SimBuild
2012

Fifth National Conference of IBPSA-USA
Madison, Wisconsin
August 1-3, 2012

FAULT DIAGNOSIS IN HVAC SYSTEMS BASED ON THE HEAT FLOW MODEL

Alexander Schiendorfer[1], Gerhard Zimmermann[2], Yan Lu[1] and George Lo[1]
[1]Siemens Corporation, Corporate Research and Technology, Princeton, NJ
[2]Informatik, University of Kaiserslautern, Germany

ABSTRACT

Fault Detection and Diagnosis based on the Heat Flow Model (HFM) provides a generic and extensible framework for monitoring HVAC systems. It supports the finding and fixing of faulty components. During the fault detection phase, measured sensor and control values are used to perform estimations based on the physical properties of the system. Discrepancies of estimated and measured values are collected as a detection failure vector. Diagnosis seeks to find the most probable cause for the observed failures. In HVAC systems, the failures and faults form an m-n relation. Our proposed diagnosis is performed with an associative network to map the relations among failures and faults using the inherent fault simulation capabilities of the HFM nodes at runtime. The similarity of the detection failure vector to the simulated failure vector indicates the probability of the corresponding fault. To find the best method of fault diagnosis, this paper examines different similarity metrics for HFM based FDD, including Euclidean distances, Manhattan distance, root of sum of products, Jaccard index, and a table based metric. The effectiveness of the proposed diagnosis approaches is presented with a case study based on a reference implementation using Simulink and Java.

INTRODUCTION

Buildings consume a significant amount of energy and resources. In the United States buildings use 72% of electricity, 54% of natural gas and 38.9% of the total energy consumption according to the U.S. Department of Energy (U.S. Department of Energy 2009). They are designed to provide comfortable air quality, temperature and humidity for people working in the buildings. However, many HVAC systems cannot meet energy consumption expectations due to multiple faults that can occur throughout their lifecycle. Physical problems such as stuck dampers, leaking valves, or wrongly configured controllers are examples of HVAC faults that need to be detected and diagnosed by commissioning. Some faulted HVAC systems can introduce 20% of energy waste (Roth and Quartararo 2005) which motivates the development of automatic fault detection and diagnosis (FDD) systems.

Related Work

There are two main categories for detection systems (Wu and Sun 2010): statistical and model based FDD. Statistical FDD systems try to detect abnormalities by comparing real-time data to data gained after recommissioning of a building or from a detailed building model providing fault-free data. (Morisot and Marchio 1999) pointed out that the quality of FDD highly depends on the input data when using artificial neural networks for detection of failures. The system needs to be provided with correct data for as many circumstances as possible including different air temperatures and humidity. Real data covering important cases are not always available.

On the other hand, model-based FDD systems simulate the expected physical values of a HVAC system by building a model of it. They then calculate the outcome for given outside air temperature values and other parameters. A model-based approach to fault detection and diagnosis consists of a mathematical description of the system using equations of the thermodynamic processes that happen in the HVAC components (Salsbury and Diamond 2008).

Motivation

Most existing FDD systems are hardware or level dependent and require configuration and calibration (Wu and Sun 2010). The proposed Heat Flow Model is a model based approach using a simplified physical model designed for fast reconfiguration.

An HFM based simulation engine estimates the properties of mass flows based on physical models and compares them with measured values from sensors. Using object oriented programming, HVAC components are modeled on an abstract level as classes which can be parameterized for concrete components. Program libraries are thus developed and reconfigured at runtime. By simulating possible faults within the scope of one component's data, the programmed components do not require reference data.

In (Zimmermann, Lu, and Lo 2011) we presented results for different faults and their diagnosis rate for one particular recognition strategy. In this paper we try to determine more efficient metrics based on the performance for all possible simulated faults.

HFM BASED FAULT DETECTION

The Heat-Flow-Model for FDD

The proposed method for fault diagnosis uses the Heat Flow Model (HFM) which is described in (Lu, Zimmermann, and Lo 2010). HFM is component-level FDD based on first principles i.e. the analyis of the underlying physical processes involved with mass air flows (Have 1997). *Heat flow* hereby refers to a generalization for all kinds of flows that are related to the energy or heat flow such as temperature, humidity or air pressure. In essence, a heat flow model is a directed graph with its nodes representing the components of the HVAC system and arcs representing the mass flow connections among them such as ducts or pipes. Existing BIM, such as an IFC model, ease the creation of the HFM graph. HFM nodes contain estimation functions to predict the changes on the flow properties such as air temperature or air pressure to perform detection and diagnosis in steady state. A *flow variable* refers to a physical value such as temperature or humidity and is sent towards adajcent HFM components.

A node consists of three major elements:

- Ports in upstream and downstream mass flow directions to connect nodes
- Estimation and simulation formulas to calculate the changes to an incoming flow variable
- Rule definitions to define which flow variables are compared to detect failures

Figure 1 depicts the internal structures of a modeled flow node. It contains flow ports for both directions (upstream and downstream). Two adjacent nodes thus refer to the same value in reality. Take for instance an HVAC system with a temperature sensor laying upstream of a heating coil followed by a cooling coil which would result in an HFM as shown in Figure 2. Sensor nodes trigger the propagation and estimation of the flow variables. Hence, the particular sensor node sends its incoming temperature flow variable to the *forwardIn* port of the heating coil which adds an estimated temperature rise based on its current control value and sends it to the cooling coil.

Logically in parallel, the cooling coil adds the estimated temperature drop to a value coming from a sensor laying downstream in a reverse propagation. Thus, the values being sent forward by the heating coil and the ones sent backward by the cooling coil can be compared.

Sensor and control data are sent to their HFM components such that the estimation and propagation can be done with the information available to a single node. Each sensor node initiates a propagation of a flow variable in one timestep.

The FDD engine works in three phases which will be explained in the following sections:

1. **Fault Detection:** Sending sensor and control data to the HFM components, estimating their changes, propagating them to neighbors and storing a *detection failure vector* of comparisons.
2. **Fault Simulation:** Simulating faults to create *simulated failure vectors* that are compared with the detected values.
3. **Fault Diagnosis:** Finding the *closest vector* among the simulated ones to ”blaming” it for causing the observed data.

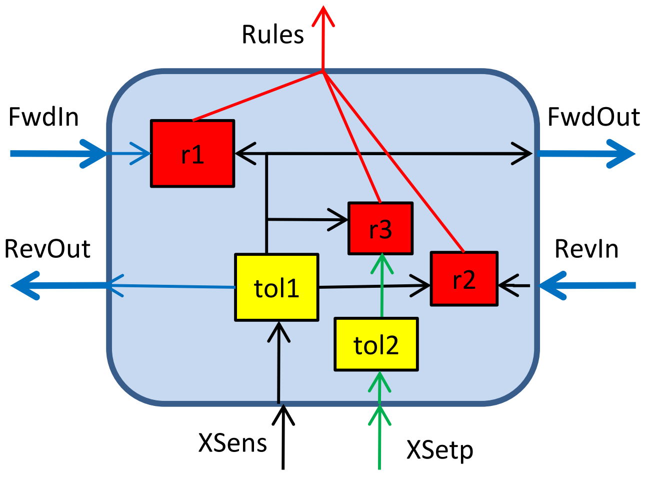

Figure 1: Example of an HFM node with ports fwdIn, fwdOut, revIn and revOut as well as rule definitions r1, r2 and r3 that combine the values of ports. Tol1 and tol2 stand for added sensor tolerances.

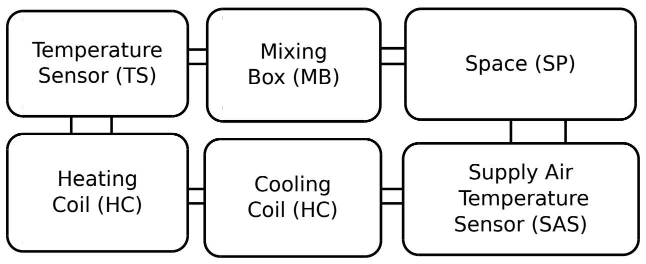

Figure 2: Exemplary simplified HFM graph of an HVAC system

Rule Evaluations, Intervals and Uncertainty

To come up for the uncertainty associated with measured values and due to estimations, intervals are used for flow variables. A generic interval comparison rule for intervals is used throughout the HFM graph as shown in Equation 1. Concrete instantiations of this rule consist of the definition of the originating ports and an identifier. Each rule is evaluated to get a *failure value*. Typical examples are the pairs sensor setpoint interval and actual sensor input or adjacent nodes in the flow as mentioned in the example earlier. If there is any overlap between the two intervals failure value 0 is reported. If however intervals are disjointed in either direction, a nonzero failure value is stored for this rule.

For instance, the rule ”SupplySensDuctTempSens - FwdFail” compares the intervals of the forward outgoing temperature flow of the supply sensor duct with the reverse incoming temperature flow. Vectors of failure values are collected for detected and simulated faults hence they will be referred to as ”detected” and ”simulated” failure vector. A more rigorous formal definition for the objects involved is given in Section *HFM FAULT DIAGNOSIS*.

Faults and Failures

Faults and detected failure values are generally in an m:n relationship as illustrated in the form of an associative network in Figure 3. One fault causes multiple failure values (Baer 1996). In turn, those failure values can be caused by multiple faults. For instance a too high supply air temperature might indicate a malfunctioning cooling or heating coil or a drifting sensor.

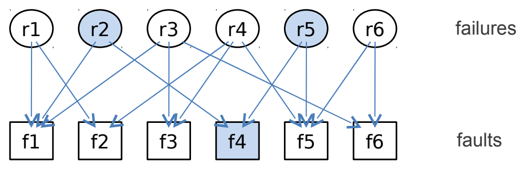

Figure 3: Faults are related to multiple failure values and vice versa

A failure value fv is a the result of an evaluation of Equation 1. Moreover, failure values have to be normalized using typical value ranges. This value reflects a semantic decision built into the FDD engine as a failure value is not equal to 0 if and only if there is *no* overlap of the compared intervals. Thus a partial overlap is counted as no failure due to the large intervals resulted from estimation and simulation.

$$fv(I_1,I_2) = \begin{cases} 0 & \text{if } I_1, I_2 \text{overlapping} \\ I_2.\text{min} - I_1.\text{max} & \text{if } I_2.\text{min} > I_1.\text{max} \\ I_2.\text{max} - I_1.\text{min} & \text{if } I_2.\text{max} < I_1.\text{min} \end{cases} \quad (1)$$

Local Fault Simulation Capabilities of Nodes

After the collection of the detected failure vector the flow is simulated with one fault inserted in one component at a time (single fault assumption) in the *fault simulation phase*. The modeled flow proceeds as in the detection phase except for sensor and other input values which are replaced by simulated values. The FDD engine activates every possible fault in any component to create a simulated failure vector.

Assume that the changes of the air heat flow when passing a cooling coil can be approximated using the following equation.

$$T_{out} = T_{in} - u_{cc} * T_{\text{avgDrop}} \quad (2)$$

If the engine orders the cooling coil to simulate a stuck cooling coil at 0 this equation changes to:

$$T_{out} = T_{in} - 0 * T_{\text{avgDrop}} \quad (3)$$

Because of these component-local simulation capabilities the FDD system no measured history data are needed for fault diagnosis, in contrast to other, learning-based simulation methods for diagnosis such as artificial neural networks. Also, this enables reuse of components for different HVAC systems since those are kept independent from each other.

HFM FAULT DIAGNOSIS

Fault diagnosis aims to find the most probable fault or faults with patterns similar to the one observed. More concretely, the detected and simulated failure vectors for an inserted fault need to match for a diagnosis decision. The proposed elementary diagnosis algorithm assigns a score to every possible fault and ranks them accordingly. Users will be most concerned with the highest ranked fault since the system recommends to check the involved component for faults.

Definitions

For n defined rules in the HFM graph the detected failure vector D contains n components. For every fault i that can be simulated there exists a simulated failure vector S_i also containing n components for the same rules.

$$D = < d_1, d_2, \ldots d_n > \quad (4)$$

$$S_i = < s_{i_1}, s_{i_2}, \ldots s_{i_n} > \quad (5)$$

A scoring metric sm is a function that returns a real number $sc_i(score)$ for a detection failure vector D and a simulation failure vector S_i denoting the similarity of these vectors. All metrics assign a *rank*, r_i, based on sc_i using the *precedes*-predicate in Equation 8. This predicate needs to be defined with the used metric since it either arranges ranks in ascending or descending order (e.g. with a distance based metric, a *small* score leads to a *high* rank whereas in an explicit score assignment a *high* score leads to a *high* rank). The rank r_i indicates the likelihood of fault i having occurred given the the pattern of detected values. The diagnosis algorithm returns a top-list, tl, containing all simulated faults ordered by the rank of their simulation vector. The relation $i \prec_{tl} j$ as noted in Equation 9 means that the simulated fault i is ranked higher than the simulated fault j.

$$sm : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \quad (6)$$

$$sc_i = sm(D, S_i) \quad (7)$$

$$r_i < r_j \leftrightarrow precedes(sm, sc_i, sc_j) \quad (8)$$

$$\forall i, j \in \text{simFaults} : i \prec_{tl} j \leftrightarrow r_i < r_j \quad (9)$$

Diagnosis algorithm

In the current version the algorithm used for diagnosis works as shown abstractly in Figure 4.

```
Require: HFM has been run in detection and set detection failure vector $D$
Ensure: returns fault $i$ with highest probability
 1: procedure SIMULATE-FAULTS(HFM, D)
 2:     for all $t$ in timesteps do
 3:         for all $i$ such that $i$ can be simulated do
 4:             $tl \leftarrow \emptyset$          ▷ $tl$ is the top-list of faults
 5:             $co \leftarrow$ FIND-COMPONENT-FOR($i$)
 6:             APPLY-FAULTY-STATE($co, i$)
 7:             SIMULATE-FLOW(HFM)
 8:             $S_i \leftarrow$ GET-RESULT(HFM)
 9:             $sc_i \leftarrow$ASSIGN-SCORE($S_i$, $D$)
10:             INSERT($tl$, $S_i$, $sc_i$)
11:         end for
12:         return sorted($tl$)      ▷ order by score $\rightarrow$ rank
13:     end for
14: end procedure
```

Figure 4: Diagnosis Algorithm based on simulation

Evaluation of similarity metrics for diagnosis

During the *fault diagnosis* phase, several simulated result vectors are calculated to serve as the basis for comparison. It is a heuristic task to find a good metric for similarity since plain vector distance does not perform precisely enough in terms of diagnosis matching.

Since partial overlap leads to 0 values, the gap between the failure values -0.2, 0 and 0.2 is more relevant than their numeric value indicates. Failure values 4.1 and 4.3, on the other hand, do not characterize a failure vector pattern as much. Consequently, smaller failure values tend to have emerged from estimation tolerances.

A good similarity metric fulfills the following requirements:

1. If the failure values are large and close, the resulting score should be high.
2. A larger absolute failure value more significantly indicates a particular fault. Smaller failure values are more prone to have resulted from estimation. Therefore large values should influence the similarity stronger than small ones.
3. The FDD engine should be able to diagnose the actual *fault* and list it high in the top list.

Different Scoring Metrics

In order to find a good metric for the proposed diagnosis algorithm, several approaches are discussed and implemented:

1. **Component-wise Multiplication** *(CWM)*: The score is calculated by multiplying D_j with s_{i_j}, taking the square root of the absolute value of the product and multiplying it with the sign of the product. Here, the largest score is ranked highest (descending metric). $sc_i = \sum_{j=1}^{n} sig(d_j * s_{i_j}) * \sqrt{|d_j * s_{i_j}|}$
2. **Euclidean Distance** *(ED)*: Since taking the square root on both sides does not affect the order relation between two vectors, only the sum of squared distances is used. This metric does not treat 0 values any differently and the absolute failure value does not influence the rating either. Close small values yield a better score than farther large values even though the large values indicate better matching (e.g. 0.1 and 0.2 leads to better results than 5.7 and 6.3. Here, the smallest score is ranked highest (ascending metric) $sc_i = \sum_{j=1}^{n} (d_j - s_{i_j})^2$
3. **Weighted Euclidean Distance** *(WED)*: Using weights helps to overcome the problems of the plain Euclidean distance. This metric takes the value of the detected failure value (d_j) as an indicator for congruence. The squared distance is divided by d_j which leads to smaller values if the absolute failure values are high. Small values mean low distance thus high ranking. Alternatively, concrete weights could be assigned to the rule outputs by experts or machine learning algorithms. $sc_i = \sum_{j=1}^{n} \frac{(d_j - s_{i_j})^2}{f(|d_j|)}$ where $f(x) = \begin{cases} 1 & \text{if } x = 0 \\ x & otherwise \end{cases}$
4. **Explicit Scoring Matrix** *(ESM)*: A matrix assigns explicit score values to failure values based on their size and sign. Figure 5 shows the score of the failure values, small ones almost do not change the score whereas large values have a strong influence. If one of the two compared values is zero the score for this comparison is zero. Two values with opposite signs reduce the overall score. A function *toIndex* sets failure values relatively to the overall maximum or minimum failure value of the detection result to achieve a normalized value which is the lookup-index for the scoring matrix. The fault with the highest score is ranked best (descending metric). $sc_i = \sum_{j=1}^{n} \text{scoringMatrix}[\text{toIndex}(d_j), \text{toIndex}(s_{i_j})]$
5. **Jaccard Index** *(JCI)*: This pattern recognition metric relies on the ratio of intersecting rules to all rule evaluations. Failure values are separated into the classes positive, neutral and negative. Then the amount of equally classified rules in detection and simulation are counted to return the similarity coefficient. $sc_i = \frac{|D \cap S_i|}{|D|}$ where $D \cap S_i := \{j | sig(d_j) = sig(s_{i_j})\}$ and $sig(x) = -1$ iff $x < 0$, $sig(0) = 0$ and $sig(x) = 1$ iff $x > 0$.
6. **Manhatten Distance** *(MHD)* : This simple metric approximates distances. Like Manhatten's street structures, distances are given in blocks hence the sum of the distances in every dimension is taken. $sc_i = \sum_{j=1}^{n} (|d_j - s_{i_j}|)$

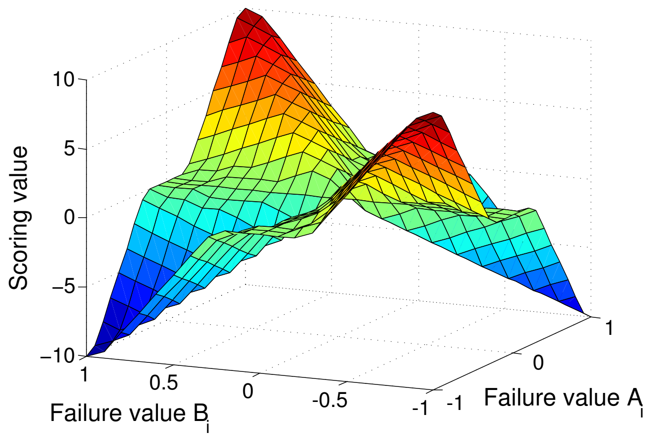

Figure 5: Explicit Scoring Matrix for two failure values A and B

Small Diagnosis Example

Figure 2 shows a small example for demonstrating diagnosis. It is an HFM graph of a simplified HVAC system consisting of a mixing box responsible for mixing outdoor and return air, a heating coil for preheating outdoor air, a cooling coil to maintain the supply air temperature at a setpoint and one space representing different zones with reheat VAVs. For simplicity reasons we do not consider mass air flow rates and other physical values and generally estimate 0.5 °C as sensor tolerances.

Detection Phase

Sending a sensor value means that the corresponding HFM sensor node receives the value at its *XSens* port in this context (tolerances are added by *tol* definitions) – Propagation refers to the estimation and passing the estimated interval to adjacent nodes using *forward* and *reverse* ports. Assume that the building information system provides the following data:

- T_{oa}**:** Outdoor air temperature sensor data that is sent to the mixing box
- T_{sa}**:** Supply air temperature sensor data
- T_{ra}**:** Return air temperature sensor data sent to the space
- u_{hc}, u_{cc}, u_{mb}**:** Heating, cooling coil, and mixing box control data
- SPT_{sa}**:**Supply air setpoint temperature

Moreover the following rules are defined for detection:

- **CCForw-SASRev:** Cooling Coil Forward - Supply Air Sensor Reverse
- **MBForw-HCRev:** Heating Coil Reverse - Mixing Box Forward
- **SASSP-SASDIn:** Supply Air Sensor Setpoint - Supply Air Sensor Data In

Hence, the length of the detection failure vectors and all simulated failure vectors will be 3. We will have a look at the effects of diagnosis for one time step.

First, the sensor and control data from Table 1 are sent to the components. In cooling mode, no additional preheating is needed. Thus u_{hc} is set to 0.0, u_{mb} is set to 0.1 to reuse most of return air with a temperature based economizer controller (Taylor and Cheng 2010) and additional cooling ($u_{cc} = 0.8$) is required. Due to occupancy, return air rises to 25 °C. Readers might examine Table 1 to guess which component fails in the scenario. A structured analysis based on HFM FDD will lead to possible faults.

Table 1: Sensor data for one time step

Data	Value
T_{oa}	30 °C
T_{ra}	22 °C
T_{sa}	20 °C
SPT_{sa}	19 °C
u_{hc}	0.0
u_{cc}	0.8
u_{mb}	0.1

In the following steps the estimations needed for rule evaluations are calculated. Sensor nodes invoke the propagation by first adding tolerances and sending the resulting intervals in both directions. *Transformation nodes* such as a cooling coil change the intervals during propagation. For estimation, the modeler of the HFM graph needed to add a minimal and maximal temperature drop for the cooling coil, e.g. 11 °C and 15 °C, respectively.

We will focus on the ”path” the intervals take from the sensor input to the port involved in a comparison and show the process in detail for the rule *CCForw-SASRev*:

1. **MB:** Reads sensor value 30 °C with tolerances $\rightarrow$ [29.5 °C; 30.5 °C] *forward*
2. **MB:** Mixes with return air ([21.5 °C; 22.5 °C]) at rate $u_{mb} \rightarrow$ [22.3 °C; 23.3 °C] (e.g. $22.5*0.9+0.1*30.5 = 23.3$) *forward*
3. **HC:** u_{hc} is set to 0, no changes in HC $\rightarrow$ *forward* to CC
4. **CC:** Using u_{cc}, interval is calculated as max = $23.3-11.0*0.8 = 14.5$, min = $22.3-15*0.8 = 10.3 \rightarrow$ [10.3 °C; 14.5 °C] *forward*
5. **SAS:** Reads sensor value 20 °C with tolerances $\rightarrow$ [19.5 °C; 20.5 °C] *reverse*

Eventually, a failure value is calculated from both intervals using Equation 1 to yield 5 as the failure value for *CCForw-SASRev*.

Similar to that the propagation in reverse direction is started from SAS.

1. **SAS:** Reads sensor value 20 °C with tolerances $\rightarrow$ [19.5 °C; 20.5 °C] *reverse*
2. **CC:** Using u_{cc} for transformation $\rightarrow$ [28.3 °C; 32.5 °C] *reverse*
3. **HC:** u_{hc} is 0, no changes in HC $\rightarrow$ *reverse* to MB

4. **MB:** Comparing HC-rev [28.3 °C; 32.5 °C] with MB-forw [22.3 °C; 23.3 °C] $\rightarrow$ failure value of *MBForw-HCRev* is 5

SAS also calculates the rule evaluation *SASSP-SASDIn* for setpoint and sensor value. In this time step [18.5 °C; 19.5 °C] for the setpoint and [19.5 °C; 20.5 °C] does not lead to a failure value for *SASSP-SASDIn* since 19.5 °C lies in both intervals.

After the the detection phase the detection failure vector $D =< 5,5,0 >$ is stored for comparison with simulated failure vectors and indicates that the system is not in a fully working state.

Simulation Phase

During the simulation phase, the engine calculates *control* and *sensor* values using feedback loops. Only the outside air temperature and supply air setpoint temperature are taken from the measurements which requires special treatment of outside air temperature sensor faults as mentioned in (Zimmermann, Lu, and Lo 2011).

In this scenario, only two possible faults are studied from the reference list of (Castro 2003). A low return air temperature compared to the supply air set point temperature and still high amount (0.8) of cooling indicate problems with the mixing box, so a simulated stuck outside air temperature damper is considered.

The HFM graph in Figure 2 shows an air flow loop. In addition, feedback loops for the control variables u_{mb}, u_{hc}, and u_{cc} exist. Steady state values for all variables are achieved by iteratively executing the air flow loop and the control loops. Otherwise, a set of differential equations had to be solved. Equation 11 shows the details of MB simulating a stuck valve where T' stands for the simulated temperature used to calculate the expected simulated sensor temperatures. Equation 10 introduces u'_{mb} as an intermediate control value using $vf \in \{-1,1\}$ for the simulated *valve fault*.

$$u'_{mb} = \text{limit}(0,1,u_{mb} + vf) \tag{10}$$

$$T'_{out} = T'_{ra} * (1 - u'_{mb}) + T'_{oa} * u'_{mb} \tag{11}$$

Note that u_{mb} rather than u'_{mb} would be identified as the simulated control value. u'_{mb} is only used to simulate the effects a stuck valve has on the temperature. MB send T'_{out} with the effects of a fully open outside air damper, yet still preserves the used u_{mb} which would be minimized due to the controller strategy.

Equation 11 assumes a linear damper characteristic and 100 % damper authority. More complex damper models can be introduced if necessary.

Assume that the simulation process starting with 30 °C outside air temperature returned the control values $u_{cc} = 0.9$, $u_{mb} = 0.15$ and $u_{hc} = 1$ and a supply air temperature of [17.8; 18.8]. Performing the same propagation steps as during detection leads to a similar outcome: *CCForw-SASRev* - 4, *MBForw-HCRev* - 4 and *SASSP-SASDIn* - 0, so $S_1 =< 4,4,0 >$.

Another possible explanation is a sensor drift in the supply air temperature sensor i.e. a too low measured value. In that simulation SAS adds an offset (e.g. -5) to T'_{out} in its simulation propagation. This would return the simulation vector $S_2 =< -3.6,-3.6,2.4 >$.

To sum up, all failure value vectors (detection and simulation) are shown in Table 2

Table 2: Sensor data for one time step

	CC-F SAS-R	MB-F HC-R	SASSP-SASDIn
D	5	5	0 °C
S_1	4	4	0°C
S_2	-3.6	-3.6	2.4

Diagnosis Phase

In diagnosis, D and S_1, S_2 are compared using the presented metrics and ranked according to their score. All metrics classified fault 1, the stuck outside air damper, correctly due to the already similar vectors. Results are shown in Table 3.

Table 3: Diagnosis Results for Small Example, simulated fault : Ranks in different metrics

Metric	sc_1	sc_2	r_1	r_2
ESM	16.0	-13.0	1	2
JCI	1.0	0.0	1	2
CWM	8.94	-8.49	1	2
MHD	2.0	19.6	1	2
WED	0.4	35.34	1	2
ED	2.0	153.68	1	2

After diagnosis, the engine suggests checking the outside air damper first. In this scenario, mechanical cooling comes up for the lost return air in this small example as the supply air setpoint temperature can still be reached. The fault in the mixing box would not only led to higher energy demands and costs, it would also remain unnoticed by people working inside the building. Diagnosis would help to point to the source of failures and suggest fast repairing.

EXPERIMENT SETUP

The case study

For the experiment's purposes the same case study used in (Zimmermann, Lu, and Lo 2011) is considered. The so called "Small bank" example consists of three spaces with individually controlled reheat VAVs and one central AHU with economizer. It was used for first experiments because a basic IFC building information model existed. This was then compiled into an HFM graph and augmented with additional required information such as sensor tolerances. Features of the experimental HVAC system of the Iowa Energy Center Energy Resource Station (ERS) were modeled to make comparison to other HVAC FDD research projects possible. To achieve valid input data resulting from a faulty state of the HVAC system a fault simulator has been built in Simulink. Faults such as sensor drifts can interactively be inserted to produce reference output.

The setup

For testing the quality of the FDD diagnosis engine, an experimental setup for different scoring metrics has been set up. 12 components were capable of simulating faults in the HFM model itself including heating coil valve stuck and leaking errors or sensor drifts. The faults have been taken from the reference list of (Castro 2003). In total, 48 faults were simulated and a reference state without failures. For every fault that can be simulated in the Simulink fault simulator, the engine has been started for detection and diagnosis. In total, sets of sensor and control data are simulated in intervals of 6 minutes for 5 days which results in 1200 time steps.

Temperatures are ranging from -14 °C to 40 °C in total and a daily sinusoidal variation between 4 am and 4 pm of 14 °C. There is regular occupancy between 7 am and 6 pm in all modeled spaces. Space temperature set-point ranges are (20 °C, 22 °C) during occupancy and (17 °C, 25 °C) otherwise. Occupancy also determines the heat load in the spaces (G. Zimmermann and G. Lo 2011).

At each time step every simulated fault is ranked as described earlier and overall the following quantities are aggregated and calculated. Note that f stands for the intended fault introduced through simulation which should be diagnosed by the engine.

- **Average Rank:** The rank f got assigned on average over all time steps. A good metric results in a lower average rank.
- **Occurrences in Top 5:** Lists how often f was ranked among the top 5 faults by diagnosis. This makes up for possible outliers regarding scores and ranks since a good metric repeatedly lists f in its top 5 list.
- **Occurrences in as top-diagnosed fault:** Lists how often f was ranked as the top fault candidate.
- **Relative Occurrences:** The percentage of the occurrences as top and in top 5 is presented relative to the total number of time steps. This gives a tractable idea of how the rankings are perceived by a mechanical operator.

RESULTS

Overall results show the average values of the criteria discussed earlier that every metric returned. Furthermore exemplary results are shown for two simulated faults.

The results of a simulated leaking cooling coil valve led to the results in Table 4. Whereas the explicit scoring matrix yielded the better average rank, Jaccard index showed the intended fault as top candidate more often. Hence the measured categories do not strictly influence each other.

Table 5 shows that metrics notably influence the quality of the diagnosis. ESM listed the sensor drift of the supply air sensor as fault correctly in 66% of all timesteps. Numeric metrics such as the Euclidean distance yielded worse results probably because of the uniform treatment of 0 and other failure values as well as ignoring large failure values.

Table 4: Results for Cooling Coil Valve Leak. R_{avg} : average rank, Occ_5 : occurrences in top 5 out of 1200 time steps, P_5 : percentage in top 5, Occ_1 : occurrences as top listed fault, P_1 : percentage as top

Metric	R_{avg}	Occ_5	P_5	Occ_1	P_1
CWM	1.91	1192.00	0.99	525.00	0.44
ESM	2.34	1121.00	0.93	210.00	0.18
JCI	3.52	793.00	0.66	432.00	0.36
ED	13.57	788.00	0.66	608.00	0.51
WED	13.59	783.00	0.65	609.00	0.51
MHD	15.61	728.00	0.61	585.00	0.49

Table 5: Results for Supply Air Sensor Temperature too High. R_{avg} : average rank, Occ_5 : occurrences in top 5 out of 1200 time steps, P_5 : percentage in top 5, Occ_1 : occurrences as top listed fault, P_1 : percentage as top

Metric	R_{avg}	Occ_5	P_5	Occ_1	P_1
ESM	1.40	1193.00	0.99	788.00	0.66
CWM	1.41	1193.00	0.99	895.00	0.75
JCI	2.60	1127.00	0.94	349.00	0.29
WED	4.41	1018.00	0.85	909.00	0.76
ED	5.96	847.00	0.71	773.00	0.64
MHD	7.25	786.00	0.66	701.00	0.58

The overall results across all simulated faults for every tested metric are shown in Table 6. Figure 7 depicts the average ranks returned by the metrics, Figure 6 shows the percentages of the intended fault being listed as top cause or in the top 5 list.

The explicit scoring matrix returned the lowest rank on average, Jaccard Index showed the intended fault in top 5 more than any other metric and CWM returned the actual fault on top position the most times. A user of the system would consequently have seen the ”true” cause at the top of the fault list many times and could check the possibly fault component.

CONCLUSION

This paper describes in detail the algorithmic strategies pursued in HFM based fault diagnosis. Different metrics were used in combination with a simulation based comparison procedure to yield the most probably occurred faults.

Currently the HFM system is applied to a new building in Berkeley, Ca., with 7 levels and 157 zones that are air-conditioned by 2 AHUs. Insights from the continuous commissioning project will be used to improve the diagnosis capability.

Table 6: Average over all tested faults for each metric. R_{avg} : average rank, Occ_5 : occurrences in top 5 out of 1200 time steps, P_5 : percentage in top 5, Occ_1 : occurrences as top listed fault, P_1 : percentage as top

Metric	R_{avg}	Occ_5	P_5	Occ_1	P_1
ESM	3.21	1012.78	0.84	296.54	0.25
JCI	3.28	1072.24	0.89	242.98	0.2
CWM	4.39	828.71	0.69	372.78	0.31
MHD	14.42	460.02	0.38	248.66	0.21
WED	14.84	464.27	0.39	252.9	0.21
ED	16.49	482.85	0.4	265.41	0.22

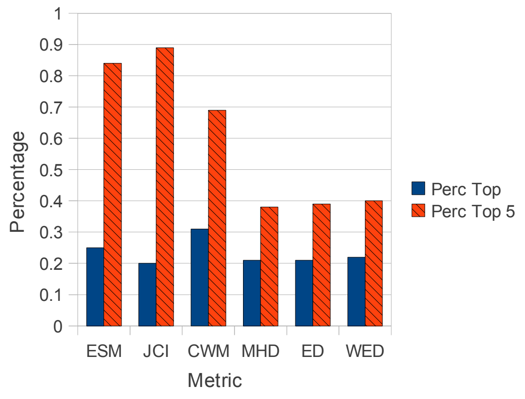

Figure 6: Percentages as top and among top 5 of different metrics

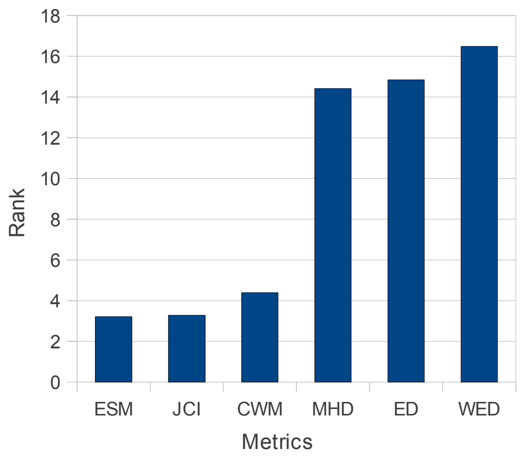

Figure 7: Average rank of different metrics

More information in BIM is expected to become available so that more precise rules can be defined. Ongoing research includes the detection and diagnosis of control errors as well as proposing more efficient control strategies.

First experiments showed that an explicit scoring matrix used for the interval comparisons that arose with the artificial setup worked well, the exact weights have yet to be defined and tested with data from the current application in Berkeley.

REFERENCES

Baer, et al. 1996. "Associative Networks." *In Building Optimization and Fault Diagnosis Source Book, IEA Annex 25.* p.219–222.

Castro, N.S., et al. 2003. "Results from Simulation and Laboratory Testing of Air Handling Units and Variable Air Volume Box Diagnostic Tools." *NISTIR 6964 Report, January 2003.*

G. Zimmermann, Y. Lu, and G. Lo. 2011. "Heat Flow Model for Building Fault Detection and Diagnosis." no. 20110093424 (April).

Have, Philip. 1997. "FAULT MODELLING IN COMPONENT-BASED HVAC SIMULATION." *Proceedings of Building Simulation'97, 1997 - ibpsa.org.*

Lu, Yan, Gerhard Zimmermann, and George Lo. 2010. "Heat flow modeling of HVAC systems for fault detection and diagnosis." *Proceedings SimBuild 2010, Fourth National Conference of IBPSA-USA, New York.* 215–222.

Morisot and Marchio. 1999. "Fault detection and diagnosis on HVAC variable air volume system using artificial neural networks." *Proc. IBPSA Building Simulation.*

Roth, K. W., Westphalen D. Feng M. Y. Llana P., and L. Quartararo. 2005. Energy Impact of Commercial Building Controls and Performance Diagnostics: Market Characterization, Energy Impact of Building Faults and Energy Savings Potential.

Salsbury, Tim, and Rick Diamond. 2008. Model-Based Model Based Diagnostics for Air Handling Units.

Taylor, and Cheng. 2010. "Why Enthalpy Economizers Dont Work." *ASHRAE Journal* 42 (November): 12–28.

U.S. Department of Energy. 2009. Building Energy DataBook. `http://buildingsdatabook.eren.doe.gov/`.

Wu and Sun. 2010. Multilevel Fault Detection and Diagnosis on Office Building HVAC Systems.

Zimmermann, Gerhard, Yan Lu, and George Lo. 2011. "A simulation based fault diagnosis strategy using extended heat flow models(HFM)." *Proceedings of Building Simulation 2011: 12th Conference of International Building Performance Simulation Association, Sydney, 14-16 November.* 405–412.