creating a vacuum situation in the space.

There are four controllers being used in the AHU: (1) damper control; (2) cooling valve control; (3) supply fan control; and, (4) return fan control. In general controllers are one of two types:

PI (proportional integral) controller output =
 $\text{error} \cdot P_CONSTANT + I_CONSTANT \cdot (\text{integratedError})$

PID (proportional integral derivative) controller output =
 $\text{error} \cdot P_CONSTANT + I_CONSTANT \cdot (\text{integratedError}) + D_CONSTANT \cdot (\text{rate of change of measured var})$

where $P_CONSTANT$, $I_CONSTANT$ and $D_CONSTANT$ are different constants.

For example, consider the cooling valve controller. The output variable is the cooling valve setting control signal; error is the difference between desired supply temperature and actual supply air temperature. The control is activated only when the error is greater than some constant threshold.

The temporal abstractions used in XDE (Hamscher, 1992) – e.g. frequency, counting, sequence, duration, change, etc. can be used to represent temporal behavior. XDE also uses two time related predicates: thru and tsame. (Thru ?l ?u ?signal ?value) means that ?signal has value ?value from time ?u to time ?l. Using such abstractions, the proportional relationship between the controller output and error may be expressed as (assuming the output variable and error have a +ve monotonic relationship):

If $\text{changedBy}(t_1, t_2, \text{ERROR}) > \text{THRESHOLD}$ then

$\text{changedBy}(t_1, t_2, \text{OUTPUT_VAR}) / \text{changedBy}(t_1, t_2, \text{ERROR}) > 0$

else

$\text{changedBy}(t_1, t_2, \text{OUTPUT_VAR}) == 0$

where the $\text{changedBy}(t_1, t_2, \text{var})$ returns the amount of change in the variable over the time period. Note that this constraint description does not capture some aspects of the controller behavior, e.g., magnitude of the control action and how long should it take for the error to be damped by control action. We are still investigating the kinds of temporal abstractions that will be useful in this domain.

Constraint-based models of behavior

The thermodynamic processes and control loops that describe the behavior of AHU (and other components like VAV boxes) enforce certain relationships between variables. For example, the mixing of the return air stream with outside air to produce the mix air stream establishes certain relationships between the temperatures based on eqn. (1). We have built a constraint propagation system hooked up with an ATMS (Forbus & DeKleer) in Java. Variables are described by "Cell" objects. Cells participate in specific roles in constraints. Most of the functionality for constraint propagation is defined in the base Constraint

class. New constraint types are defined as Java classes that inherit from the basic constraint class. The new constraint classes have to define their roles and rules that determine values for the variables they constrain. A constraint network is a collection of constraints that share variables. Figure 7 shows a portion of the constraint network that models the behavior of the AHU.

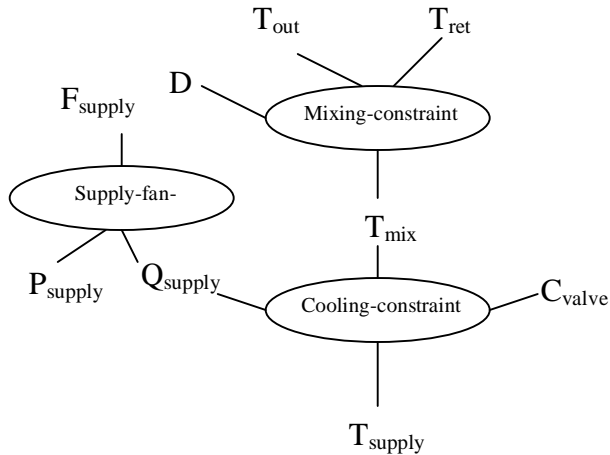


Figure 7 Portion of constraint network for AHU

Constraints depend on assumptions about components working properly. For example, the mixing constraint assumes that the dampers are functioning properly. Although not shown in Figure 7 sensors are also modeled as equality constraints between actual and measured values.

Detecting Conflicts and Identifying Diagnoses

As in GDE (De Kleer & Williams) the diagnosis process uses the ATMS. Whenever the value of a variable is changed constraint propagation with ATMS is used to identify conflicts which are no-good environments in ATMS. A nogood is a set of assumptions that cannot all simultaneously hold. At least one of the assumptions must be false. Generally, these assumptions correspond to the assertion that a particular component is working properly. Thus, diagnoses are the minimal set of assumptions that intersects every conflict. Another way of looking at the diagnoses is by computing the maximal sets of assumptions that do not intersect any conflict. A maximally consistent set of assumptions is a set of assumptions to which no other assumption can be added without rendering it inconsistent (or nogood). The ATMS interpretations algorithm (Forbus & DeKleer) identifies the maximally consistent sets of assumptions (i.e., do not include any nogoods). By making further observations and generating more conflicts, the set of diagnoses candidates can be narrowed till a single one is identified. There are two important questions to address in the above process:

- How to select the next observation to make ?

- How to update the probabilities of candidates based on the measurement?

In selecting the next candidate, GDE uses an entropy measure. XDE suggests a probe and uses combination of refinement (use a fault model) and decomposition (descend into hierarchy) to advance the diagnosis process. In the HVAC diagnosis situation, it is not possible to add new sensors since the sensor points are fixed. However, it maybe possible to do tests or localized experiments to eliminate certain candidates (e.g., the cooling valve could be closed all the way to determine if it is working - this would mean that the supply temperature is rising and you would observe a greater supply temperature than before). These kinds of tests will not be possible in settings like operating rooms or special laboratory rooms where maintaining a certain temperature range is critical. The selection of tests is guided by the probability of failure of the candidate. Once a component is ruled out in this way, its probability is redistributed among the remaining candidates. A scheme similar to GDE can be used to update probabilities after a test.

Real-time monitoring

EMCS sensor measurements can be reported at periodic intervals or whenever the sensor measurement changes by a given threshold. Since there can be a very large number of sensor points in a typical building (around 1000), requesting frequent updates can place a heavy computational demands on the EMCS server. As mentioned earlier, an abstract situation-based model of behavior based on limited sensor input can be used to guide the diagnosis process. For example, steady-state conditions can be detected by monitoring certain parameter values over a short period and time and comparing their rate of change with the expected value under normal steady-state conditions.

Related Work

Diagnosis of faults in dynamic systems has been independently pursued both in the engineering and AI communities. The research work in these communities is rarely (if ever) cross-referenced.

HVAC System Diagnosis

There have been a few efforts that have directly addressed fault detection HVAC systems. We focus on two recent projects: (1) Fault detection project in NIST's BFRL (Building Fire Research Laboratory), and, (2) Whole building Diagnostician project at Pacific Northwest Laboratories.

The BFRL project used fault signature approach to identify complete failures in an Air Handling Unit. The faults signatures were based either on (1) Differences between actual measured values of parameters and values predicted based on nominal operating conditions (determined by the constant load assumption); or, (2)

changes in un-measurable parameters in a model of controlled system with recursive parameter estimation using Kalman filters (Lee et al, 1996b). Although, the latter approach requires fewer sensors but it needs more computational resources. In a subsequent paper (Lee et al, 1996b), they use idealized steady-state patterns of the residuals to define each fault and use a neural network to learn the association between the patterns and the faults. To avoid having to retrain the entire neural network for new faults, they devised a 2-stage architecture with separate ANNs for each sub-system (Lee et al, 1997). The disadvantages of this approach include:

- The fault signatures were developed under specific assumptions (e.g. constant load) - if these assumptions are violated then the signatures are no longer valid!
- Only complete failures are considered - faults introduced by degradation in performance are not considered.
- The advantages of using a ANN for fault recognition are not clear especially since it is computationally expensive to train and use.

The Whole Building Diagnostician includes two modules - one for analysing the energy performance of the entire building and the other for monitoring the performance of air handling units for detecting 20 typical problems with outside-air control.

Qualitative Reasoning

Most of the AI approaches to diagnosis of dynamic systems (Forbus, 86; Dvorak, 92; Subramanian, 95) involve searching over a qualitative state-space of the system behavior. The qualitative state space is generated by both normal and fault models using a qualitative reasoning approach like QSIM (Kuipers, 94). The disadvantages of qualitative reasoning approaches are:

- Coverage of behaviors by qualitative models - qualitative models can be very approximate so that they miss interesting behaviors that would be manifested in quantitative models. This is especially true if we are dealing with gradual performance degradation (vs. total failures) which can manifest itself in subtle ways.
- Intractability of the search over the qualitative state space - is usually very computationally expensive as has been noted both in MIMIC and QDOCS, thus limiting the applicability of such approaches in the real world.

Conclusions and Future Work

Our preliminary work has shown the feasibility of the model-based approach to diagnosing HVAC system problems. We need to develop more comprehensive models that include all aspects of the HVAC system and test to verify that these models can cover the range of typical problems: equipment faults including complete failures and gradual performance degradation, operator

error and design assumption violations. Since fault data may not always be available in real-world settings, we are looking at simulations (e.g. HVACSIM+) to generate test data for faults. We plan to do more testing with both real-time and simulated data and hope to be able to report the results at the symposium.

The ultimate goal is to automate the entire energy management process for a facility (single building or a collection of buildings like the USC campus). Diagnosing faults in HVAC systems is only one aspect of this process. Many factors should be considered, including the physical properties of the building, occupancy changes over time, variance of electricity rates over time, and, natural energy gains/losses (Benard et al., 1992). Eventually we hope to have a collection of intelligent energy management agents that can work together to monitor, analyse and control various aspects of facilities to optimize energy usage.

References

- ASHRAE. 1981 *Fundamentals*. ASHRAE.
- De Kleer & Williams. 1987. Diagnosing Multiple Faults. *Artificial Intelligence*, 32 pg 97-130.
- Dvorak, D. 1992. Monitoring and Diagnosis of Continuous Dynamic Systems Using Semiquantitative Simulation. Ph.D Diss, Dept. of Computer Science, University of Texas, Austin.
- Falkenhainer, B, Farquhar, A., Bobrow, D., Fikes, R., Forbus, K., Gruber, T., Iwasaki, Y., and Kuipers, B. 1994. "Compositional Modeling Language," KSL-94-16. Dept. of Computer Science, Stanford Univ.
- Forbus. K. 1986. Interpreting measurements of physical systems, *Proceedings of AAAI-86*. AAAI Press.
- Forbus & DeKleer. 1993. *Building Problem Solvers*. MIT Press.
- Hamscher, W. C. Modeling Digital Circuits for Troubleshooting," in *Readings in Model-based Diagnosis*. Morgan Kaufman Publishers.
- Hamscher, W., Console, L., and, de Kleer, J. 1993. *Readings in Model-based Diagnosis*, Morgan Kaufman Publishers.
- HVACSIM+. Building Fire Research Laboratory at NIST.
- Kuipers, B. 1994. *Qualitative Reasoning*, MIT Press.
- Lee, W. Y.; Park, C.; House, J. M.; Kelly, G. E. 1996a. Diagnosis of an Air-Handling Unit Using Artificial Neural Networks. *ASHRAE Transactions*, Vol. 102, No. 1, 540-549, 1996.

Lee, W. Y.; Park, C.; Kelly, G. E. Fault. 1996b. Fault Detection in an Air-Handling Unit Using Residual and Recursive Parameter Identification Methods. *ASHRAE Transactions*, Vol. 102, No. 1, 528-539, 1996.

Lee, W. Y.; House, J. M.; Shin, D. R. 1997. Fault Diagnosis and Temperature Sensor Recovery for an Air-Handling Unit. *ASHRAE Transactions*, Vol. 103, No. 1.

Marsella, S. and Johnson, L. 1997. A Pedagogical Agent for Team-Training in dynamic multi-agent virtual worlds. *Proceedings of ITS-98*.

McQuiston, F. C. and Parker, J.D. 1994 *Heating, Ventilating, and Air Conditioning Analysis and Design*. John Wiley & Sons.

Subramanian, S. 1995. Multiple fault Diagnosis of Continuous Dynamic Systems using Behaviourial Modes. Ph.D Diss., Dept. of Computer Science, University of Texas at Austin.

Whole Building Diagnostician.
<http://www.buildings.pnl.gov:2080/wbd/content.htm>