Intelligent Application of Fault Detection and Isolation on HVAC System

By

Davood Dehestani

Submitted to the Faculty of Engineering and Information Technology

in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

at the University of Technology, Sydney

UNIVERSITY OF TECHNOLOGY SYDNEY

March 2014

Certificate of Authorship / Originality

I, Davood Dehestani, certify that the work in this thesis has not previously been submitted for a degree, nor has it been submitted as part of the requirements of a degree, except as fully acknowledged within the text.

I also certify that this thesis has been written by me. Any help that I have received in my research work and in the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

April 2013

Acknowledgements

First and foremost, I would like to express my sincere gratitude and appreciation to my supervisors, Dr. Steven Su and Professor Hung Tan Nguyen, for their constant guidance to my deeper understanding of the knowledge, and their invaluable comments during the whole research work on this dissertation. They always push the limit of my ultimate research ability and never accept less than my best efforts. I also express my thanks to Dr. Ying Guo from CSIRO for her valuable advices and her assist on my research.

Special thanks are given to Dr. Guang Hong and Dr. Jafar Madadnia for making their help over the years with various aspect of hardware configuration during laboratory test. Also I would like to thanks from my friends Vahik Avakian and Peter Tawadros for their help on the air condition laboratory. Thanks for everyone involved in the experimental studies, whether mentally or physically support, each of you have contributed significantly to the results in this thesis.

Thanks for all of my friends in UTS which help me in any part of this study. Special thanks from my best friend Yashar Maali for all of his support during this research. I would also like to thank Phyllis Agius, Craig Shuard and Gunasmin Lye and any people who is working to provide supports of research students during postgraduate studies.

Last but certainly not the least; I want to thank my wife Fahimeh for her persistent understanding and support. I wouldn't have had the will to continue my course of study were it not for her encouragement and comfort, and I wouldn't have had the ability were it not for her willingness to take on more than her share.

Table of Contents

List of Figure

List of Table

Abstract

Efficient heating, ventilation, and air-conditioning (HVAC) systems are one of the big challenges today around the world. The fault detection and isolation (FDI) play a significant role in the monitoring, repairing and maintaining of technical systems for the final destination of safety and cost reduction. FDI makes an infrastructure to effectively reduce total cost of maintenance and thus increases the capacity utilization rates of equipment. Reduction of energy wasting in the system by real-time fault detection is another goal. Among all HVAC system's studies, the focus of this thesis is on developing of fast and reliable FDI structure that can cover all subsections of HVAC system including cooling tower, chiller and air handling units (AHU) which greatly affect building energy consumption and indoor environment quality.

The first stage of this study is to develop and validates a mathematical HVAC model then follows by simulation and sensitivity analysis. The simulation makes a good capability of producing artificial fault free and faulty data for review of any upcoming failure over the HVAC system. These data with wide range of fault severities can be used to assess the performance of HVAC automated fault detection and isolation (AFDI) system.

Two categories of process history diagnosis methods have been reviewed and assessed for the development of AFDI algorithms at second stage of this study. Principal component analysis (PCA) and support vector machine (SVM) classification are two chosen algorithm which have been analysed in depth and initially tested by simulated data from stage one. This review has been continued by developing online SVM algorithm with incremental learning technique and then tested both on simulated and operational data.

An experimental rig is designed and applied in the last stage of this research. This setup is configured inside the HVAC laboratory of UTS to collect operational data for the operating test. Operational data as outcome of this stage was then used for test of developed AFDI from last stage. Artificial neural network (ANN) algorithm compressed in frame of black box model for fault free reference. Finally, a combination of black box model and developed AFDI was tested and evaluated for cooling tower and air handling unit (AHU) faults based on operational data. The result shows increasing of robustness, performance and accuracy for the proposed AFDI over the operational data.

Keyword: heating, ventilation, and air-conditioning (HVAC), fault detection and isolation (FDI), mathematical model validation, Support Vector Machine (SVM), robust fault detection (RFD).

Chapter 1

1 Introduction

1.1 Problem Statement

Heating, Ventilating, and air-conditioning (HVAC) systems are important parts of a building system. HVAC systems provide building occupants with a comfortable and productive environment. The energy consumption of building HVAC systems constitutes 14% of the primary energy consumption in the U.S (DOE, 2003), and about 32% of the electricity generated in the U.S (ASHRAE, 2000). It is a big challenge to improve the whole building energy efficiency while maintaining the indoor environmental quality.

Faults in building HVAC systems, including design problems, equipment and control system malfunction, may result in energy waste. If early detection and diagnosis of faults are possible, energy waste could be avoided. Examination of data from a number of UK buildings showed avoidable waste levels in the range 25 to 50%. In a well-managed building, avoidable waste levels of below 15% can be achieved (Cibse, 2000). Many studies in the past have shown that a significant fraction, as much as 30%, of energy consumption by commercial buildings is wasted (Ardehali, 2003). Additionally, using computer simulations and field measurements, EPRI estimated that change in energy consumption in the range of 10 to 35% was not uncommon due to minor adjustments in equipment and controls.

Faults in different subsystem of a HVAC system indicate that some components are not operating properly according to the design intent. They may occur at any stages of building energy system operation, including improper system design, installation and operation. The two basic fault categories, classified based on the abruptness of the occurrence: degradation and abrupt failure of components (Annex, 2009). Degradation faults happen after some time of operation and gets worse gradually over time, such as the fouling of the tubes of cooling coil.

These faults usually cannot be noticed until the degradation has exceeded a critical level. Abrupt failure faults means the equipment suddenly stops working and require immediate service to resume normal operation, e.g., a stuck damper or supply fan stop. They have more noticeable effects than degradation faults. The scope of this study includes both degradation faults and abrupt failure faults. The degradation faults are introduced and analysed with performance data at different levels of severity. Theoretically, all HVAC devices including control software and monitoring system could develop faults. Therefore, faults are categorized based on the specific device corrupted by a fault, with the devices grouped into four categories: sensor, controlled device, equipment, and monitoring system and controller. Such categories are mostly used among control engineers. Some parts of HVAC in many buildings lack system effectiveness and integration. Hardware failures, software errors, and the human factors related to the misuse of HVAC products prevent buildings from achieving energy efficiency that is expected. Improper system design, installation and operation cause building energy inefficiency. Eventually all of these faults lead to avoidable waste. Routine maintenance and commissioning are effective ways to identify system faults and create an energy-efficient building, but highly skilled engineers are needed to conduct the process, and the process is time and cost consuming.

Energy management and control systems (EMCS) are widely used in modern buildings and are composed of many types of sensors and controllers. Sensors measure temperature, flow rate, pressure, humidity, etc. and send signals to controllers, and finally to central stations. In fact, large amounts of data are available on the EMCS central station. Most EMCS systems record great amounts of data every day and store the data in databases. Through the computer network, these data are generally available online. However, modern HVAC systems and control systems have become more and more complicated. It is very difficult for an average building operator to understand the stored data directly and to conduct advanced building operation or automated fault detection and isolation (FDI) tasks without other help. This abundance of data has been described as a data rich but information poor situation and has stimulated research into better ways of examining the data.

In compare with the large amount of data gathered and processed, the state of the art of data mining and information abstraction is poor. Besides the large quantity of data, a number of problems exist with the EMCS data quality, such as missing data, erroneous values due to sensor accuracy, and faulty measurements due to sensor failure. In general, there is a need to provide building operators with tools to analyse EMCS data and to provide user-friendly information.

1.2 Aims

This thesis presents development for advanced FDI methods focus on HVAC system. The contribution is not only for one part of a HVAC system but also cover general HVAC system including heating system, cooling system, air handling unit and control system. The great developments in data communication, computing and data visualization along with decreasing costs of sensors, actuators, and controllers give an opportunity to better utilize the collected EMCS data for FDI purposes in this thesis. The objectives of FDI techniques are to automatically detect faults and isolate their causes at an early stage; and to prevent additional facility damage and energy waste. As basic definition in any FDI system fault detection in this dissertation is a process of determining whether there are faults in a HVAC system. Also fault isolation or diagnostics involves fault identification, which includes the location, significance and causes of a fault.

Most important part of developing is testing and result checking during each phase of development. Access to rich source of data is bullet point and highly recommended in any research. Those data can be generated by simulation methods or by experimental data depend on expense and accessibility of laboratories. In this study a comprehensive simulation accounted in first phase to find best match configuration of algorithms and then different tests have been allocated to each stage of development from first phase (best algorithm configuration).

The energy consumption of existing buildings could be largely decreased by performing continuous commission using FDI technologies. FDI technologies can be used to automatically identify failures in operation of HVAC equipment and systems. If FDI can identify inefficient system performance and alert building operators, the systems can be fixed sooner, thus reducing the time of operating in failure modes and saving energy while improve indoor air quality.

Extensive research has been conducted during the past decades in the FDI area to identify different technologies that are suitable for building HVAC system (Katipamula, 2005a). Physical redundancy, heuristics or statistical bands, including control chart approach, pattern recognition techniques, and innovation-based methods or hypothesis testing on physical models are usually used to detect faults. Information flow charts, expert systems, semantic networks, artificial neural network, and parameter estimation methods are commonly used to isolate faults. Heuristic rules and probabilistic approaches are used to evaluate faults.

While different methodologies have been developed quality of core algorithms have never been focussed in parallel. This study has been focussed on quality of responses against different range of fault detection algorithms. Combination of various FDI algorithms have been reviewed and tested to identifying of quality response with focus on HVAC system behaviour. These configurations mostly chosen based on statistical, intelligent algorithm and random optimization based on genetic algorithm and particle swarm. An infrastructure for experimental test was built up by focussing on such numerical applicable algorithms. Some initial data have been generated to checking for performance of different configuration of algorithms and then best match configuration choose to develop in quality of detection and isolation.

However, while FDI is well established in the process control and other industries, it is still not widely used in HVAC systems. Current FDI methods developed for HVAC systems often require extensive training data and high data quality. Moreover, unsatisfactory false alarm rate and the lack of good FDI strategies for degrading faults and fault diagnosis also prevent the HVAC industry from embracing FDI strategies. With the development of sensor and computer technologies, massive amounts of real-time measurements provide the chance for data-driven FDI methods, which have already received increased attention recently. Another imperative need is to efficiently evaluate different FDI technologies and products, which is not an easy task, and is well appreciated by professionals in this area (Reddy, 2007).

To assist in the development and evaluation of chiller system FDI methods, ASHRAE 1043-RP (Bendapudi, 2002) produced several experimental data sets of chiller operation under fault-free as well as faulty data (under different faults and four severity levels for each degrading fault) as well as a dynamic simulation model for centrifugal chillers. However, only limited experimental studies under restrictive scope are available to evaluate whole HVAC FDI methods. A dynamic HVAC simulation model that is capable of producing fault free and faulty operation data for commonly used all HVAC sections configurations and control & operation strategies is thus needed. Such dynamic HVAC model needs to be properly validated with experimental data for both fault-free and faulty operation before any credibility can be placed on their prediction accuracy and usefulness. Moreover, experimental data that can aid in HVAC FDI development and evaluation are also needed.

In fact this thesis tried to develop an applicable FDI algorithm to cover a generic HVAC system not based on different suppliers. One of most bullet points of this study is experimental test of each stage and configuration to make enough reality of final algorithm.

This reality and reliability could earn from HVAC laboratory of UTS that given enough data in different outdoor set points.

Finally the aims of this study can be outlined as:

1. Develop a comprehensive simulation for general HVAC system not based on vendor's specification based on continuity laws.
2. Best configuration find of FDI algorithms based on testing with simulation data source to make high compatibility with different sections of HVAC system.
3. Developing of new light learning technique based on statistical in core of online intelligent methods with minimum time cost and high accuracy.
4. Test developed methods with experimental data to increase reliability of algorithm in real case study.
5. Applicability of developed algorithm as part of monitoring and control section in industrial HVAC system to prior fault finding as well as fault isolation.
6. Increase safety in HVAC system by fault prior finding to active related alarm in monitoring system.
7. Decrease of cost both in operator and service as well as devices that might be damaged by faulty section.

1.3 Thesis Contributions

Based on the above discussion, there are needs for 1) a dynamic HVAC simulation model that can simulate both fault free and faulty operation data and reviews efficiency of different FDI methods based on fault analysing; then 2) develop better FDI methodologies for applying on all sections of HVAC system. Therefore, in this study, the first contribution is to develop a dynamic simulation model of a HVAC system that:

- is based on first conversion principles of mass and heat;
- can capture the characteristics of time varying measurements and outdoor effects;
- can be used to study the impact of common faults that occur in such systems.

The second contribution of the study is to obtain experimental data based on choosing appropriate data by assistance of fault analysis simulation from HVAC model. That data can be used to develop and validate a HVAC FDI method based on new development. Installation of any new sensors and choosing the best data are based on the mathematical model simulation and fault analysis in previous stage. Existing experimental data will firstly be identified and examined. If existing data are not sufficient, experiments will be designed and implemented under both normal operating conditions and under known faulty conditions.

The third contribution is to develop a practical FDI technique for HVAC system that:

- can be developed based on historical measurement data;
- is affordable and efficient for different parts of HVAC system;
- is robust under varying operating conditions;
- is capable of handling different types of faults, including sensor or process faults, abrupt or degradation faults;

As we can see in figure 1.1 three objects cover this study based on simulation, experiment and FDI methodology development. The simulation model generated from Objective 1 provides infrastructural data for fault analysis and then proper experimental data from objective 2. This objective also prepares fault free and faulty data to check the best algorithm for FDI system from objective 3. The objective 2 is also used for development of the proposed FDI method in real case study from Objective 3. Compared with experimental data, the simulation model produces operational data under reproducible and easy to configure conditions. The simulation results from Objective 1 give best idea for install efficient measurement devices and sensors in the experimental rig based on the fault sensitivity analysis. Furthermore, objective 1 helps for pre-assessing of possible FDI algorithms from objective 3 with consideration of minimum cost effect for test in wide range of artificial faults. Objective 2 also finalize the power of proposed algorithm for FDI in a real case study with experimental data. Totally, an experimental rig is set upped in the HVAC laboratory of UTS and a range of operational data set, including AHU, Chiller and cooling tower are generated for both fault free (healthy) and faulty situation.

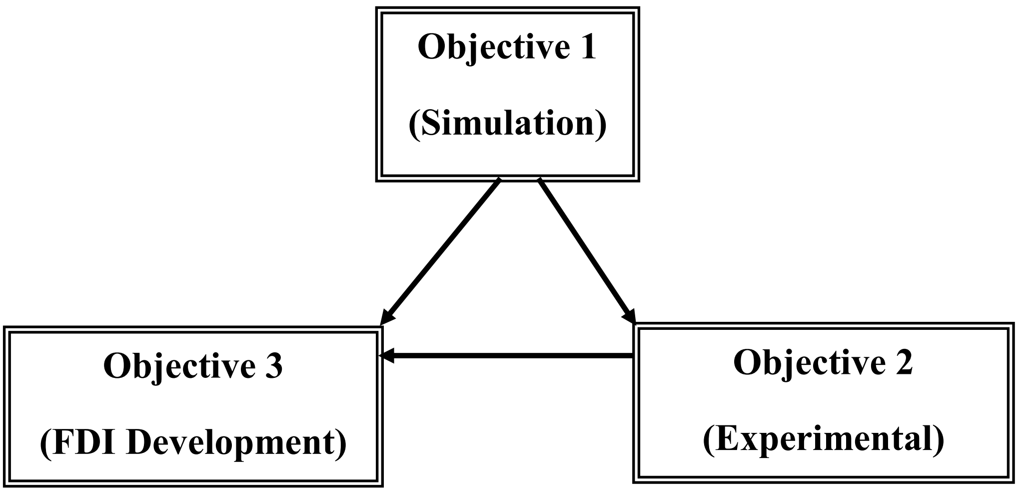


Figure 1.1 Flow chart relationships among objectives for this research

1.4 Publications

D. Dehestani, S. Su, H. T. Nguyen, Y. Guo, Robust Fault Tolerant Application for HVAC System Based on Combination of Online SVM and ANN Black Box Model, European Control Conference 2013 (ECC13), July 2013, 978-3-952-41734-8/©2013 EUCA

D. Dehestani, HVAC Model Based Fault Detection by Incremental Online Support Vector Machine, International Journal of Computer Theory and Engineering (IJCTE), Vol. 5, No. 3, pp 472-476, June 2013.

D. Dehestani, J. Madadnia, H. Koosha, F. Eftekhari, Comprehensive Analysis for Air Supply Fan Faults Based on HVAC Mathematical Model, Advanced Materials Research (Volumes 452 - 453), PP 460-468, 2012.

D. Dehestani, S. Su, H. T. Nguyen, Y. Guo, Experimental sensitivity analysis for Heat Ventilating and Air Conditioning (HVAC) system by Feed Forward Neural Network (FFNN) model application, 10th International Conference of Healthy Buildings, Brisbane, 2012.

D. Dehestani, S. Su, H. T. Nguyen, Y. Guo, Intelligent model based fault detection of HVAC cooling tower by chiller model sensitivity based on ANN model and SVM classifier, 10th International Conference of Healthy Buildings, Brisbane, 2012.

D. Dehestani, S. Su, H. T. Nguyen, S.H. Ling, Y. Guo, Intelligent Fault Detection and Isolation of HVAC System Based on Online Support Vector Machine, 3rd International Conference on Machine Learning and Computing (ICMLC), Vol. 3, pp104-108, February, 2011.

D. Dehestani, S. Su, H. T. Nguyen, S.H. Ling, Y. Guo, Online Support Vector Machine Application for Model Based Fault Detection and Isolation of HVAC System, International Journal of Machine Learning and Computing (IJMLC), volume 1, No. 1, pp. 66-72, April 2011.

D. Dehestani, J. Madadnia , V. Vakiloroaya, F. Eftekhari, Comprehensive Analysis for Air Supply Fan Faults Based on HVAC Model, International Conference on Management, Manufacturing and Materials Engineering (ICMMM 2011), Nov. 2011.

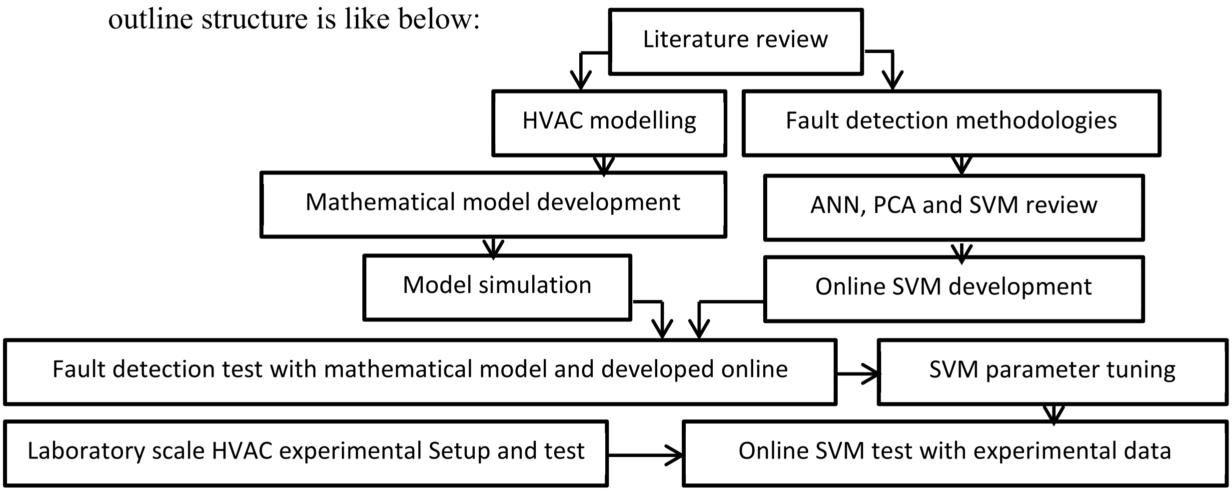
D. Dehestani, V. Vakiloroaya , J. Madadnia, M. Khatibi, Design Optimization of the Cooling Coil for HVAC Energy Saving and Comfort Enhancement, International Conference on Management, Manufacturing and Materials Engineering (ICMMM 2011), Nov. 2011.

Book Chapter
D. Dehestani, S. H. Ling, S. Su, H. T. Nguyen, Y. Guo, Computational Intelligence and Its Applications: Evolutionary Computation, Fuzzy Logic, Neural Network and Support Vector Machine Techniques, Imperial College Press, London, ISBN-13: 9781848166912

1.5 Structure of Thesis

In this study, a mathematical HVAC dynamic model that is capable of generating both fault free and faulty operational data was developed and validated based on various outdoor situations and different control tuning. A fault sensitive analysis developed founded on this simulation for different range of faults. Extensive experiments were designed and conducted in three different days to generate operational data for a large variety of common HVAC faults. The existing experimental data were analysed to identify and summarize fault symptoms associated with common HVAC faults. Two different methods including principle component analysis (PCA) and support vector machine (SVM) were reviewed based on simulation and experimental data. Finally, a new FDI methodology that utilized the artificial neural network (ANN) fault free model and incremental learning technique in combination with online SVM was developed and evaluated using experimental and simulated data for fault detection. In this thesis, Chapter 2 supplies a literature review about methodology on FDI and FDI on HVAC system. In Chapter 3, the foundation of a HVAC system and mathematical model which govern on HVAC system are introduced briefly.

Chapter 4 introduces important faults and their patterns for a HVAC system then follows sensitivity analyse of HVAC system to these patterns based on simulation of mathematical model from chapter 3. Chapter 5 reviews the details of two powerful algorithms of PCA and SVM for fault detection based on their efficiency in fault detection and isolation. Finally, an incremental learning algorithm combined with online SVM developed for FDI system in top of ANN model extension for fault free model. Chapter 6 initially introduces the HVAC experimental setup and recording data in first stage and second part reviews the performance of extended FDI application from chapter 5 on the extracted experimental data. Finally, conclusions, contributions, and recommendations are given in Chapter7. The outline structure is like below:



Chapter 2

2 Literature Review

There are many approaches for fault detection on the systems which make confusion for first time of application. Categorized different type of fault detection methods and general review is basic way to study of fault detection algorithms. General review of fault detection systems as well as brief comparison applied in this chapter. Most research relating to fault detection and isolation on refrigerators, air conditioners, chillers and air-handling units (AHUs) which represent most of the HVAC research completed to date have been reviewed. Furthermore, Support Vector Machine (SVM) algorithm as one of the new powerful classification method is appraised in this chapter in a simple way. This section will shed the light on the advancement and achievement in literature and the shortages from the current available approaches that this research aims to overcome.

2.1 Methodology on Fault Detection and Isolation

The purpose of this section is to review the common methods for fault detection. In the past decade, considerable research has been devoted to the area of system fault detection and isolation (FDI).

FDI methods can be classified into two broad categories, model based methods and data based methods. The two categories differ by the knowledge used to diagnose the cause of faults, although both may use simulation models and measurement data. Model based methods use "prior knowledge" (knowledge available in advance) to identify the differences between model simulation results and actual operation measurements. Simulation models are commonly based on first principles and do provide process insight. However, they may not fit the process data that well and are not able to explain systematic variation. Data based methods may not use any physical knowledge; instead, they can be driven completely by recorded measurement data. These data driven models fit the data properly, but cannot be generalized to different situations and do not always generate good process insight.

Model based methods are further divided into quantitative and qualitative modelling methods. Quantitative models are based on mathematical relationship derived from the underlying physical knowledge. Quantitative methods rely on explicit mathematical models of a system to detect and diagnose faults. By understanding the physical relationships and characteristics of a system, such as HVAC system, mathematical equations to represent each component of the system can be developed and solve to simulate the steady and transient behaviour of the systems. Another broad method is qualitative modelling, which uses rule based methods developed based on prior knowledge. Qualitative models use the qualitative rule relationships to detect and diagnose faults instead of quantitative mathematical equations. The rules are derived from expert knowledge, process history data and quantitative models simulation data. Expert knowledge is normally summarized to a database in the form of if-then statements.

Data based models are derived from process history data, and are subdivided into black box model and grey box model. Their difference is whether model parameters have physical meaning. Black box models use non-physical based relationship to represent the characteristics of a system. Model parameters do not represent actual physical properties. Black box models use techniques such as linear or multiple linear regression, artificial neural networks, and fuzzy logic. In a grey box model, the model parameters are determined based on physical principles. Parameter estimation techniques are often used to obtain those parameters from measurement data. Comparing with black box modelling, grey box modelling needs higher-level user expertise to form the model parameters and estimate parameter values. An overview on general FDI concepts and gave a chart for classification of diagnostic algorithms (Figure 2.1) presented before (Venkatasubramanian, 2003a) (2006b) (2006c).

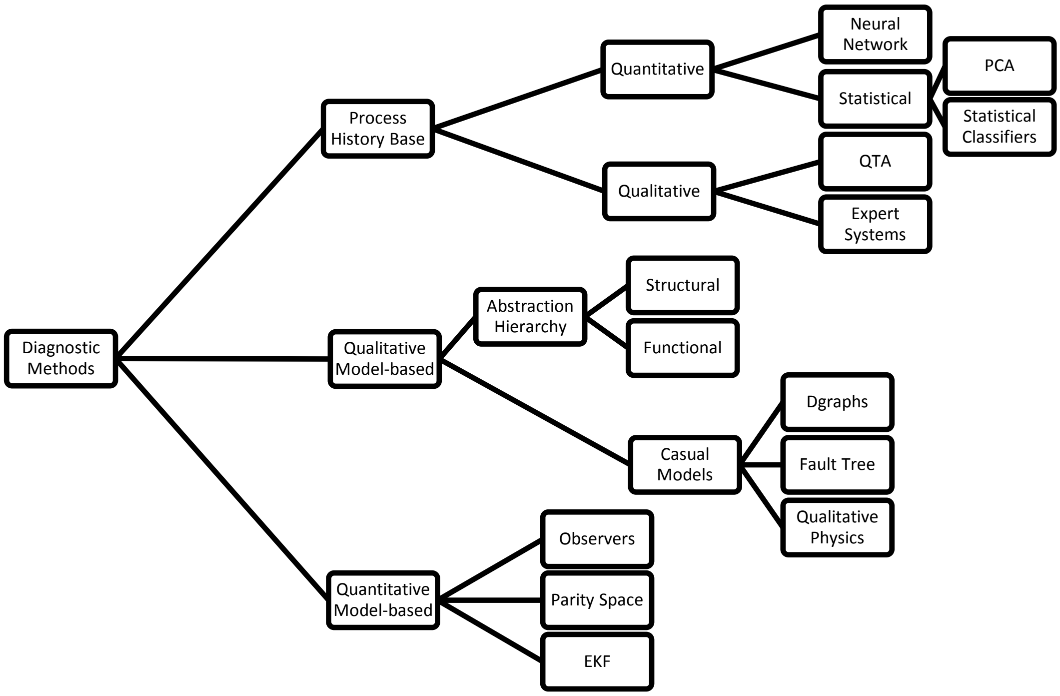


Figure 2.1 Classification of diagnostic algorithms (Venkatasubramanian, 2003a)

FDI methods are broadly classified into three general categories, quantitative model based methods, qualitative model based methods, and process history based methods. This category provided a classification in terms of the manner in which these methods approach the problem of fault diagnosis. These disparate methods were compared and evaluated for a common set of desirable characteristics that one would like the diagnostic systems to process. In this study, we start with quantitative model base for the first attempt and then focus on FDI system that is built on process history data.

This classification of diagnostic algorithms in Figure 2.1 can help us to narrow the optional FDI methods further. PCA/PLS, statistical classification and neural networks methods are most promising one. Finally, a combination of neural network for healthy model with statistical classifier method as the basic fault detector attracts our attention for its capability that the development of new intelligent statistical classifier such as SVM can apply in an online algorithm for update of model based on new data.

2.1.1 Model-Based Methods

The model-based fault detection can be broadly classified as qualitative or quantitative. The model is usually developed based on some fundamental understanding of the physics of the process. In quantitative models this understanding is expressed in terms of mathematical functional relationships between the inputs and outputs of the system. In contrast, in qualitative model equations these relationships are expressed in terms of qualitative knowledge about a process. Typical qualitative models are causal models and abstraction hierarchies. Details are given in the following.

2.1.1.1 Quantitative Model-Based Methods

Most quantitative model-based methods are residual-based. Relying on an explicit model of the monitored plant, these model-based FDI methods require two steps. The first step generates inconsistencies between the actual and expected behaviour. Such inconsistencies, also called residuals, reflect the potential faults of the system. The second step chooses a decision rule for diagnosis.

There exists a wide variety of residual based approaches for linear systems, e.g. the observer-based approach, the parity space approach, and the parameter estimation approach. There exist some model-based approaches that do not count on the residual for the indication of faults. One representative example is based on the use of multiple models (MM). It runs a bank of filters in parallel, each based on a model matching the possible system structures due to different failures. In non-interacting MM, the single-model-based filters are running in parallel without mutual interaction. Such an approach is quite effective in handling problems with an unknown structure or parameter but without structural or parametric changes. However, the problem of FDI does not fit well into such a framework because, in general, the system structure or parameter does change as a component or subsystem fails.

A notable recent advance in MM is the development of the interacting multiple-model (IMM) estimator (Zhang, 2008). By comparison, IMM can overcome the above weaknesses of the non-interacting MM approach by explicitly modelling the abrupt changes of the system by "switching" from one model to another in a probabilistic manner. In IMM, the model probabilities are used as an indication of a failure because it provides a meaningful measure of how likely each fault mode is at a given time.

Quantitative model-based methods have some desirable characteristics. If one has complete knowledge of all inputs and outputs of the system, including all forms of interactions with the environment, fault diagnosis would be a well-defined problem regardless of the number of faults present. On the other hand, if there is only a single sensor indicating whether the system is normal or faulty, then nothing can be diagnosed including the proper functioning of the sensor itself. The effectiveness of any diagnostic procedure is limited by the availability of sensor information (Aravena, 2002).

A crucial need in the model-based approach is to state the significance of the observed changes with respect to the noise, unknown inputs which cannot, in any reasonable way, be modelled as random processes with known statistics. This is the general limitation of all the model-based approaches that have been developed so far. One of the popular ways of doing this is the method of disturbance decoupling. In this approach, all uncertainties are treated as disturbances and filters are designed to decouple the effect of faults and unknown inputs so that they can be differentiated (Chen, 2007), (Viswanadham, 2008).

The other alternative is that the FDI problem has been addressed from a statistical point of view, with faults modelled as deviations in the parameter vector of a stochastic system. Fault detection and isolation have been stated as hypotheses testing problems (Basseville, 2002). The key feature of this method is its ability to handle noises and uncertainties, to reject nuisance parameters and to select one among several hypotheses. First of all, FDI problems in dynamic systems are reduced to the universal static problem of monitoring the mean value of a Gaussian vector through the help of a convenient residual generation. Then different hypotheses testing methods are investigated for FDI (Basseville, 2002). Moreover, the types of models the analytical approaches can handle are limited to linear and some very specific nonlinear models. For a general nonlinear model, linear approximations can prove to be poor and the effectiveness of these methods might be greatly reduced. When a large-scale process is considered, the size of the bank of filters can be very large increasing the computational complexity.

2.1.1.2 Qualitative Model-Based Methods

The qualitative models can be developed either as qualitative causal models or abstraction hierarchies. Figure 2.1 shows the taxonomy of domain knowledge based on these two broad categories. In casual models, the cause-effect relations can be represented in the form of signed diagraphs. Causal models are a very good alternative when the quantitative models are not available but the functional dependencies are understood. Another form of model knowledge is through the development of abstraction hierarchies based on decomposition. The idea of decomposition is to be able to draw inference about the behaviour of the overall system solely from the laws governing the behaviour of its subsystems. Abstraction hierarchies help to quickly focus the attention of the diagnostic system to problem areas.

One of the advantages of qualitative methods based on deep knowledge is that they can provide an explanation of a fault propagation path. This is indispensable when it comes to decision-support for operators. They can also guarantee completeness in that the actual fault will not be missed in the final set of faults identified. However, they suffer from the resolution problems resulting from the ambiguity in qualitative reasoning. When quantitative information is partially available, one could use the order-of-magnitude analysis or interval-calculus to improve the resolution of purely qualitative methods (Aravena, 2002).

In the case of the qualitative model-based approaches, the combinatorial complexity is unavoidable and can only be partly alleviated with an efficient search (Reiter, 2008). Because of many multiple fault combinations, the search for multiple faults by specifying them explicitly as different classes and obtaining training patterns for them is not feasible. From an industrial application viewpoint, the majority of fault diagnostic applications in process industries are based on model free or process history based approaches. This is due to the fact that process history based approaches are easy to implement, requiring very little modelling effort and prior knowledge. Further, even for processes for which models are available, the models are usually steady-state models. It would require considerable effort to develop dynamic models specialized towards fault diagnosis applications.

2.1.2 Model-Free Methods

Unlike the model-based approaches where a priori knowledge about the system is needed, in model-free methods, only the availability of the large amount of historical data is needed. They are also known as the black box approach. In this research, our goal is to develop global HVAC health indicators that do not rely on mathematical models yet are capable of detecting process malfunctions. There are different ways in that data can be transformed and presented as a priori knowledge to a detection system. This is known as feature extraction.

In terms of feature extraction, model-free methods can be either qualitative or quantitative in nature. Two of the major methods that extract qualitative history information are the expert systems and trend modelling methods. Methods that extract quantitative information can be non-statistical or statistical methods. Neural networks are an important class of non-statistical classifiers. Nowadays data mining is one of the most active research fields. The key advantage of data mining-based fault detection is that it can automatically generate concise and accurate detection models from large amounts of data.

2.1.2.1 Qualitative Feature Extraction

2.1.2.1.1 Expert Systems

The main advantages in the development of expert systems for diagnostic problem solving are: ease of development, transparent reasoning, ability to reason under uncertainty and the ability to provide explanations for the solutions provided.

There are a number of researchers who have worked on the application of expert systems for diagnostic problems. Becraft (2003) has proposed an integrated framework comprising of a neural network and an expert system. A neural network is used as a first-level filter to diagnose the most commonly encountered faults in chemical process plants. Once the faults are localized within a particular process by the neural network, a deep knowledge expert system analyses the result, and either confirms the diagnosis or else offers an alternative solution. Tarifa (2007) has proposed a hybrid system that uses signed directed graphs (SDG) and fuzzy logic. The SDG model of the process is used to perform qualitative simulation to predict possible process behaviours for various faults. Those predictions are used to generate (if-then) rules that are evaluated by an expert system using information about the actual process state and fuzzy logic.

There are two types of methods for modelling knowledge for an expert system. They are shallow knowledge and deep knowledge (Chung, 2010). Shallow knowledge expert systems use (if-then) type rules as the primary means of knowledge representation. These rules are formulated based on a large collection of empirical observations. In cases where the failure modes are not well known (e.g.: some faults are unanticipated and have very low probability of occurring), these systems are inadequate, and deep knowledge systems are more appropriate. When confronted with an unfamiliar problem an expert can resort to “first principles”. Through an in-depth understanding of the problem, an expert can resolve problems that have not been well documented by prior observation. In this situation, the knowledge used by the expert is referred to as “deep knowledge”. This approach provides a broader knowledge base, as well as modularity for incorporating new knowledge. However, in all applications, the limitations of an expert system approach are obvious. The expert-based fault detection system fails to generalize and detect new faults without known signatures. Knowledge-based systems developed from expert rules are very system-specific. Their representation power is quite limited, and they are difficult to update (Rich, 2005).

2.1.2.1.2 Trend Analysis

A second approach to qualitative feature extraction is the abstraction of trend information. For tasks such as diagnosis, qualitative trend representation often provides valuable information that facilitates temporal reasoning about the processes behaviour. In a majority of cases, process malfunctions leave a distinct trend in the sensors monitored. These distinct trends can be suitably utilized in identifying the underlying abnormality in the process. Thus, a suitable classification and analysis of process trends can detect the fault earlier and lead to quick control. Some papers and projects have shown that trend modelling can be used to explain the various important events happening in the process, do malfunction diagnosis and predict future states. The following is an overview of some trend analysis methods and applications.

• ***Triangular Episodic Presentation***

Cheung (2002) has built a formal framework for the representation of process trends. A language called triangular episodic representation is formulated and used in trend extraction. It is based on temporal episodes modelled geometrically as triangles to describe the local temporal patterns in data and introduces triangulation to represent trends. Triangulation is a method where each segment of a trend is represented by its initial slope, its final slope (at each point, or critical point of the trend) and a line segment connecting the two critical points. A series of triangles constitutes a process trend. Through this method, the actual trend always lies within the bounding triangle which illustrates the maximum error in the representation of the trend.

• ***Wavelets***

Vedam proposed a wavelet theory based nonlinear adaptive system for identification of trends from sensor data named W-ASTRA and later proposed dyadic B-Splines-based trend analysis. It uses the concept of multi-resolution analysis in the neural network input. Sensor data is projected onto scaling functions at different levels. First of all, the coefficients from the highest level are used to identify the primitives. If a unique primitive identification is possible then the next set of samples is collected or else the coefficients from the next lower level are used. Then W-ASTRA compare the sensor trends with their fault signature which is the segment of its trend that characterizes its behaviour for a given fault class.

• ***Qualitative Temporal Shape Analysis***

Konstantinov (2009) proposed a generic methodology for qualitative analysis of the temporal shapes of process variables with the help of an expandable shape library that stores shapes like decreasing concavely, decreasing convexly and so on. This procedure consists of three phases: analytical approximation of the process variable, its transformation into symbolic form based on the signs of the first and second derivatives of an analytical approximation function and a degree of certainty calculation.

The biggest challenge in applying trend analysis for FDI is how to automatically do trend extraction from noisy process data. In order to obtain signal trend not too susceptible to momentary variations due to noise, some kind of filtering needs to be employed. One may simply use a filter (such as an auto-regressive filter) with a priori chosen filter coefficients (specifying the required degree of smoothing). However these types of filters suffer from the fact that they cannot distinguish well between a transient and true instability (Gertler, 2009). The essential qualitative characters might be distorted by these filters. Avoiding this problem requires that the trend be viewed from different time scales or different levels of abstraction.

Dash (2001) proposed an interval-halving polynomial fit approach for automatic trend extraction from noisy process data. This approach parameterizes the data as a sequence of primitives with the "goodness of fit" determined with respect to noise. The interval-halving approach is a recursive method, where initially a single primitive is sought to characterize the entire data record, and when failing, the interval is halved and the process is repeated on the halved length scale until success is achieved. The procedure is recursively applied until the entire data is covered. Wavelet based de-noising is applied to remove noise. To determine the "goodness of fit", i.e., significance of error, they use the estimate of noise provided by the wavelet analysis.

2.1.2.2 Quantitative Feature Extraction

2.1.2.2.1 Neural Network

In general, the learning strategy can be classified into supervised and unsupervised learning. In supervised learning strategies, by choosing a specific topology for the neural network, the network is parameterized in the sense that the problem at hand is reduced to the estimation of the connection weights. The connection weights are learned by explicitly utilizing the mismatch between the desired and actual values to guide the search. This gives supervised techniques the ability to correctly identify a known error for which the symptoms are not known. The most popular supervised learning strategy in a neural network has been back-propagation.

The neural network which utilizes the unsupervised estimation technique is known as the self-organizing neural network as the structure is adaptively determined based on the input to the network, thus unsupervised learning may be used to identify new classes of errors previously not considered. Ortega, etc. (2005) constructed a neural-based diagnostic system to inspect the defects of the ropes of mining shifts automatically. A network composed of three sub-networks with error back propagation and momentum coefficient acquired the best results. Hierarchical neural network architecture for the detection of multiple faults was proposed by Watanabe (2004). Bakshi (2003) proposed Wavenet: a multi-resolution hierarchical neural network. Wavenet is an NN with one hidden layer whose basis functions are drawn from a family of orthonormal wavelets.

There are also other architectures such as self-organizing maps. There are some limitations, however, to methods that are based solely on historic process data. It is the limitation of their generalization capability outside of the training data. This problem can be alleviated by radial and ellipsoidal units by avoiding a decision in case there are no similar training patterns in that region. This allows the network to detect unfamiliar situations arising from novel faults. Besides its lack of ability to generalize to unfamiliar regions of measurement space, networks also have difficulty with multiple faults (Venkatasubramanian, 2006c). This brings out a crucial point of distinction between model based approaches and classifiers based on historic process data.

2.1.2.2.2 Data Mining – Classification

Data mining is concerned with uncovering patterns, associations, changes, anomalies, and statistically significant structures and events in data. Simply put, it is the ability to take data and pull from it patterns or deviations which may not be seen easily to the naked eye. Another term sometimes used is knowledge discovery. The recent rapid development in data mining has made available a wide variety of algorithms drawn from the fields of statistics, pattern recognition, machine learning, and database.

The key advantage of data mining-based fault detection is that it can automatically generate concise and accurate detection models from large amounts of data. The methodology itself is general, and therefore can be used to build fault detection systems for a wide variety of computing environments. Data mining techniques such as Support Vector Machines (SVM) and the Association Rule have been investigated in the context of fault detection. SVM is a relatively new type of learning algorithm. When used for classification, SVM separates a given set of binary-labelled training data with a hyperplane that is maximally distant from them (known as maximal margin hyperplane) (Witten, 2000). For cases in which no linear separation is possible, they can nonlinearly map the input vector into a high dimensional feature space where the data can be linearly classified. The hyperplane found by the SVM in feature space corresponds to a nonlinear decision boundary in the input space. Given a test instance, its distance from the hyperplane can be calculated and, following some threshold, it can be determined if the instance is anomalous. Sample applications in detecting novel data can be found in (Scholkopf, 2008; Burges, 2009).

However, as a classifier, prior knowledge for the learned domain and novel region is needed to provide a learning basis for SVM tools. There has been an increased interest in data mining-based approaches to build detection models for intrusion detection systems (IDS). These models generalize from both known attacks and normal behaviour in order to detect unknown attacks. They can also be generated in a quicker and more automated method than manually encoded models that require difficult analysis of audit data by domain experts. Several effective data mining techniques for detecting intrusions have been developed (Yairi, 2001; Lee, 2000; Eskin, 2000), many of which perform close to or better than systems engineered by domain experts.

Lee (2000) gives the idea to first compute the association rules and frequent episodes from audit data which capture the intra- and inter- audit record patterns. These patterns are then utilized, with user participation, to guide the data gathering and feature selection processes. In some cases, all positive examples are alike but each negative example is negative in its own way. Negative examples come from an unknown number of negative classes. In other cases, one class is sampled very well, while the other class is severely under-sampled. The measurements on the under-sampled class might be very expensive or difficult to obtain. The objective becomes making a description of a target set of objects and to detect which new objects resemble this training set. The difference with conventional classification is that in one-class classification only examples of one class are available. The objects from this class are called the target objects. All other objects are named the outlier objects. In the literature, a large number of different terms have been used for this problem.

The term one-class classification originates from (Moya, 2003), but also outlier detection and novelty detection (Ritter, 2007) are used. One possible approach to one-class classification is to use a density method which directly estimates the density of the target objects. By assuming a uniform outlier distribution and by the application of Bayes' rule, the description of the target class is obtained. For instance, in (Tarassenko, 2005) the density is estimated by a Parzen density estimator. In (Ritter, 2007) not only is the target density estimated, but also the outlier density. Unfortunately, this procedure requires a complete density estimate in the complete feature space. Especially in high dimensional feature space this requires huge amounts of data. Furthermore, it assumes that the training data is a typical sample from the true data distribution. In most cases the user has to generate or measure training data and one might not know beforehand what the true distribution might be. This makes the application of the density methods problematic. Alternatively, boundary methods have been developed which only focus on the boundary of the data. They try to avoid the estimation of the complete density of the data and therefore work with an uncharacteristic training data set. For the boundary methods, it is sufficient that the user can indicate just the boundary of the target class by using examples. An attempt to train just the boundaries of a data set is made in (Moya, 2006).

Neural networks are trained with extra constraints to give closed boundaries. Tax (2001) presented a new type of one-class classifier for the support vector data description. It models the boundary of the target data by a hypersphere with minimal volume around the data. The boundary is described by a few training objects, the support vectors. We develop online classification with the incremental learning technique to binary classification (normal and faulty) with SVM. Using the training data we establish standard algorithm to define its decision making platform. Measurements that fall outside the healthy data are classified as indicating a fault. When both normal and faulty data are available, we consider using SVM for binary classification due to the excellent generalization performance (accuracy on test data) in practice.

2.2 FDI on HVAC System

2.2.1 Overview

There is a large body of literature on FDI for HVAC systems, investigating various faults of the sensors, refrigerators, heat pumps, chillers and AHU system. Katipamula and Brambley (2005a), (2005b) provide a detailed review which will not be repeated here. Grimmelius et al. (2009) developed an empirical fault diagnostic system for a chiller, which combined fault detection and diagnostics in a single step. A reference model based on multivariate linear regression was developed with data from a normally operating chiller used to estimate values of process variables. Differences between actual measured values and the estimated one from the reference model are defined as residuals. Symptoms of residuals corresponding to 58 faulty conditions were collected to form a chiller faulty behaviour matrix. Scores for given symptoms were determined based on expert knowledge about the corresponding faults. Then the scores were assigned indicating the degree of symptom match in the matrix. A total score was calculated by adding the individual scores of all expected symptoms in the symptom matrix. The possible fault was determined by the normalized total score.

Peitsman and Bakker (2009) illustrated the possibilities of using a black box model for fault detection by comparing diagnostic capabilities of two types of models, a multiple input/output auto regressive exogenous (ARX) model and artificial neural network (ANN) models. Both of the two models can be categorized as black box modelling. A two-level modelling approach was applied for both ARX and ANN models. In a two-level model, system-level models were used to detect "faulty" operation and component-level models were used to diagnose the cause of the fault. The two-level system model had the advantage that the system models were easier to calculate, and hence faster in use and more suitable for real-time applications. ANN models appeared to have a slightly better performance than the ARX models in detecting faults at both the system and the component levels.

Rossi and Braun (2007) presented a method for automated detection and diagnosis of faults in vapour compression air conditioners which only required nine temperature measurements and one humidity measurement. A steady state model was used to describe the relationship between operation conditions and the expected output states under normal operation conditions. By comparing the measurements of the output states under steady state operations with those predicted by the steady state model, residuals were calculated. For fault detection, statistical properties of the residuals were used for evaluation and a set of rules based on directional changes were used to identify the unique cause of each fault. The thresholds selection of critical residuals considered the trade off between the sensitivity of the method and the false alarm rate. Following this research, Braun and Chaturvedi (2010) described the development and evaluation of an inverse model. The goals of this study were to allow model training using only a limited number of data acquired over a short period of time and make accurate prediction associated with different pre-cooling strategies. The model was a hybrid or “grey-box” approach that used a transfer function model with a simple physical representation for energy flows in the building. About two or three weeks of data were necessary.

Jia and Reddy (2012) proposed an online model based FDI method for medium to large chillers. Six process faults were identified based on five important characteristic features which were developed from fifteen variables evaluated. Because data labelled as fault free and faulty generally were not available for a field chiller, online tuning of the fault detection thresholds had to be calculated heuristically over time. Clean fault free data were required to generate the regression model for chiller fault free behaviour. The authors provided clear guidance for practical implementation of the proposed FDI methodology. The study was only limited to process faults.

Reddy (2007) proposed a general methodology for evaluating FDI methods using steady state data and identified and evaluated four multivariate model based FDI methods against some laboratory chiller performance data. All four methods evaluated belong to the same general class, namely data driven methods. The four methods included model-free fault detection with diagnosis table, multiple linear regression model with diagnosis table, Principle Component Analysis (PCA) model with diagnosis table, and linear discriminate and classification approach. The second method was identified the most promising chiller FDI tool based on sensitivity analysis on various false alarm rates.

2.2.2 FDI on Refrigerators

One of the early applications of FDD was to vapour-compression-cycle-based refrigerators (McKellar, 2007; Stallard, 2005). Although (McKellar, 2007) did not develop an FDD system, he identified common faults for a refrigerator based on the vapour-compression cycle and investigated the effects of the faults on the thermodynamic states at various points in the cycle. He concluded that the suction pressure (or temperature), discharge pressure (or temperature), and the discharge-to-suction pressure ratio were sufficient for developing an FDD system. The faults considered were compressor valve leakage, fan faults (condenser and evaporator), evaporator frosting, partially blocked capillary tubes, and improper refrigerant charge (under and over charge).

Building upon McKellar's work, (Stallard, 2005) developed an automated FDD system for refrigerators. A rule-based expert system was used with simple limit checks for both detection and diagnosis. Condensing temperature, evaporating temperature, condenser inlet temperature, and the ratio of discharge-to-suction pressure were used directly as classification features. Faults were detected and diagnosed by comparing the change in the direction of the measured quantities with expected values and matching the changes to expected directional changes associated with each fault.

2.2.3 FDI on Air Conditioners and Heat Pumps

There are many applications of FDD to air conditioners and heat pumps based on the vapour compression cycle. Some of these studies are discussed below ((Yoshimura, 1999; Kumamaru., 2001; Inatsu, 2002; Wagner, 2002; Rossi, 2005; Rossi, 2006; Breuker, 2008b; Ghiaus, 1999; Chen, 2000). Breuker and Braun (2008a) summarized common faults in air conditioners and their costs of service for various faults were estimated from service records.

Yoshimura and Ito (1999) used pressure and temperature measurements to detect problems with condenser, evaporator, compressor, expansion valve, and refrigerant charge on a packaged air conditioner. The differences between measured values and expected values were used to detect faults. Expected values were estimated from manufacturers' data, and the thresholds for fault detection were experimentally determined in the laboratory. Both detection and diagnosis were conducted in a single step. No details were provided as to how the thresholds for detection were selected. Wagner and Shoureshi (2002) developed two different fault detection methods and compared their abilities to detect five different faults in a small heat pump system in the laboratory. The five faults included abrupt condenser and evaporator fan failures, capillary tube blockage, compressor piston leakage, and seal system leakage. The first method was based on limit and trend model-based approach. In the second approach, differences between predictions from a simplified physical model and the monitored observations are transformed into useful statistical quantities for hypothesis testing. The transformed statistical quantities are then compared to predetermined thresholds to detect faults.

The two fault detection strategies were operated in parallel on a heat pump in a psychrometric room. The qualitative method was able to detect four of five faults that were introduced abruptly, while the simplified physical model-based method was successful in only detecting two faults. Because the selection of thresholds for both methods is critical in avoiding false alarms and reduced sensitivity, Wagner and Shoureshi (2002) provide a brief discussion of how to trade off diagnostic sensitivity against false alarms. Their implementation is only capable of detecting faults and does not include diagnosis, evaluation, and decision making.

Rossi (2005) described the development of a statistical rule-based fault detection and diagnostic method for air-conditioning equipment with nine temperature measurements and one humidity measurement. The FDD method is capable of detecting and diagnosing condenser fouling, evaporator fouling, liquid-line restriction, compressor valve leakage, and refrigerant leakage. In addition to the detection and diagnosis, Rossi and Braun (2006) also describe an implementation of fault evaluation. A detailed explanation of the fault evaluation method can be found in Rossi and Braun (2007). The methods were demonstrated in limited testing with a rooftop air conditioner in the laboratory.

Breuker (2007) performed a more detailed evaluation of the methods developed by Rossi (1995).The detailed evaluation relied on steady-state and transient tests of a packaged air conditioner in a laboratory over a range of conditions and fault levels (Breuker, 2008b). Seven polynomial models (ranging from first to third order) were developed to characterize the performance of the air conditioner (evaporating, condensing, and compressor outlet temperatures, suction line superheat, liquid line sub-cooling, temperature rise across the condenser, and temperature drop across the evaporator) using steady-state data representing normal (un-faulted) operations. The steady-state normal data are also used to determine the statistical thresholds for fault detection, while transient data with faults were used to evaluate FDD performance. Breuker and Braun (2008b) concluded that refrigerant leakage, condenser fouling, and liquid line restriction were detected and diagnosed before 8% reduction in capacity or COP occurred. The technique, however, was less successful in detecting evaporator fouling and compressor valve leakage. The authors also concluded that increasing the measurements from 6 (2 inputs and 4 outputs) to 10 (3 inputs and 7 outputs) and using higher order polynomial models improved the performance by a factor of two.

Ghiaus (1999) presented a bond-graph model for a direct-expansion vapour-compression system and applied it to diagnosing two faults in an air conditioner. The author states that this qualitative approach of modelling faults does not need a priori knowledge of possible faults as long as the bond model is complete and accurate.

2.2.4 FDI on Chillers

Several researchers have applied FDD methods to detect and diagnose faults in vapour-compression- based chillers; some of the studies are summarized below (Grimmelius et al. (2009); Gordon and Ng (2005), 1995; Stylianou and Nikanpour (2006); Tsutsui and Kamimura (2006); Peitsman and Bakker (2006); Stylianou (2007); Bailey (2008); Sreedharan and Haves (2001); Castro (2002)).

Table 2.1 Symptom Patterns for Selected Faults (Grimmelius et al. 1995)

Fault Modes	**Compressor Suction Pressure**	**Compressor Suction Temperature**	**Compressor Discharge Pressure**	**Compressor Discharge Temperature**	**Compressor Pressure Ratio**	**Oil Pressure**	**Oil Temperature**	**Oil Level**	**Crankcase Pressure**	**Compressor Electric Power**	**Subcooling of Refrigerant**	**ΔT Refrigerant and Cooling Water**	**ΔT Cooling Water**	**Inlet Temperature at Expansion Valve**	**Filter Pressure Drop**	**Evaporator Outlet Pressure**	**Superheat**	**ΔT Chilled Water**	**Evaporator Outlet Temperature**	**Number of Acting Cylinders**
Compressor, Suction Side, Increase in Flow Resistance	↓	→	→	→	→	↓	→	→	↓	↓	→	→	→	→	→	↑	↑	↓	→	→
Compressor, Discharge Side, Increase in Flow Resistance	↑	→	↑	→	→	↑	→	→	↑	→	→	→	→	→	→	↑	↑	↓	→	→
Condenser, Cooling Water Side, Increase in Flow Resistance	→	→	↑	→	→	→	→	→	→	↑	↓	→	↑	→	→	→	→	→	→	→
Fluid Line Increase in Flow Resistance	→	→	→	→	→	→	→	→	→	↓	→	→	→	↓	→	→	↑	↑	→	→
Expansion Valve, Control Unit, Power Element Loose from Pipe	↑	→	→	→	→	↑	→	→	↑	↑	→	→	→	→	→	↑	↓	↑	→	→
Evaporator, Chilled Water Side, Increase in Flow Resistance	↓	→	→	→	→	↓	→	→	↓	↓	→	→	→	→	→	↓	↑	↑	→	→

Comstock et al. (1999) and Reddy et al. (2001) provide a detailed review of FDD literature relating to chiller systems up to their respective times. Comstock et al. (2002) presented a list of common chiller faults and their impacts on performance.

Grimmelius et al. (2009) developed a fault diagnostic system for a chiller, in which fault detection and diagnostics are carried out in a single step. The FDD method uses a reference model based on multivariate linear regression that was developed with data from a properly operating chiller to estimate values for process variables for a healthy (un-faulted) chiller. These estimates are subsequently used to generate residuals (i.e., differences between actual measured values and the values from the reference model).

Patterns of these residuals are compared to characteristic patterns corresponding to faulted conditions, and scores are assigned indicating the degree to which the patterns match the pattern corresponding to each fault mode. Fault modes with good fits (high scores) are judged as probably existing in the chiller. Fault modes with poor fits (low scores) are judged as unlikely to exist in the chiller, and faults with intermediate scores are labelled as possibly existing. Twenty different measurements are used including temperatures, pressures, power consumption, and compressor oil level. In addition to the measured variables, some derived variables, such as liquid sub-cooling, superheat, and pressure drop, are used. The inputs to the model also include the outdoor ambient temperature and load conditions.

Table 2.2 Scoring of Fault Modes for a Highly Idealized Example

Fault Mode/ Score	**Symptom 1**	**Symptom 2**	**Symptom 3**	**Symptom 4**	**Total Score**	**Normalized Score**
F1	10	10	10	10	40	1.0
Score	↓	→	↓	↑		
F2	0	9	0	3	12	0.3
Score	↑	→	↑	→		
Measurement Based Pattern	↓	→	↓	↑		

To identify potential fault modes, the chiller is classified into seven components: compressor, condenser, evaporator, expansion valve, liquid line immediately downstream of the condenser and including a filter drier, liquid line with solenoid and sight glass between the other liquid line and the evaporator, and the crankcase heater. Fault modes are associated with any component that is serviceable, which leads to 58 different fault modes. A cause and effect study of the 58 fault modes helped establish the expected influence of the faults on the components, measured variables, and subsequent chiller behaviour. Symptoms are defined as a difference in any measured or derived variable from its expected value for normal un-faulted operation (i.e., the value given by the reference model).

Symptoms associated with all 58 fault modes were generated and arranged into symptom patterns. Fault modes having identical symptom patterns were aggregated into a single fault mode, reducing the total number of fault modes from 58 to 37. These symptom patterns are arranged in a symptom matrix as shown in Table 2, with each row giving the symptom pattern associated with a particular fault. A symptom (cell in the matrix) shown by an arrow pointing up, ↑, indicates a value for the variable greater than that given by the reference model. Likewise, an arrow pointing down, ↓, indicates a symptom corresponding to a value for the variable less than the value from the reference model, and a horizontal arrow, →, indicates the fault has no effect on the corresponding variable. To diagnose a fault, a symptom pattern corresponding to a set of measurements is compared to the symptom patterns for all of the fault modes. Scores are assigned to each fault mode indicating the probability that its symptom pattern matches the measured symptom pattern as follows.

For each fault mode, each symptom is compared to its corresponding measured symptom and assigned a score between 0 and 10. If the symptom for the fault mode matches the measured symptom very well, it is assigned a high score (close to 10). If it weakly matches, it is assigned a score around 5, and if it does not match well at all, it is assigned a score close to zero. A total score for each fault mode is generated by adding the individual scores of all symptoms and dividing the total by the maximum possible score per pattern (i.e., the number of symptoms in the pattern multiplied by 10) to obtain a normalized score. These normalized scores are then classified into three categories. A normalized score of 0.9 or higher indicates a probable fault, a score between 0.5 and 0.9 indicates a possible fault, and scores lower than 0.5 indicate that the fault is likely not present.

A highly simplified example is shown in Table 2. Symptom patterns for two faults, F1 and F2, are shown along with a symptom pattern derived from measurements. Each pattern consists pattern based on how well the symptom shown in the symptom matrix corresponds to the symptom based on measurements. For example, Symptom 1 for fault mode F1 corresponds identically to Symptom 1 in the pattern derived from measurements, so it is assigned a score of 10. The normalized scores in this example lead to the conclusion that fault F1 with a score of 1.0 probably exists in this system and fault F2 with a score of 0.3 is likely not present. In actual implementation, this methodology accounts for uncertainty in measurements by establishing threshold bands around numerical values of measured and derived variables and using the proximity to them in assigning scores to symptoms. The exact algorithm for assigning numerical scores, however, is not available in the paper.

Although the method proved effective in identifying faults in systems before the chiller system failed completely, faults with only a few symptoms tended to get high scores more often. Because the reference model is a simple regression model developed with data from a specific test chiller, the same model cannot be used on other chillers but instead new models would need to be developed for each chiller. Nonetheless, this generic approach provided a foundation for diagnostic work that followed. Stylianou and Nikanpour (2006) used the universal chiller model developed by Gordon and Ng (2005) and the pattern matching approach outlined by Grimmelius et al. (2009) as part of their FDD system. Like Grimmelius et al. (2009), Stylianou and Nikanpour also perform detection and diagnosis in a single step. The methods used in the FDD system included a thermodynamic model for fault detection and pattern recognition from expert knowledge for diagnosis of selected faults. The diagnoses of the faults are performed by an approach similar to that outlined by Grimmelius et al. (2009). Seventeen different measurements (pressures, temperatures, and flow rates) were used to detect four different faults: refrigerant leak, refrigerant line flow restriction, condenser water-side flow resistance, and evaporator water-side flow resistance.

The FDD system is subdivided into three parts: one used to detect problems when the chiller is off, one used during chiller start-up, and one used at steady-state conditions. The off-cycle module is deployed when the chiller is turned off and is primarily used to detect faults in the temperature sensors. The temperature sensor readings at different locations on the system are compared to one another after the chiller is shut down and reaches steady state (under the assumption that the temperature of refrigerant will reach equilibrium conditions and reach the ambient state when the chiller is shut down overnight). The differences are then compared to the difference observed during commissioning (if the sensors are calibrated during commissioning, the differences should be zero). The monitored rate of change of a sensor value is used to check whether a particular sensor has reached steady state or not before comparing measurements across sensors.

Table 2.3 Fault Patterns Used in the Diagnostic Module (Stylianou 1996)

Fault Modes	**Discharge Temperature**	**High Pressure Liquid Line Temperature**	**Discharge Pressure**	**Low Pressure Liquid Line Temperature**	**Suction Line Temperature**	**Suction Pressure**	**ΔT_{cond}**	**ΔT_{Evap}**
Restriction in Refrigerant Line	↑	↓	↓	↓	↑	↓	↓	↑
Refrigerant Leak	↑	↓	↓	↓	↑	↓	↓	↑
Restriction in Cooling Water	↑	↑	↑	↓	↓	↓	↑	↓
Restriction in Chilled Water	↑	↓	↓	↓	↓	↓	↓	↓

The start-up module is deployed during the first 15 minutes after the chiller is started. The module uses four measured inputs (discharge temperature, crankcase oil temperature, and refrigerant temperatures entering and leaving the evaporator) scanned at five-second intervals to detect refrigerant flow faults, which are easier to detect before the system reaches steady state. To detect faults, the transient trends in measured variables during start-up are compared to the baseline trend from normal start-up. For example, a shift (in time or magnitude) in the peak of the discharge temperature may indicate liquid refrigerant flood back, refrigerant loss, or a refrigerant line restriction. Because ambient conditions affect the baseline response, the baseline response has to be normalized before a comparison is made.

The steady-state module is deployed after the chiller reaches steady state (steady-state condition is established by monitoring the rate of change of the sensor values just as in off-cycle analysis) and stays deployed until the chiller is turned off. In this mode, the module performs two functions: (1) verifies performance of the system and (2) detects and diagnoses selected faults. Performance is verified using the thermodynamic models developed by Gordon and Ng (2005). For fault diagnostics, linear regression models are used to generate estimates of pressure and temperature variables that are then compared to actual measurements in an approach similar to that described by Grimmelius et al. (2009). The estimated variables are compared to the measured values, and the residuals are matched to predefined patterns corresponding to the various faults using a rule-base (as shown in Table 3).

Although Stylianou and Nikanpour (2006) extended the previous work of Gordon and Ng (2005) and Grimmelius et al. (2009), their evaluation of the FDD systems was not comprehensive and lacked several key elements including sensitivity and rate of false alarms. In addition, it is not clear whether the start-up module can be generalized easily. Stylianou (2007) replaced the rule-based model used to match the patterns shown in Table 3 with a statistical pattern recognition algorithm. This algorithm uses the residuals generated from comparison of predicted (using linear regression models) and measured pressures and temperatures to generate patterns that identify faults. Because this approach relies on the availability of training data for both normal and faulty operation, it may be difficult to implement in the field. Only limited testing of the method was presented in the paper.

Tsutsui and Kamimura (2006) developed a model based on a topological-case-based reasoning (TCBR) technique and applied it to an absorption chiller. Case-based reasoning is a knowledge-based problem-solving technique that solves new problems by adapting old solutions. It is based on defining neighbourhoods that provide the needed measure of similarity between cases. In contrast, TCBR defines "the neighbourhood theoretically, based on the assumption that the input/output relationship is locally continuous" (Tsutsui, 2006). Tsutsui and Kamimura (2006) also compared the diagnostic capabilities of TCBR with a linear regression model. The authors state that although the linear regression model had a better overall modelling error (mean error) than the TCBR model, the TCBR model was better at identifying abnormal conditions.

Peitsman and Bakker (2006) used two types of black-box models (ANN and ARX) to detect faults in the system and at the component level of a reciprocating chiller system. The inputs to the system models included condenser supply water temperature, evaporator supply glycol temperature, instantaneous power of the compressor, and flow rate of cooling water entering the condenser (for the ANN only). The choice of the inputs was limited to those that are commonly available in the field. Using these inputs with both the ANN and ARX models, 14 outputs were estimated. For the ANN models, inputs from the current and the previous time step and outputs from two previous time steps were used. Peitsman and Bakker (2006) compared diagnostic capabilities of two types of models a multiple input/output ARX model and ANN models. They used a two-level approach in which system-level models were used to detect "faulty" operation and component-level models were used to diagnose the cause of the fault. They developed 14 system-level models and 16 component- level models to detect and diagnose faults in a chiller; however, only one example (air in the system) is described in their paper. ANN models appeared to have a slightly better performance than the ARX models in detecting faults at both the system and the component levels.

The authors also note that it is critical to find a global minimum when using ANN models. If an incorrect initial state is chosen, it may lead to a local minimum rather than the global minimum. Bailey (2008) also used an ANN model to detect and diagnose faults in an air-cooled chiller with a screw compressor. The detection and diagnosis were carried out in a single step. The faults evaluated included refrigerant under- and overcharge, oil under- and overcharge, condenser fan loss (total failure), and condenser fouling. The measured data included superheat for heat exchanger circuits 1 and 2, sub-cooling from circuits 1 and 2, power consumption, suction pressure for circuits 1 and 2, discharge pressures for circuits 1 and 2, chilled water inlet and outlet temperatures from the evaporator, and chiller capacity. Each heat exchanger circuit has its own compressor. The ANN model was applied to normal and "faulty" test data collected from a 70-ton laboratory air-cooled chiller with screw compressor.

Sreedharan and Haves (2001) compared three chiller models for their ability to reproduce the observed performance of a centrifugal chiller. Although the evaluation was meant to find the most suitable model for chiller FDD, no FDD system was proposed or developed. Two models were based on first principles (from Gordon and Ng (2005) and a modified ASHRAE Primary Toolkit from Bourdouxhe et al. (2007)) and the third was an empirical model. While each model has some distinct advantages and disadvantages, they concluded that the accuracies of all three models were similar. Hydeman et al. (2012) reported that the three models compared by Sreedharan and Haves (2001) were not accurate in predicting the power consumption of chillers with variable condenser water flow and centrifugal chillers operating with variable-speed drives at low loads. They reformulated the Gordon and Ng model and found that it performed better than the three models described above.

Castro (2002) used a physical model developed by Rossi (2005) along with a k-nearest neighbour classifier to detect faults and a rule base to diagnose five different faults (condenser and evaporator fouling, liquid line restriction, and refrigerant under- and overcharge) in a reciprocating chiller. The FDD implementation detected and diagnosed condenser fouling, refrigerant undercharge at faults level of 20% or greater, and evaporator fouling and liquid line restriction at fault levels of 30% or greater.

2.2.5 FDI on Air Handling Unit (AHU)

Recent and representative FDI studies for Air Handling Unit (AHU) system as the main part of HVAC system are summarized in this section. Glass et al. (2005) presented a qualitative model-based approach for detecting faults in an AHU. The study analysed certain steady states of the subsystem of the plant in terms of qualitative criteria. Steady state conditions of the system were observed for a sufficient length of time. Then, the controller outputs of preheating coil, damper, and cooling coil were converted to qualitative values, such as “CLO” stands for “close”. The temperature measurements of outside air, supply air and return air were input to a model-based predictor which outputs the expected qualitative controller states. If the qualitative controller outputs are outside the allowed sectors during a certain temperature condition, a fault is detected. Faults are detected based on the discrepancies between the measured qualitative controller outputs and the corresponding model-based predictions. No discussions or examples were provided for fault diagnosis. Fault detection sensitivity and ability to deal with false alarms were not discussed.

Lee et al. (1996a), (1996b) presented two methods for detecting faults in a laboratory variable air volume (VAV) AHU. The first method defined residuals that provided a measure of the difference between the normal operation condition of the system and the measured one. A fault was detected when the residuals show significant difference. In the second method, parameters of ARX model were estimated by employing a single-input/single output ARX system identification method. Model parameters were determined using the Kalman filter recursive identification method. When a process operates under normal operating conditions, the parameters in a continuously updated model of the process would be at their normal values. If some physical changes in the system cause deviation from the normal state, some or all of the parameters would deviate from those normal values. Therefore, the faulty condition could be identified. Eight typical faults were discussed in this paper because each faults has a unique signature, no separate diagnosis is necessary. For the parameter identification method, it worked well for complete or abrupt failure, but does not fit for performance degradation fault such as fouled cooling coil fault. In addition, parameters were found to be very sensitive and varying with load. If building load change rapidly, this method may not detect faults.

Henk (2007) described the application of ARX models for real time model-based fault diagnosis in AHU system. In the system identification approach adopted in this paper, two hierarchical levels of modelling were distinguished, the system level and the component level. The system level model was used to detect faults. After a fault had been detected, the fault would be diagnosed on the component level in order to obtain specific information about which part of the system is defective. From the viewpoint of model-based diagnosis, the AHU is a "sequential system". This means that the components of the system are connected in a chainlike way. The AHU was decomposed into a network of several parts, such as the return fan, the mixing box, the cooling coil, the heating coil, and the supply fan.

House et al. (1999) demonstrated the application of several classification techniques for FDI of seven different faults in an AHU. The test data for the comparison were generated using a simulation model. Residuals were used as signatures for various faults and defined as the difference between the actual and expected values of a variable or parameter. An expected value could be a set point or a model prediction. In this study, the training data are labelled, that is, the class of operation of each data point is known as priori. Although all training data are taken from data sets for which the status of operation is known to be either normal or faulty, the residuals at any given time are not always consistent with what was expected for a particular operating status. ANN classifiers, nearest neighbour classifiers, nearest prototype classifiers, a rule-based classifier, and a Bayes classifier were compared for both fault detection and fault diagnosis. The Bayes classifier appeared to be a good choice for fault detection. For fault diagnosis, there were not significant differences in the performance of six classifiers. In addition, all of the classification methods as well as neural networks method need both fault-free and faulty data for the development of models.

Carling (2010) presented a comparison of three fault detection methods for AHUs. The three methods included a qualitative method that compares controller outputs and model-based predictions, a rule-based method that examines measured temperatures and controller outputs, and a model-based method that analyses residuals based on steady-state models. The main conclusions from the comparison of three fault detection methods based on field data of an air-handling unit and artificially introduced faults were as follows:

- The qualitative method was easy to set up, required no additional sensors, and generated few false alarms. However, it detected only a few of the faults introduced.
- The rule-based method used standard sensors, was straightforward, and detected more faults. Some of the rules were system specific and were not applicable to the current plant and therefore generated false alarms.
- The residual method also detected more faults. However, it required additional sensors, was time-consuming to set up, and generated false alarms due to varying airflow and water flow rates.

Norford et al. (2002) presented two methods for detecting and diagnosing faults in AHUs. One of the FDI methods used a first-principles based-model of system components, and the second method was based on semi-empirical polynomial correlations of sub-metered electrical power with flow rates or process control signals generated from historical data. The two fault detection method compared the differences between the observed system behaviour and a reference model of the system operation. Experiments were conducted to artificially introduce faults on three AHUs, totally eleven faults were introduced. The two methods were then evaluated using experimental data. The criteria used in the evaluation of the two FDI methods are sensitivity, robustness, the number of sensors required, and ease of implementation. Both methods detected nearly all of the faults, but diagnosis was more difficult than detection. The electrical power correlation method was generated from data collected from normal system operation, under closed-loop control, and demonstrated greater success in diagnosis, although the limited number of faults addressed in the tests contributed to this success. The first-principles based-model was sensitive to the occurrence of non-ideal system behaviour which was caused by seasonal weather change, considered it a design fault. The grey box electrical power models were less sensitive to the occurrence of non-ideal system behaviour; however, it cannot detect a fault that affects performance but had no effect on the electrical load.

Lee et al. (2004) proposed a scheme for on-line FDI at the sub-system level in an AHU. Four local level control systems were included: (i) cooling coil sub-system which includes supply air temperature; mixed air temperature; mixed air humidity; supply air flow rate; and cooling coil control signal; (ii) supply fan sub-system which includes supply air duct static pressure; supply air flow rate; mixing box damper control signal; and supply fan control signal; (iii) return fan sub-system which includes supply air flow rate; return fan control signal; mixing box damper control signal, and return air flow rate; (iv) mixing box sub-system which includes outdoor air temperature; return air temperature; supply air flow rate; return air flow rate; mixing box damper control signal; and mixed air temperature. This approach requires representative training samples to adequately span the variation in the data and store the training sample, which is also required by statistical method. Clustering techniques group samples so that a group can be represented by a single prototype node. Drifting sensor faults were examined in this paper. However, the four residuals generated from four local component systems are not sufficient to diagnose faults on the component level. As noted previously, the present method only indicates the presence of a fault on the subsystem level. Thus, the supply-air temperature sensor fault and the fouled coil would be classified as the same fault. To further locate faults, expert knowledge combined with some statistical method is helpful; functional test methods are the last choice if measurements are not sufficient to isolate faults.

Chapter 3

3 HVAC Foundation and Simulation

Note:

3.1 HVAC Foundation

3.1.1 Introduction

All occupied buildings require a supply of outdoor air. Depending on outdoor conditions, the air may need to be heated or cooled before it is distributed into the occupied space. As outdoor air is drawn into the building, indoor air is exhausted or allowed to escape (passive relief), thus removing air contaminants. A HVAC system call to the process for accomplish of the mentioned requirements.

A heating system (“H” in HVAC) is designed to add thermal energy to a space or building in order to maintain some selected air temperature that would otherwise not be achieved due to heat flows (heat loss) to the exterior environment. A ventilating system (“V”) is intended to introduce air to or remove air from a space to move air without changing its temperature. Ventilating systems may be used to improve indoor air quality or to improve thermal comfort. A cooling system ("C" is not explicitly included in the HVAC acronym) is designed to remove thermal energy from a space or building to maintain some selected air temperature that would otherwise not be achieved due to heat flows (heat gain) from interior heat sources and the exterior environment. Cooling systems are normally considered as part of the “AC” in HVAC; AC stands for air-conditioning (Trane, 2004).

An air-conditioning system, by ASHRAE definition, is a system that must accomplish four objectives simultaneously. These objectives are to: control air temperature; control air humidity; control air circulation; and control air quality. Although the word “control” is often loosely construed, encompassing anything from pin-point control for central computer facilities to ballpark control for residences, the requirement that an air-conditioning system simultaneously modify four properties of air demands reasonably sophisticated systems. This module will focus on air-conditioning systems, as owner and occupant expectations for many common building types tend to require the use of this broad family of systems.

There are two basic methods of conditioning a room or building. The first type is a radiant system; the second type is a forced air system. Radiant Systems usually involve running hot or chilled water through pipes that loop around the structure and radiate into the conditioned space via a floor surface or radiator pipe. Forced air systems use a fan to push air through a duct system where it is conditioned by a coil on a furnace or air handler before being returned to the space. Most systems in the Fredericksburg Virginia area are forced air systems fuelled by either gas (LP, or CNG) or electricity (Heat Pump / Air Conditioner). There are also various types of air conditioning systems which are normally divided into windows, split, package and central air condition. The application of a particular type of system depends upon a number of factors like how large the area is to be cooled, the total heat generated inside the enclosed area.

Heating systems (such as portable electric heaters or fireplaces), ventilating systems (such as whole-house fans or make-up air units), and sensible-cooling-only systems are also used in buildings. The emphasis, however, will be on multi-function air-conditioning systems. Modern public and commercial buildings generally use mechanical ventilation systems to introduce outdoor air during the occupied mode. Thermal comfort is commonly maintained by mechanically distributing conditioned (heated or cooled) air throughout the building. In some designs, air systems are supplemented by piping systems that carry steam or water to the building perimeter zones.

3.1.2 Type of HVAC System

3.1.2.1 Zone; Single and Multiple

A single air handling unit can only serve more than one building area if the areas served have similar heating, cooling, and ventilation requirements, or if the control system compensates for differences in heating, cooling, and ventilation needs among the spaces served. Areas regulated by a common control (e.g., a single thermostat) are referred to as zones. Thermal comfort problems can result if the design does not adequately account for differences in heating and cooling loads between rooms that are in the same zone. This can easily occur if: The cooling load in some area(s) within a zone changes due to an increased occupant population, increased lighting, or the introduction of new heat-producing equipment (e.g., computers, copiers). Areas within a zone have different solar exposures. This can produce radiant heat gains and losses that, in turn, create unevenly distributed heating or cooling needs (e.g., as the sun angle changes daily and seasonally).

Multiple zone systems can provide each zone with air at a different temperature by heating or cooling the airstream in each zone. Alternative design strategies involve delivering air at a constant temperature while varying the volume of airflow, or modulating room temperature with a supplementary system (e.g., perimeter hot water piping).

3.1.2.2 Volume; Constant and Variable

Constant volume systems, as their name suggests, generally deliver a constant airflow to each space. Changes in space temperatures are made by heating or cooling the air or switching the air handling unit on and off, not by modulating the volume of air supplied. These systems often operate with a fixed minimum percentage of outdoor air or with an "air economizer" feature (described in the Outdoor Air Control discussion that follows).

Variable air volume systems maintain thermal comfort by varying the amount of heated or cooled air delivered to each space, rather than by changing the air temperature. (However, many VAV systems also have provisions for resetting the temperature of the delivery air on a seasonal basis, depending on the severity of the weather). Overcooling or overheating can occur within a given zone if the system is not adjusted to respond to the load. Under ventilation frequently occurs if the system is not arranged to introduce at least a minimum quantity (as opposed to percentage) of outdoor air as the VAV system throttles back from full airflow, or if the system supply air temperature is set too low for the loads present in the zone.

3.1.3 Basic Component of an HVAC System

A typical HVAC system, as shown in figure 3.1, normally divided into five main sections including:

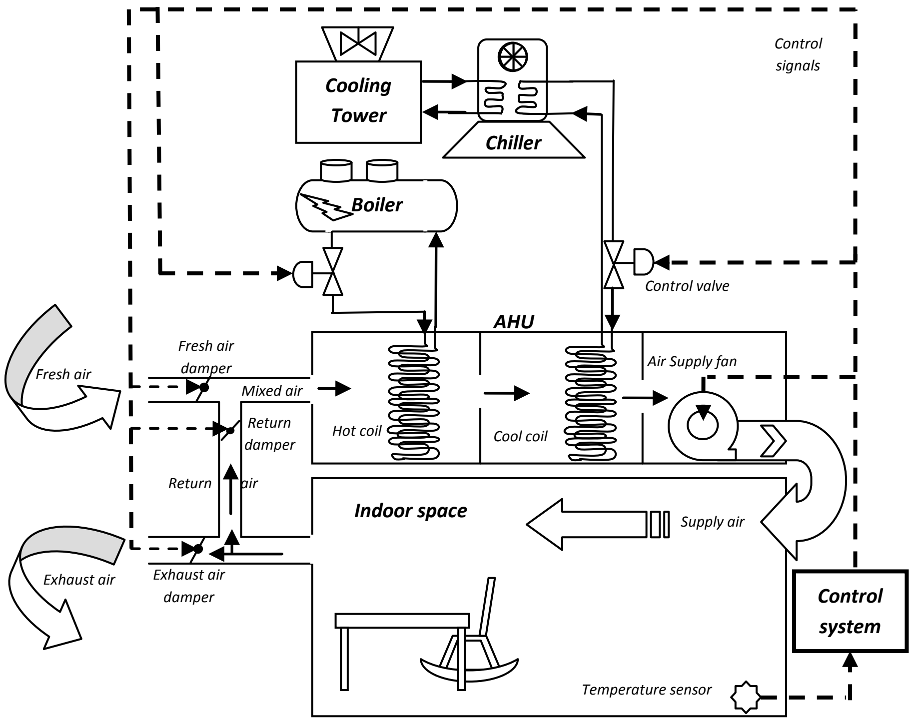


Figure 3.1 General schematic of HVAC with main sections

Indoor-outdoor space and accessories (IOS), Air handling unit (AHU), Cold water unit (CWU), Hot water unit (HWU) and Central control system (CCS). Any main section includes some basic component. The basic components of an HVAC system that delivers conditioned air to maintain thermal comfort and indoor air quality can be introduce in any of above section.

IOS and accessories can be divided to outdoor air intake, ducts, return air system, exhaust or relief fans and air outlet. An AHU consider vital part of a HVAC system which is containing critical components including mixed-air plenum, outdoor air control, Air filter, heating and cooling coils, humidification and/or de-humidification equipment and supply fan. CCS play role of brain for a HVAC system by sensing the condition then make a decision and finally take proper action. It is including Sensors, controllers and actuators. The responsibility of HW system is preparing desired weather by warming the outdoor weather. So the source of hot water may need for HVAC system. This source can be boiler and pumps which are using to deliver hot water. HVAC has more responsibility than warming air in cold season and it should make desirable temperature for hot seasons. Cold water needs for hot days in summer and that is responsibility of CW system which include cooling tower and water chiller

3.1.3.1 Indoor-Outdoor Space and Accessories (IOS)

3.1.3.1.1 Outdoor Air Intake

Outdoor air introduced through the air handler can be filtered and conditioned (heated or cooled) before distribution. Other designs may introduce outdoor air through air-to-air heat exchangers and operable windows. Indoor air quality problems can be produced when contaminants enter a building with the outdoor air. Rooftop or wall-mounted air intakes are sometimes located adjacent to or downwind of building exhaust outlets or other contaminant sources. Problems can also result if debris (e.g., bird droppings) accumulates at the intake, obstructing airflow and potentially introducing microbiological contaminants. If more air is exhausted than is introduced through the outdoor air intake, then outdoor air will enter the building at any leakage sites in the shell.

3.1.3.1.2 Ducts and Dampers

Ducts are used in all HVAC system as channels for distributes conditioned air throughout a building air. These channels also can distribute dust and other pollutants, including biological contaminants. A damper is a valve or plate that stops or regulates the flow of air inside a duct or other air handling equipment. A damper may be used to cut off central air conditioning (heating or cooling) to an unused room, or to regulate it for room-by-room temperature and climate control. Its operation can be manual or automatic. Manual dampers are turned by a handle on the outside of a duct. Automatic dampers are used to regulate airflow constantly and are operated by electric or pneumatic motors, in turn controlled by a thermostat or building automation system.

3.1.3.2 Air Handling Unit (AHU)

3.1.3.2.1 Return, Exhaust and Mixed-Air System

An AHU may include different critical component including dampers, coils and fans as shown in figure 3.2. In many modern buildings the above ceiling space is utilized for the un-ducted passage of return air. This type of system approach often reduces initial HVAC system costs, but requires that the designer, maintenance personnel, and contractors obey strict guidelines related to life and safety codes (e.g., building codes) that must be followed for materials and devices that are located in the plenum. In addition, if a ceiling plenum is used for the collection of return air, openings into the ceiling plenum created by the removal of ceiling tiles will disrupt airflow patterns. It is particularly important to maintain the integrity of the ceiling and adjacent walls in areas that are designed to be exhausted, such as supply closets, bathrooms, and chemical storage areas. After return air enters either a ducted return air grille or a ceiling plenum, it is returned to the air handlers (Arthur, 2003). Some systems utilize return fans in addition to supply fans in order to properly control the distribution of air. When a supply and return fan are utilized, especially in a VAV system, their operation must be coordinated in order to prevent under- or over pressurization of the occupied space or over pressurization of the mixing plenum in the air handler.

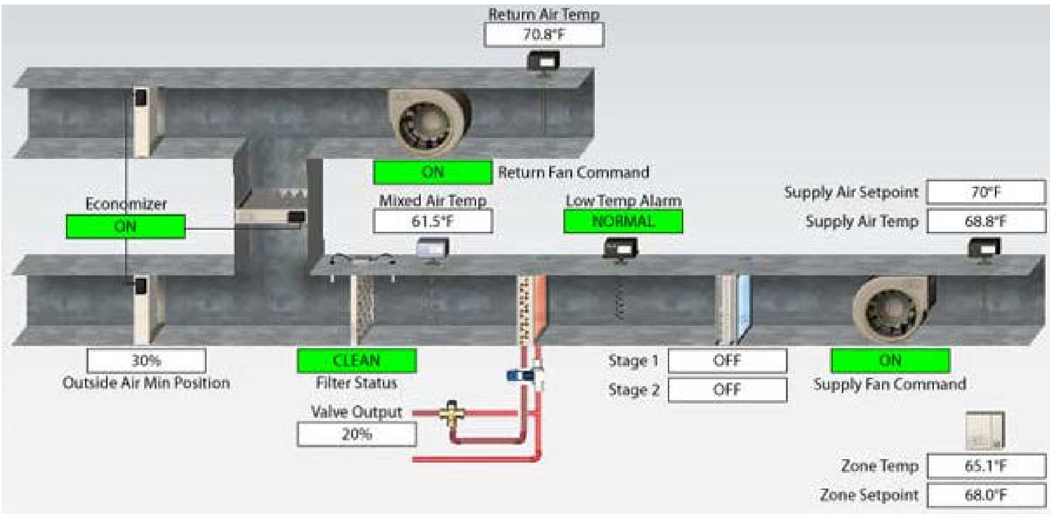


Figure 3.2 Air Handeling Unit (AHU) package

Most buildings are required by law (e.g., building or plumbing codes) to provide for exhaust of areas where contaminant sources are strong, such as toilet facilities, janitorial closets, cooking facilities, and parking garages. Other areas where exhaust is frequently recommended but may not be legally required such as beauty salons, smoking lounges, shops, and any area where contaminants are known to originate.

Outdoor air is mixed with return air (air that has already circulated through the HVAC system) in the mixed-air plenum of an air handling unit. Indoor air quality problems frequently result if the outdoor air damper is not operating properly (e.g., if the system is not designed or adjusted to allow the introduction of sufficient outdoor air for the current use of the building). The amount of outdoor air introduced in the occupied mode should be sufficient to meet needs for ventilation and exhaust make-up. It may be fixed at a constant volume or may vary with the outdoor temperature. When dampers that regulate the flow of outdoor air are arranged to modulate, they are usually designed to bring in a minimum amount of outdoor air (in the occupied mode) under extreme outdoor temperature conditions and to open as outdoor temperatures approach the desired indoor temperature.

3.1.3.2.2 Filters

Filters are primarily used to remove particles from the air. The type and design of filter determine the efficiency at removing particles of a given size and the amount of energy needed to pull or push air through the filter. Filters are rated by different standards and test methods such as dust spot which measure different aspects of performance.

3.1.3.2.3 Coils, Humid Equipment and Fan Supply

Heating and cooling coils are placed in the airstream to regulate the temperature of the air delivered to the space. Malfunctions of the coil controls can result in thermal discomfort. Condensation on under insulated pipes and leakage in piped systems will often create moist conditions conducive to the growth of fungus and bacteria. During the cooling mode (air conditioning), the cooling coil provides dehumidification as water condenses from the airstream. Dehumidification can only take place if the chilled fluid is maintained at a cold enough temperature (generally below 7°C for water). Condensate collects in the drain pan under the cooling coil and exits via a deep seal trap. Standing water will accumulate if the drain pan system has not been designed to drain completely under all operating conditions (sloped toward the drain and properly trapped). Under these conditions, bacteria will proliferate unless the pan is cleaned frequently.

In some buildings (or zones within buildings), there are special needs that warrant the strict control of humidity (e.g., operating rooms, computer rooms). This control is most often accomplished by adding humidification or dehumidification equipment and controls. In office facilities, it is generally preferable to keep relative humidity above 20% or 30% during the heating season and below 60% during the cooling season. The use of a properly designed and operated air conditioning system will generally keep relative humidity below 60% RH during the cooling season, in office facilities with normal densities and loads (Arbor, 2003).

After passing through the coil section where heat is either added or extracted, air commonly use ducts that are constructed to be relatively airtight. Elements of the building construction can also serve as part of the air distribution system (e.g., pressurized supply plenums or return air plenums located in the cavity space above the ceiling tiles and below the deck of the floor above). Proper coordination of fan selection and duct layout during the building design and construction phase and ongoing maintenance of mechanical components, filters, and controls are all necessary for effective air delivery.

3.1.3.3 Cold Water Unit (CWU)

3.1.3.3.1 Cooling Towers

A cold water unit normally include cooling tower, chiller and pumps as illustrate in figure 3.3.

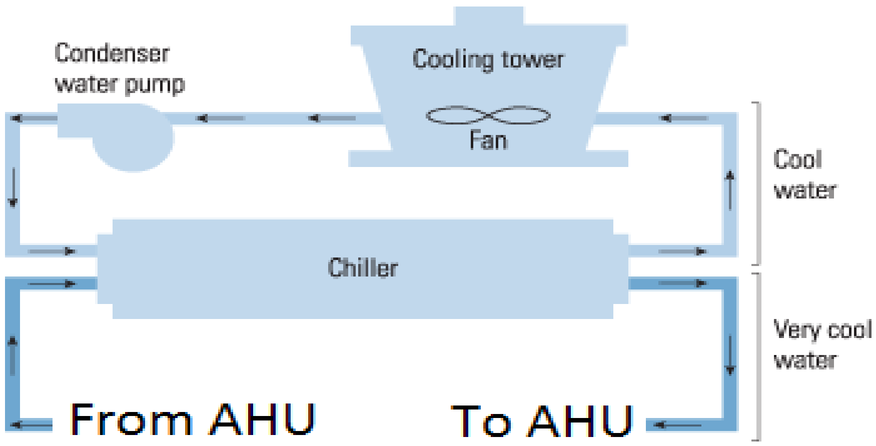


Figure 3.3 Cold Water Unit Diagram for HVAC System

Maintenance of a cooling tower ensures proper operation and keeps the cooling tower from becoming a niche for breeding pathogenic bacteria. Cooling tower water quality must be properly monitored and chemical treatments used as necessary to minimize conditions that could support the growth of significant amounts of pathogens. Proper maintenance may also entail physical cleaning (by individuals using proper protection) to prevent sediment accumulation and installing drift eliminators.

3.1.3.3.2 Water Chillers

Chillers system (figure 3.3) is frequently found in large building air conditioning systems because of the superior performance they offer. A water chiller must be maintained in proper working condition to perform its function of removing the heat from the building. Chilled water supply temperatures should operate in the range of 45°F or colder in order to provide proper moisture removal during humid weather. Piping should be insulated to prevent condensation (Herbert, 2003).

3.1.3.4 Hot Water Unit (HWU)

Like any other part of the HVAC system, a boiler (figure 3.4) must be adequately maintained to operate properly. However, it is particularly important that combustion equipment operate properly to avoid hazardous conditions such as explosions or carbon monoxide leaks, as well as to provide good energy efficiency. Elements of boiler operation that is particularly important to indoor air quality and thermal comfort include:

- Operation of the boiler and distribution loops at a high enough temperature to supply adequate heat in cold weather.
- Maintenance of gaskets and breeching to prevent carbon monoxide from escaping into the building.
- Maintenance of fuel lines to prevent any leaks.
- Provision of adequate outdoor air for combustion.
- Design of the boiler combustion exhaust to prevent re-entrainment, (especially from short boiler stacks, or into multi-story buildings that were added after the boiler plant was installed).

Modern office buildings tend to have much smaller capacity boilers than older buildings because of advances in energy efficiency. In some buildings, the primary heat source is waste heat recovered from the chiller (which operates year-round to cool the core of the building).

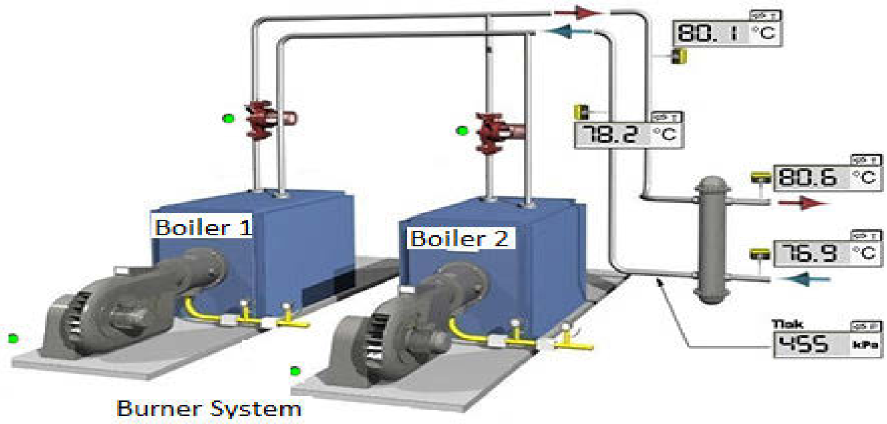


Figure 3.4 HWS system for hot water generation

3.1.3.5 Central Control System (CCS)

HVAC systems can be controlled manually or automatically. Most systems are controlled by some combination of manual and automatic controls.

The control system can be used to switch fans on and off, regulate the temperature of air within the conditioned space, or modulate airflow and pressures by controlling fan speed and damper settings. Most large buildings use automatic controls, and many have very complex and sophisticated systems (a sample of monitoring page is represent in figure 3.5). Regular maintenance and calibration are required to keep controls in good operating order. All programmable timers and switches should have “battery backup” to reset the controls in the event of a power failure. Local controls such as room thermostats must be properly located in order to maintain thermal comfort. Problems can result from:

- thermostats located outside of the occupied space (e.g., in return plenum)
- poorly designed temperature control zones (e.g., single zones that combine areas with very different heating or cooling loads)

- thermostat locations subject to drafts or to radiant heat gain or loss (e.g., exposed to direct sunlight)
- thermostat locations affected by heat from nearby equipment

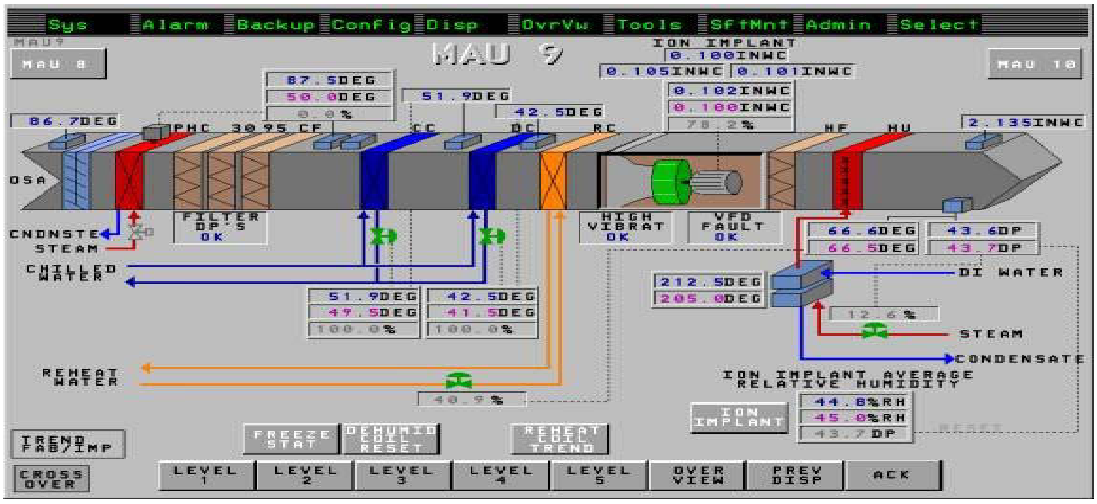


Figure 3.5 Control and HMI monitoring system form HVAC AHU

To test whether or not a thermostat is functioning properly, try setting it to an extreme temperature. This experiment will show whether or not the system is responding to the signal in the thermostat, and also provides information about how the HVAC system may perform under extreme conditions.

3.1.4 The Future of HVAC

How has technology changed in the HVAC field? Well, using PLCs (programmable logic controllers) and BMS in HVAC are the trend nowadays. But a great deal of development of the HVAC system lies on the ever-changing technology and continuous innovation. Companies are adopting automatic fault detection technology after they found proper algorithm for this purpose. A lot of engineers are also focused on further improving this technology through the use of intelligent application, which will work for both model and virtual simulation. The installation of an HVAC system is imperative if we want to achieve maximum comfort and be healthy in our homes, office spaces, or other building facilities. But you also need to consider the building size in installing an HVAC system. Optimum efficiency and comfort level are best achieved if the system is appropriate for the size. After all, any ineffective system usually means more incurred costs in the future. You should also see to it that HVAC is carefully integrated to the overall building design so other aspects needed for proper operations, such as cabling, are not sacrificed.

3.2 HVAC Modeling

3.2.1 History of HVAC Modelling

Energy analysis plays an important role in developing an optimal HVAC and architectural design for new buildings and in determining optimal retrofit and commissioning measures for existing buildings. In most cases, computer simulations are required to develop an optimal design or preventing application due to the complex nature of building energy systems, although manual hand calculations may be more suitable for some simple cases. Before the 1960s, building energy calculations and system sizing were conducted using manual methods such as degree day, equivalent full load cooling, and bin heating/cooling methods if done at all.

Automated calculation methods evolved over the next two decades, with first generation automated methods developed between 1965 and 1975, and with the second generation of automated methods developed between 1975 and 1983. The second generation of automated methods includes both detailed simulation methods and simplified methods. Programs representative of detailed simulation methods are BLAST (Hittle, 2007) and DOE 2.0 (Huang, 2000). These programs are capable of considering a building's dynamic behaviour using hourly simulations. However, detailed input information is required to produce correct output (Kusuda, 2001).

Evaluation of a comprehensive model for a HVAC system always was a problem as many people developed it with some assumption for a special goal. Liu et al. (2008) studied simplified AHU model calibration using whole building cooling and heating energy consumption data. The study indicated that two-zone models work well provided that the interior and exterior zones are properly determined. Although this representation is greatly simplified, the key zone parameters (zone comfort conditions and HVAC energy use) can be simulated well (Katipamula, 2003; Knebel, 2003).

Regarding evaluation of such model with capability for multipurpose application a model developing for general purpose such as pre-control tuning and model based fault detection and isolation has not been developed. More application was based on training system such as neural network for model validation especially for fault detection.

Sensitivity analysis for finding sensitive features relate to especial fault is the most important part of a fault detection system. Some analysis applied for just one fault on one part and nobody reviewed effect of several different fault types on one part. Motivated by these facts, this section proposes a comprehensive mathematical model with an inclusive sensitivity analysis for air supply fan in different fault shape.

Another way for review of system behaviour is development of black box model based on some methods such as ANN. The black box model is suitable method for such systems in real case study, especially when there is not enough knowledge about the process or system. This type of models is normally closer to real case as there are no assumptions to in the stage of model generation. For these cases we just need to input-output data as the input-output layer of model. It is also possible to retrain the model based on new data and revised the model based on new training.

Although the numerical model is closer to real case compare to mathematical models but it has some disadvantage, such as large amount of measuring data under different operation conditions are required to achieve high modelling accuracy. Furthermore, these methods need adequate and reliable sensors for data acquisition and hence, may not be cost-effective. So it seems using of mathematical theory for first step of research on system behaviour study.

3.2.2 Methodology and Requirements

In order to be able to make an assessment of the possible faults and sensitivity analysis of different faults, the system itself including every component mathematically modelled. A mathematical model of the system makes it possible to incorporate the system into dedicated programs such as Simulink. Simulink can then run simulations of the behaviour of the system over time. Every fault of the system can be simulated and then the sensitivity on different variable for every simulation can be compared. Thus the possible fault sensitivity is obtained.

Modelling is the generation of mathematical equations used to describe the response of dynamic processes. Usually this is done by de- scribing the system using well known balance laws (e.g. the law of conservation of mass). Then these equations convert to Laplace transformed or state space model to describe how an input signal transfers through a device (e.g. a valve or a pipe) and becomes the output signal. When a physical system has been modelled it can be simulated using a designated program, e.g. MATLAB. Simulating a model is an easy and fast method to show how the model responds to inputs or any undesired change. For instance the response of a process excited by a step response may be simulated on a computer in a matter of seconds; while in real time it might take several hours or days before the systems settles. Thus, one of the most important reasons for using mathematical models is that it has the potential of saving time with minimum cost effect of any artificial fault on system.

By running tests on a computer it is easier and faster to analyse different type of faults and their sensitivity before the system is implemented into the real system. It is also safer to run tests on the model rather than on the actual system. Another important reason why mathematical modelling is important and has become as popular is that modern fault detectors can be test on the system to check the performance. This information can be used any real system for pre-selection of any possible measurement in order to using in FDI system.

When modelling HVAC systems the following two laws of balance are imperative. The law of conservation of mass states that the mass of a closed system will remain constant over time; the mass can neither be created nor be destroyed. This means that mass stored in a fixed volume is only altered due to mass inputs and mass outputs. This often referred to mass balance.

The law of conservation of energy states that the energy of a closed system will remain constant over time. As for the mass, energy can neither be created nor destroyed. For that reason energy stored in a fixed volume is only altered due to energy inputs and energy outputs. It can however change form, e.g. kinetic energy can become thermal energy due to friction.

3.2.3 Mathematical Model

3.2.3.1 Model Description

In parallel to the modelling of other energy systems, HVAC modelling was developed by Arguello-Serran (1999), Tashtoush (2005) during two decades but these models are generated for a special work. Our new generated model is a general dynamic model without any extra assumption which controlled by a control system same as real HVAC system.

Dynamic models of HVAC system tend to be computationally demanding and for practical utility, particularly in the case of control development and fault detection, must be amenable to numerical solution within an acceptable time frame. A transient model that takes several hours or perhaps days to solve on a computer is of limited practical value and the efficiency of the numerical solution procedure is clearly of paramount practical importance (Pandit, 2003). Models of HVAC system also contain a large number of parameters and the ability to either estimate or measure the model parameters is another key issue for practical applications (Du, 2006). A detailed mathematical model, whose parameters are not open to estimation, or easy measurement, is clearly of limited practical use (Kim, 2005). Our primary objective is to develop nonlinear transient mathematical model of a HVAC system that can be solved efficiently and make a possibility for reviewing the affection of different kind of faults on different variables. A close loop simulation of this model toward its control system for control of different parameters is another new application on simulation in which investigated in this research.

Figure 3.6 shows a general schematic of a HVAC system which is used for mathematical derivation in this research. It consists of four main sections including Air Handling Unit (AHU), Room space (IOS), chillier (CWU) and control system (CCS). Fresh air passes from a heat exchanger cooling coil section to change heat between fresh air and cooling water when HVAC system starts to work. Cooled fresh air is forced by a supply fan to the room. After just a few minutes, the return damper opens to allow room air come back to AHU. The mixing air (this factor is zero when return damper is fully closed) passes from cooling coil section to get its heat and humidity. A trade off among exhaust, fresh and return air is decided by control unit. Also the temperature of the room is regulated by adjusting the flow rate of cooling water by a certain control valve.

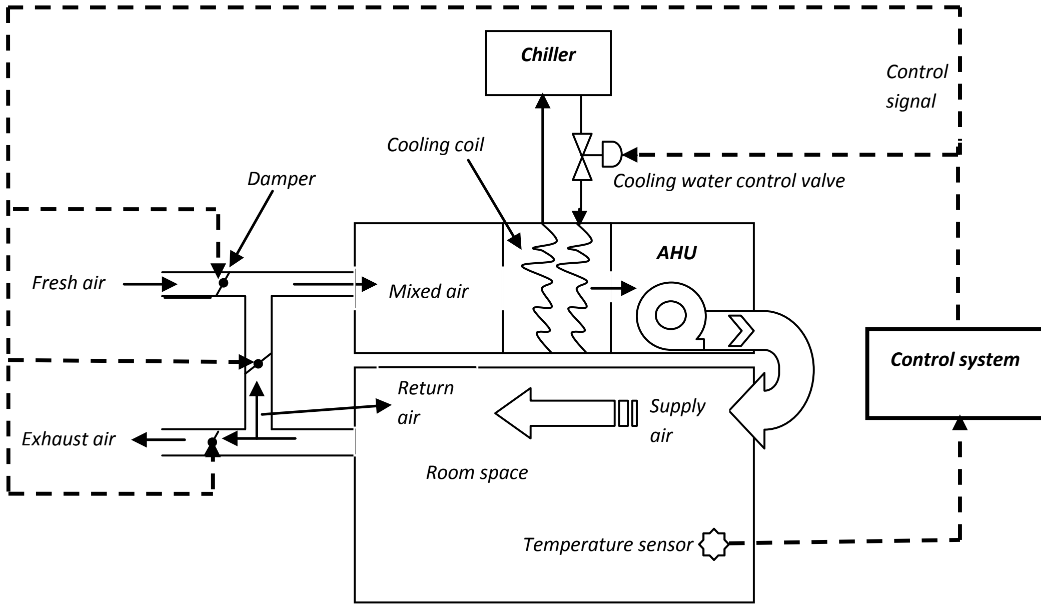


Figure 3.6 General schematic of a HVAC system used in mathematical modeling

Also a square building with a single room is modelled to give simplicity in modelling. When modelling the temperature changes in a room there are numerous variables to consider. Heat conduction through walls, floor and ceiling, heat storage in materials, solar radiation, and heat transferred through the ventilation and heat contribution from radiators are the most significant. When modelling a room or a building the ceiling and the roof can be modelled in the different manner to the walls. Using this approach makes possibility for extending building in any future modelling to make up the whole building. Thus the model of a room in this report will distinguish between walls, the ceiling and the roof.

Modelling the floor is also similar to the walls. The difference is that the floor is not in contact with the outdoor temperature but with the temperature of the ground or the floor below. In this report the floor model has been simplified to a thermal resistance of the floor, in series with the thermal mass of the floor. This is possible since there is only one floor in the building. Otherwise the floor needs to be modelled in the same way as the walls and the ceiling.

When modelling a window it is assumed that the window itself is mass less, that is, it does not accumulate any heat. Because of this simplification there is not any thermal mass component for windows. Hence the window model consists of two heat resistors; one that represents the of heat conduction through the window, and one that represents the heat convection between the indoor air close to the window and the rest of the air in the room.

3.2.3.2 Mathematical Equations and Assumptions

To evaluate the indoor thermal environment, some lumped parameter models of single-zone HVAC systems have been proposed (Federspiel, 2002; Dounis, 2001; Colin, 1999), but none of them considered all the necessary factors of control strategy toward their models. In this paper, a complete HVAC and thermal space model for reviewing of different kinds of faults and is derived by enhancing Arguello (1999) and Federspiel's (2002) models based on four section of Figure 3.6 In the model, four environmental-dependant variables for fault determining are involved, and the ambient disturbance are also considered.

This model consists of several major components (House, 2000): variable-frequency compressor, heat exchanger, variable speed fan, connecting ductwork, damper and mix air components. Some real assumptions are made as follows:

(i) The wall temperature is equal to the mean radiant temperature, i.e. $T_{mrt} = T_w$;

(ii) The indoor air relative velocity is proportional to the supply air flow rate, i.e. $V_{air} = kf_{mix}$;

(iii) The humidity mass ratio is proportional to the vapour pressure, i.e. $w = K_{wv}p$;

(iv) The heat transfer coefficients are the sum of a natural convective heat transfer coefficient and a forced convective heat transfer coefficients (Braun, 2002; Federspiel, 2002) i.e.

$$h = h_c + V_{air}^{2/3} h_v ;$$

The mathematical model is derived from the energy conservation and mass balance in different system components, and the sensible and latent heat exchange are both considered. In the air flow mixer, the return air and the fresh air are missed perfectly, and hence, the equation can be given as:

$$T_{mix} = \frac{1}{r}T_o + \frac{r-1}{r}T_a \tag{3.1}$$

$$W_{mix} = \frac{1}{r}W_o + \frac{r-1}{r}W_a \quad \Rightarrow \quad P_{mix} = \frac{1}{r}P_o + \frac{r-1}{r}P_a \tag{3.2}$$

In the heat exchanger, for the supply air, the equation is given as:

$$\rho C_p V_{he}\frac{\partial T}{\partial t} = f_{mix}\rho C_p(T_{mix} - T_s) + f_{mix}\rho H_{fg}K_{wv}(p_{mix} - p_s) + Q_{he} + \ell h'_{he}\min\{p(T_{he}) - p_s, 0\} \tag{3.3}$$

$$VK_{wv}\frac{\partial p_s}{\partial t} = \frac{\ell h_{he}}{H_{fg}V_{he}}\min\{p(T_{he}) - p_s, 0\} + K_{wv}\frac{f_{mix}}{V_{he}}(p_{mix} - p_s) \tag{3.4}$$

Where ℓ is the Lewis relation, which is derived as $\ell = \frac{H_{fg}K_{wv}}{C_p}$ in this model; h_{he} is heat transfer coefficient on the surface of heat exchanger, which is derived as $h'_{he} = h_{he}V'^{2/3}_{air}A_{he}$; Q_{he} is the thermal power from the heat exchanger, which is derived as:

$$Q_{he} = h_{he}V'^{2/3}_{air}A_{he}(T_{he} - T_s) \tag{3.5}$$

For the heat exchanger, the governing equation is derived as:

$$C_{he}\frac{\partial T_{he}}{\partial t} = -h_{he}V'^{2/3}_{air}A_{he}(T_{he} - T_s) - \ell h'_{he}\min\{p(T_{he}) - p_s, 0\} + Q_{in} \tag{3.6}$$

In the thermal space, the governing equations can be given as:

$$\frac{\partial W_a}{\partial t} = \frac{f_{mix}}{V_a}(W_s - W_a) \quad \Rightarrow \quad \frac{\partial p_a}{\partial t} = \frac{f_{mix}}{V_a}(p_s - p_a) \tag{3.7}$$

$$\rho C_p V_a\frac{\partial T_a}{\partial t} = f_{mix}\rho C_p(T_s - T_a) + f_{mix}\rho H_{fg}K_{wv}(p_s - p_a) + Q_{load} + Q_w \tag{3.8}$$

where Q_w includes the thermal power from the side walls, the roof, the floor and the side windows for simplification, and can be derived as:

$$Q_w = h_w A_w(T_w - T_a) \tag{3.9}$$

where h_w is heat transfer coefficient on the surface of side walls, which is derived as:

$$h_w = h_c + V^{2/3}_{air}h_v \tag{3.10}$$

in the side walls, the equation describing the heat transfer process can be derived as:

$$C_w \frac{\partial T_w}{\partial t} = -h_w A_w (T_w - T_a) - h_o A_w (T_w - T_o) \tag{3.11}$$

Since the thermal space model is developed for different application, three control inputs are provided in this HVAC system as $u = \{Q_{in}, f_{mix}, r\}$: heating / cooling capacity Q_{in} is controlled by the variable-frequency compressor; variable-air-volume is controlled by the variable-speed fan to adjust indoor air flow rate f_{mix}; system-to-fresh-air volumetric flow-rate ratio *r* is controlled by return air damper. Also we assumed cooling capacity Q_{in} is controlled by the flow rate of the chilled waterf_{cc}. The inlet water temperature is set as constant $T_{water\_in} = 7\,°C$, whilst the outlet water temperature $T_{water\_out} = 12\,°C$, which may be disturbed by the cooling load. Therefore, Q_{in} can be derived as follow:

$$Q_{in} = f_{cc} C_{water} (T_{water-in} - T_{water-out}) =$$

$$h_{water} f_{cc}^{2/3} A_{cc_w} \left(\frac{T_{water-in} + T_{water-out}}{2} - T_{cc} \right) \tag{3.12}$$

where $h_{water} f_{cc}^{2/3}$ is regarded as the heat transfer coefficient in the cooling coil under different chilled water flow rates, and $(T_{water-in} + T_{water-out})/2$ is regarded as the average temperature of the water heat exchanger. Then for the chilled water, the equation is derived as:

$$T_{water-out} = T_{water-in} - \frac{Q_{in}}{f_{cc} C_{water}} \tag{3.13}$$

$$Q_{in} = h_{water} f_{cc}^{2/3} A_{cc_w} \left(T_{water-in} - \frac{Q_{in}}{2 f_{cc} C_{water}} - T_{cc} \right) \tag{3.14}$$

$$Q_{in} = 2 h_{water} f_{cc} A_{cc\_w} C_{water} (T_{water-in} - T_{cc}) / (2 f_{cc}^{1/3} C_{water} + h_{water} A_{cc\_w}) \tag{3.15}$$

Therefore, the governing equation in the cooling coil can be derived as:

$$C_{cc} \frac{\partial T_{cc}}{\partial t} = -h_{cc} A_{cc-air} (T_{cc} - T_s) - \ell h_{cc} A_{cc-air} \min\{p(T_{cc}) - p_s\, , 0\} +$$

$$2 h_{water} f_{cc} A_{cc\_w} C_{water} (T_{water-in} - T_{cc}) / (2 f_{cc}^{1/3} C_{water} + h_{water} A_{cc\_w}) \tag{3.16}$$

Based on the above equations, the nonlinear dynamic model for the HVAC and thermal space can be derived as follows:

$$\frac{\partial T_s}{\partial t} = \frac{f_{mix}H_{fg}K_{wv}}{C_p V_{AHU}}\left(\left(\frac{1}{r}p_o + \frac{r-1}{r}p_a\right) - p_s\right) + \frac{f_{mix}}{V_{AHU}}\left(\left(\frac{1}{r}T_o + \frac{r-1}{r}T_a\right) - T_s\right) +$$

$$\frac{h_{cc}A_{cc-air}(T_{cc} - T_s)}{\rho C_p V_{AHU}} + \frac{\ell h_{cc}A_{cc-air}}{\rho C_p V_{AHU}}\min\{p(T_{cc}) - p_s\,, 0\} \quad (3.17)$$

$$\frac{\partial T_a}{\partial t} = \frac{f_{mix}H_{fg}K_{wv}}{C_p V_a}(p_s - p_a) + \frac{f_{mix}}{V_a}(T_s - T_a) + \frac{Q_{load}}{\rho C_p V_a} + \frac{h_w A_w (T_w - T_a)}{\rho C_p V_a} \quad (3.18)$$

$$\frac{\partial T_{cc}}{\partial t} = \frac{2h_{water}f_{cc}A_{cc_w}C_{water}}{C_{cc}\left(2f_{cc}^{1/3}C_{water} + h_{water}A_{cc_w}\right)}(\,T_{water-in} - T_{cc}) + \frac{h_{cc}A_{cc-air}}{C_{cc}}(T_{cc} - T_s) -$$

$$\frac{\ell h_{cc}A_{cc-air}}{C_{cc}}\min\{p(T_{cc}) - p_s\,, 0\} \quad (3.19)$$

$$\frac{\partial T_w}{\partial t} = \frac{h_o A_w (T_w - T_o)}{C_w} + \frac{h_w A_w}{C_w}(T_w - T_a) \quad (3.20)$$

$$\frac{\partial p_s}{\partial t} = \frac{f_{mix}}{V_{AHU}}\left((\frac{1}{r}p_o + \frac{r-1}{r}p_a) - p_s\right) + \frac{\ell h_{cc}A_{cc-air}}{\rho V_{AHU}H_{fg}K_{wv}}\min\{p(T_{cc}) - p_s\,, 0\} \quad (3.21)$$

$$\frac{\partial p_a}{\partial t} = \frac{f_{mix}}{V_a}(p_s - p_a) \quad (3.22)$$

All last six equations {(3.17), (3.18), (3.19), (3.20), (3.21), (3.22)} consist of six main variables plus equations (3.13) and (3.15) for calculation of T_{water_out} used as the main equations in simulation by code generation in MATLAB space.

3.2.4 Parameters and Simulation

Three kinds of disturbances are also taken into consideration in the system design as $d = \{Q_{load}, T_o, p_o\}$: variation of indoor cooling / heating load Q_{load}, ambient temperature T_o and humidity $RH_o(\,p_o)$. When the HVAC system works in cooling mode, water vapour condensation may occur in the heat exchanger, which means that the corresponding term $\min\{p(T_{cc}) - p_s\,, 0\}$ is nonzero. Therefore, the above model is nonlinear. But in the heating mode, no water vapour condenses, so it means $\min\{p(T_{cc}) - p_s\,, 0\}$. Furthermore, if the controlled parameters f_{mix} and r are only adjusted discontinuously at some specific moment, they can be regarded as constant within the running period. It is a good approach to linearization of model for simplicity, especially at steady state condition.

However, a nonlinear model simulated in MATLAB and Simulink environment based on following mathematical equations. Figure 3.7 shows the block diagram of our HVAC model with a feedback controller. All dampers duty, air supply fan speed and cooling water flow rate are controlled by control system as shown in figure 3.7.

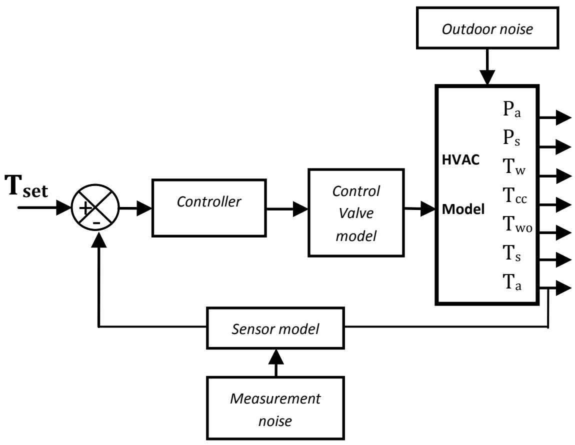


Figure3.7 HVAC schamatic with feedback controller and noise effect

The model is consisting of eight variables which any of them is representative of any section in an HVAC system. Six variables among of eight variables considered as main variables and two remind indicators are assumed as auxiliary variables. So two pressure (air supply pressure P_s and room air pressure P_a) and four temperature (wall temperature T_w , cooling coil temperature T_{cc} , air supply temperature T_s , and room air temperature T_a) are assumed for six main variables. But cooling water flow rate (f_{cc}) and water outlet temperature (T_{water_out}) considered for auxiliary variables. There are different elements for control temperature within a HVAC system but the main factor for temperature control is obviously cooling water flow rate that normally allocated in exist pipe of a chiller section. This flow rate has been controlled by a control signal from control centre system (CCS). CCS usually gets a temperature set-point from operator or end user and compares it with current room temperature. The difference between set-point and room temperature assume as error and controller makes a proportion signal based on this error to send on the cooling water control valve. Then this valve can regulate cold water flow rate through the cooling coil so outlet temperature from AHU can be change based on this forced change. Different strategy can be used on a controller from on-off application, PID controller or even new advanced control strategy including model predictive controller (MPC) or intelligent controller. But in many cases a PID controller has been used due to its high performance and low in cost. PID are define based on proportion, integration and differential elements but a PID in a HVAC controller is normally reduce to PI element including proportion and integration. Differential element often has negative effect in many cases as a delay character (temperature) is main variable for control in this system. Usually a PI controller signal can be tuned in control section of an HVAC system and room temperature acts as feedback signal to compare with set-points.

For our simulation we consider summer state for HVAC system as it can give an opportunity for check fault sensitivity of chiller. Final eight equations including equations 3.13, 3.15, 3.17, 3.18, 3.19, 3.20, 3.21 and 3.22 are used in an M-file of MATLAB program including seven outputs and three inputs according to figure 3.8. Six variables including account as steady state variables equations to be used in Simulink environment. All above equations are coupled and model is nonlinear due to outdoor nonlinearity effects of heat and temperature.

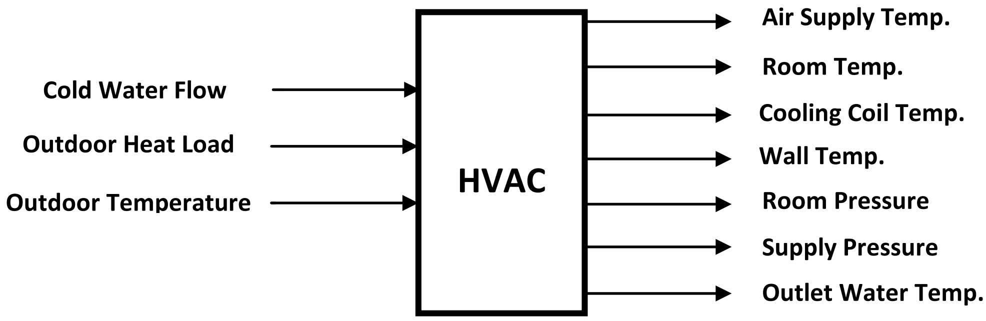


Figure 3.8 HVAC Multi-Input Multi-Output (MIMO) Model

There are different expression for define a model in Simulink environment such as predefine functions or user define functions. Predefine operators like transfer functions can be use in easy way to represent a model however there are some limitations in these functions. Define any type of functions including nonlinear and complicate functions can be used in user define sections such as MATLAB Function (MF) or embedded function. There are some advantages for using MF especially for defines a steady space equations regard usage of integrators feedback to MF. For example in equations that output is derivation of input (steady space) like $\dot{T} = f(T)$ in which input function is T and output is $\dot{T}$ we can use feedback of function output in series with an integrator to prepare function input. This method is like as figure 3.9 which all states are feedback to the system with an integrator. It is also possible to define initial condition of each state in the integrators.

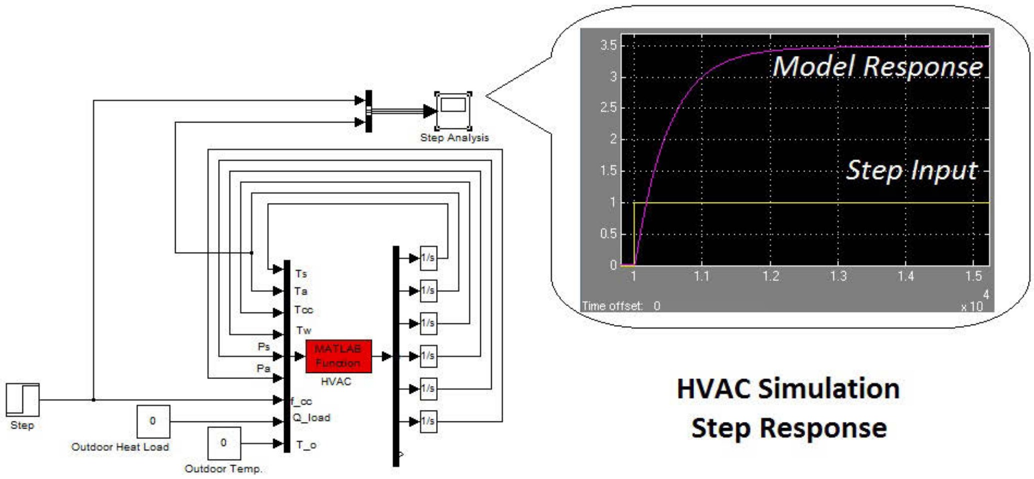


Figure 3.9 HVAC Simulink diagram and step response checking

For our simulation all initial conditions take as environment condition at start time. For example if it is at morning time for simulation then the temperature is morning temperature for starting point. In our first attempt we establish our model without environment (zero in value) effect to analysis for step response (figure 3.9). If we neglect derivation terms, which are just for internal of model, then there are three inputs including cold water flow rate (f_{cc}), outdoor heat ($Q_{\_load}$) and outdoor temperature ($T_{\_o}$) and seven outputs including air supply temperature (Ts), room temperature (T_a), cooling coil temperature (T_{cc}), wall temperature (T_W), air supply partial pressure (P_s), room partial pressure (P_a) and water outlet temperature ($T_{\_water\_out}$) as illustrated in figure 3.9. This also outlined in figure 3.8 as a MIMO system for our simulation in this study. In HVAC system the purpose is usually to control room temperature so for this goal we have access to manipulate cold water flow rate by a feedback controller. There are different ways to control a HVAC system but PID controller is regular one in many cases. Step response analysis often used for finding PID coefficient in many systems and process. In this analysis a step input imply to the cold water flow rate as system input and room temperature response will be check as the process output variable. This response shows in figure 3.9 and will be analysed in details in figure 3.10.

3.2.4.1 Parameter Setting

A control valve applied for the control of water flow rate (f_{cc}) on the chiller. This control valve can be controlled by the Proportion-Integration (PI) controller as one of regular method in real industrial process. In this study, we therefore select PI controller, and tune the controller by using Ziegler–Nichols method (Reaction Curve Method). The procedure was applied by an input of step function with magnitude of one in specific time to open loop model then the response of system analysed according to Ziegler–Nichols rule for drive PID coefficients. Step input applied on the simulation process based on step time of 2.7 hours (10000 Sec) after process start. This step has been used on cold water flow as the main control input to this simulation. As we can see in figure 3.9, the room temperature (process variable) changes from 0 to 3.5 degree of centigrade based on step force of flow rate (ΔMV = 3.5).

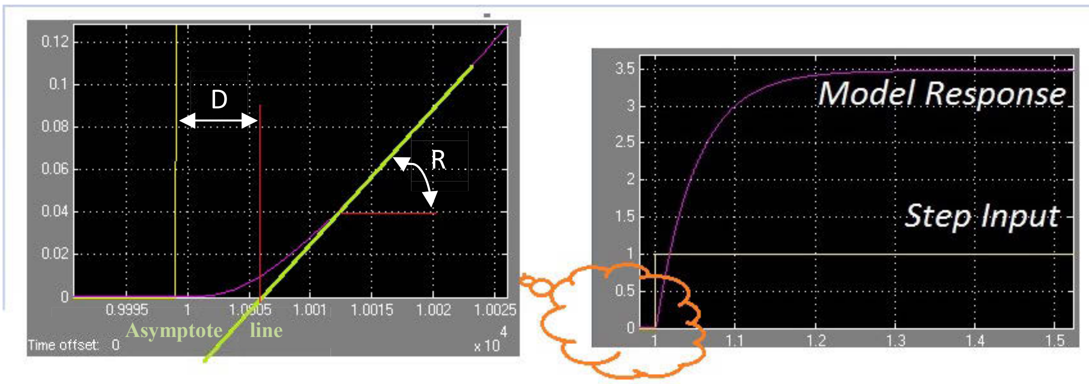


Figure 3.10 HVAC Simulink diagram and step response checking

According to this Ziegler–Nichols method an asymptote line should pass the inflection point in process step response which is shown in figure 3.10. Then some parameters like rate of change of controlled variable (R) and the time between the intercept of tangent line and the initiated change in output (D) are calculate for define of PID coefficients.

Table 3.1 is for P, PI or PID controller coefficients based on Reaction Curve Method. So based on figure 3.10 value of D is $0.0007{\times}10^4$ which is 7 seconds and R calculate as $(\tan^{-1}(0.05/7.5))$ which is 0.4. So according to table 3.1:

$\mathbf{K_c}$ = 0.9ΔMV / (D × R) = 0.9×3.5/(7×0.4) = 1.125 and $\mathbf{T_i}$ = 3.3D = 3.3×0.0007 = 0.0023.

Table 3.1 Calculation of PID Parameters based on Ziegler Nichols

	$\mathbf{K_C}$	$\mathbf{T_I}$	$\mathbf{T_D}$
P	ΔMV / (D × R)	—	—
PI	0.9ΔMV / (D × R)	3.3D	—
PID	1.2ΔMV / (D × R)	2.0D	0.5D

The best coefficient with this method was 1.125 for proportional and 0.0023 for integration coefficient.

In order to simulate the environmental disturbance in real application, three disturbances are considered in the model: outdoor pressure, output temperature and outdoor heating (or cooling) load. Outdoor heat/cooling loading will disturb the system but it cannot be measured directly. Though it can be estimated based on the supply/return air temperature/humidity via a load observer, for convenience, it is assumed the two last disturbances are sinusoidal functions. We also consider some artificial sensor noise in order to closing our simulation to real situation. So the sensor signals are subject to the measuring noises, which are regarded as white, zero-mean and uncorrelated Gaussian processes in the simulation. This white noise is adjusted for 0.5 value of power for sensor measurements.

Table 3.2 shows some major parameters that used in this simulation. Some of these parameters are their values in steady state condition and their real values are change at transient mode. Without loss of generality, simulation runs in cooling mode as most HVAC faults are in cooling mode during summer period. It is easy to consider heating mode just by changing some parameters in table 3.2. Range of disturbances are 26-32 °C and 0.8-1.2 kW for outside temperature and heat loading respectively. For reviewing the effect of air fan supply we shall apply some artificial faults in their direct related variables.

Table 4.1 Major parameter set for simulation

Parameter definition	**Setting value**
Temperature set point	24 °C
Room space dimension	5m×5m×3m
Indoor cooling load range	0.8-1.2 kW
Outdoor temperature range	26-32 °C
Outdoor humidity	55-75%
Max chilled water flow rate	0.5 kg/s
air flow rate(steady state)	980 m3/h
Mixed air ratio(steady state)	4
Noise of temperature	5% mean value
Air handling unit volume	2m×1m×1m
Out side pressure	1 atm.
Inlet water temperature	12 °C

Air flow rate is the strong variable which is directly related to air fan supply. It is also good assumption if mixed air ratio consider as a variable related to return damper position. But in this study we just reviewed first fault in details rather than more general faults. Air flow rate will be varying between their maximum and minimum value just during the fault and get their constant value as described in table I in healthy mode during steady state condition. Also inlet water temperature is set as $T_{water\_in} = 12$ °C whilst the outlet water temperature is varying due to disturbance of the cooling load.

3.2.4.2 Simulation

Figure 3.11 shows our feedback simulation elements including process inputs/outputs, PID controller, valve saturation element, set-point and scopes. After finding PID coefficients and environment situation it was turn to apply conditions on the Simulink model. A PID feedback controller was applied toward a saturation element which acts as a valve for upper and lower limitation. Then environment situation like as heat and temperature was applied based on a summer day for minimum and maximum temperature of 23 and 33 degree of Celsius.

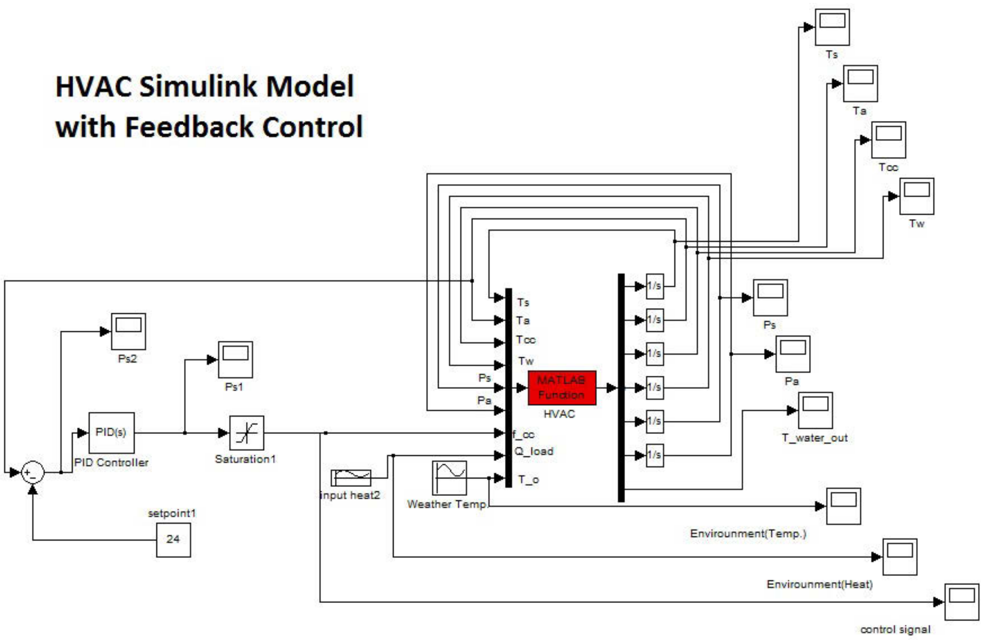


Figure 3.11 HVAC model with feedback controller in simulink environment

The initial values are set for temperature as the ambient temperature at the morning of day. The simulation runs 15 hours from 7:00 am to 10:00 pm for a complete working day. This simulating for real situation gives us a possibility for reviewing the effect of noise in maximum value (at noon).

Initial results of this simulation can be seen in figure 3.12. This figure is divided to four graph including a, b, c and d which any graph represent of a special character. This graph tries to shows how other temperatures are change with outdoor impact. Graph c is situation of outdoor temperature for a summer day that changes from 23℃ to 33℃ during a normal day. As we can see in figure 3.10 the set point is adjusted on 24℃ for this day and this is a reason to a constant value of 24 for blue graph in figure a. blue curve represent room temperature, red one is evidence of air supply temperature, yellow curve is for wall temperature and purple curve try to say about cooling coil temperature. It is clear room temperature try to maintain in constant vale of 24℃ because of set point is in 24℃. However, wall temperature follows outdoor temperature gradually with a rough difference of 3℃ but try to make its temperature close to room temperature.

Interest thing is reverse curve of air supply and cooling coil temperatures which are trying to keep room temperature in its set point. This completely illustrated at noon time when outdoor temperature reach to its maximum value. Figure b is evidence of cooling coil flow rate which was act as control signal. It is clear this variable follows control commands as it found the highest value in noon time when outdoor temperature is in high. Figure d demonstrate the outlet water temperature which is slow change (less than 1℃) but acts in direction of the process to control the room temperature in the constant value.

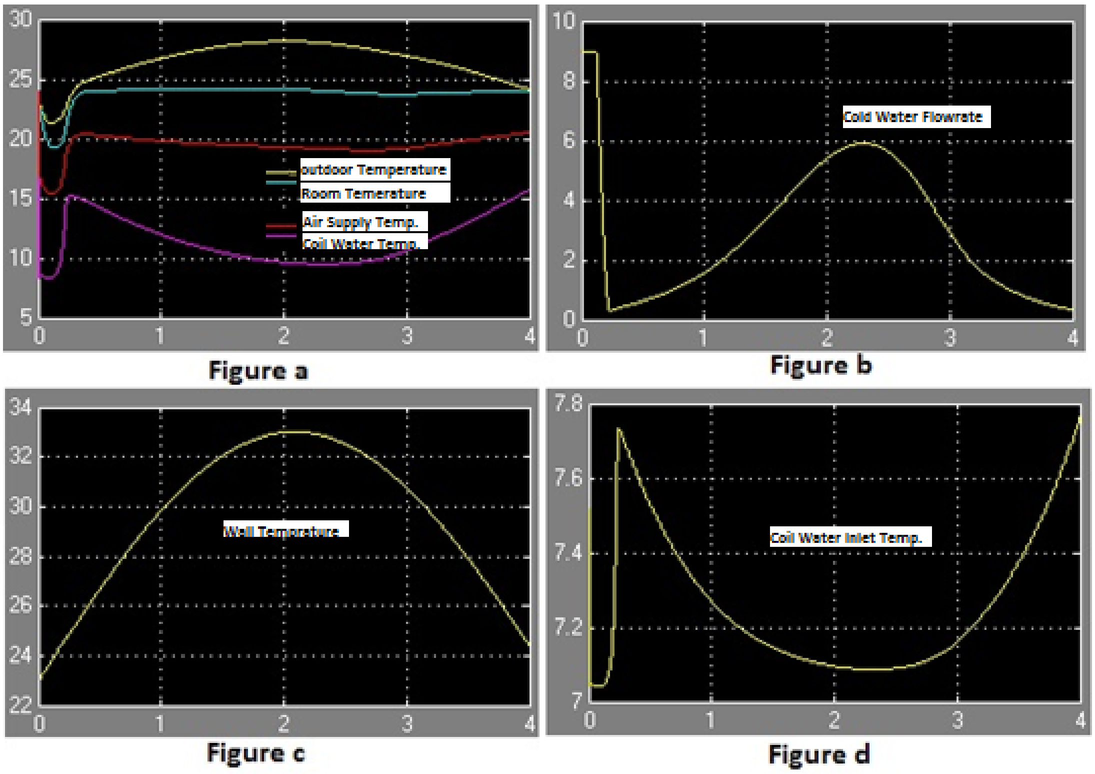


Figure 3.12 HVAC model with feedback controller in simulink environment

In another simulation we try to shows a different outdoor situation and effect of noise measurement by sensors that can be effect on process quality. The overall condition is same as last simulation but outdoor temperature change from 25℃ to 32℃. Also the response of this simulation shows the quality of model work via noise effect. Figure 3.13 (left picture) shows some output temperatures of healthy model (no fault accrue) for set point temperature of 24 °C.

Red curve is outdoor temperature, blue one is wall temperature, black curve is room temperature and green one represent air supply temperature. The first quarter hour of simulation is the transient of system for a good control strategy. It means the system needs 15-30 minutes to reach its steady state condition. Room temperature could stay in the desired value (set point) but other temperatures change with outdoor profile. As we can see the wall temperature is a function of ambient temperature but air supply temperature act inverse with outdoor temperature due to its effort for adjusting the room temperature with the set point.

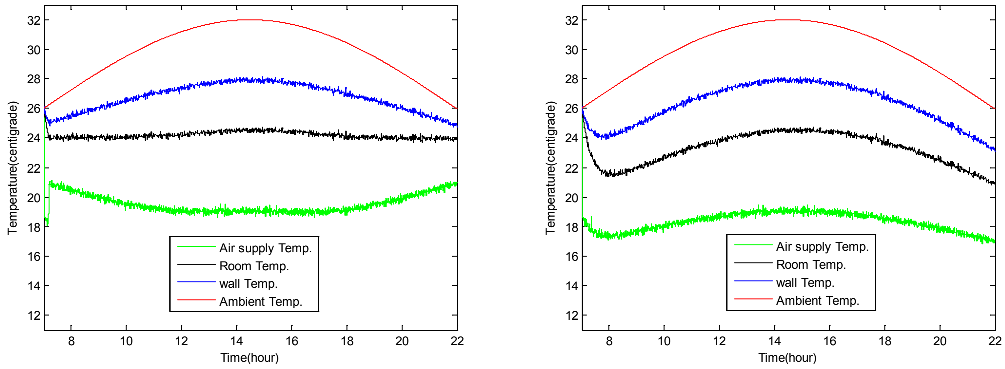


Figure 3.13 Simulation response to rich and poor control tunning

Figure 3.13 (right picture) is an evidence of a poor control strategy for an HVAC system. In this figure PID controller is not tunned well then room could not keep track of set point. As room temperature could not stay at fixed value of set point then it follows the ambient temperature like wall. It should be noted that the steady state condition takes more than one hour for poor control against a good control tuning. By testing the model with Ziegler-Nichols method and get reasonable response we can could confirm our model behave like real HVAC case in terms of PID tuning. So this model can be used for HVAC designer to estimating their start point of PID coefficient in control strategy.

3.2.5 Conclusion

This chapter presents a comprehensive mathematical model for HVAC systems by consideration reality of possible effective factors on a HVAC system. The new model is based on the physical-mathematical relations that are govern on different parts of the HVAC system and hence, can provide reliable results. Both low efficient and strong controllers have been applied in the computer simulation to present the efficiency of model against criteria. In addition to this a wide range of different environment conditions have applied to review model nonlinearity with outdoor forces.

Final model included three inputs and seven outputs which are present a nonlinear model regarding outdoor nonlinearity. Inputs are included outdoor heat load, outdoor temperature and input water flow rate to AHU. On the other hand outputs contain air supply temperature, room temperature, cooling coil temperature, wall temperature, room partial pressure (a representative of room humidity), air supply partial pressure (a representative of HVAC humidity) and water outlet temperature. Also a white noise of measurement applied to the model which is representative of noise measurement and environment noise effect on the measurement system though the HVAC system.

An accurate parameter set have been applied to the model based on outdoor real case study for a summer day. These parameters included indoor and outdoor characters plus control system parameters which are used in feedback close loop model. Finally, a primary simulation of the system presented in this chapter to check validation of mathematical model against outdoor condition and room set point. This simulation confirms the equations work right in the model and it can meet our aspect criteria in a very high simulation performance. The results of this mathematical model have been used in next chapters to establish a fault sensitivity analysing and then impediment to an intelligent algorithm. In addition it will be a useful package to review potential variables on any real HVAC case study before start to applying a FDI system.

Chapter 4

4 HVAC Faults and Sensitivity

4.1 HVAC Faults

4.1.1 Introduction

A HVAC fault diagnosis system is used to detect and if possible to locate faults in a HVAC's operation. A fault may give rise to unfavourable energy consumption or indoor environment, non-optimal operation of a subsystem or faulty operation of a component or a part. Building fault detection and optimization starts with the detection of degradation in the performance of a HVAC as a whole. For example, once it is noticed that the HVAC consumes too much energy the immediate cause needs to be established, i.e. identify the non-optimally operating subsystem or the faulty component. In fault diagnosis, components or subsystems are monitored continuously and faults in their operation can be detected after which the severity of the fault to HVAC overall performance must be evaluated.

A process fault is usually considered to cause an event (i.e. failure) where the required operation of the process suddenly halts. The process is assumed to operate in the required way when the product quality is within predetermined limits. In the context of fault diagnosis, this kind of definition of fault is inadequate. The requirements for the process operation should be stricter and, for example, the operating point deflection from the designed operation point (target) should be considered as a process defect. A defect thus does not necessarily have to affect the product quality in any way because small deflections can be corrected automatically by controllers. This definition has the advantage that it considers the defect with regard to the process and its components themselves, not with regard to the product quality. It is then possible to detect defects before they affect the product quality and thus predict faults. In fault diagnosis, the emphasis should thus be put on monitoring the development of the defect and on minimizing the damages or losses caused by it.

A defect is a circumstance that changes some normal operating point of a process and may cause faulty operation that can be observed from the process measurements. Faulty operation may propagate to other parts of the process or to another sub-process due to changed operating points, or the effect of the fault may be eliminated by using controllers. From the process signals, some test quantities are generated, the variation of which is a symptom of defect. Once the test quantity reaches some predetermined level that reflects the seriousness of the defect, the test quantity is set into an alarm state (symptom) and reasoning is started to find the cause of the 'alarm - symptom - fault' chain. The alarm state is set once the seriousness of the defect requires corrective actions to take place.

An example of a defect is a slowly increasing blockage in a heat exchanger. This causes faulty operation (deviation of temperatures) of the heat exchanger. The defect and the fault are small in the beginning but they increase as the heat exchanger gets dirtier. The defect may result in changes in the inlet and outlet temperatures.

The purpose of the heat exchanger is to transfer heat between the primary and secondary circuits. The reduction in heat transfer between the primary and secondary circuits can be compensated for by increasing the flow rate or the primary circuit inlet temperature, both of which result in additional energy consumption. Once the dirt build up in the heat exchanger reaches a level that the rate of heat transfer cannot be kept at its required value through controlling the process variables, the defect becomes 'observable' to the user of the system and the fault starts to propagate. In real life this possibility of observation might mean that an occupant in the building feels too cold. Thus by monitoring the increase in dirt build up in the exchanger the fault can be detected earlier instead of waiting for complains from the occupants; the increase in energy consumption can also be detected earlier.

In the previous example, the defect occurs when the exchanger starts getting dirty and the need for maintenance should be assessed from the value of the test quantity, i.e. from the amount of harm or loss of energy the defect causes to the user. If the development of the defect is rapid, the test quantity also rapidly reaches the alarm level and the prediction of the defect is difficult. With slowly developing test quantities the prediction is easier.

4.1.2 Type of Failure by Characteristic

While it is impossible to enumerate all possible failures which could occur in an HVAC system, we can broadly categorize them by expected effect and onset characteristics. Note that some of the following failures exist in all HVAC equipment to a degree (for example, sensor bias) however we consider them failures only if they are present to a greater than normal degree (where "normal" is described by somewhat arbitrarily chosen likelihood bounds).

4.1.2.1 Effect

Additive Measurement Failure: sensor bias normally presents as an additive error. The specification sheets of sensors normally provide error limits, which generally present as sensor bias, rather than zero-mean additive noise. Sensor bias is generally a much greater problem than additive noise since it cannot be integrated out of the measurements.

Multiplicative Measurement Failure: a dead or reduced-effect sensor can be modelled as a multiplicative error.

Additive Process Failure: disturbances, leaks, or unknown/unmeasured inputs in the system can be modelled as additive process failures.

Multiplicative Process Failure: deterioration of equipment, reduction of efficacy due to calcification, coil freeze, pipe clogging, etc. They can be represented through a transformation of the state transition function.

Structural Failure: failures which result in a change in performance, but cannot be classified as one of the above categories. Examples might include assigning an incorrect feedback point for temperature control, measurements being "stuck" due to high network traffic, or incorrect programming on a unit.

4.1.2.2 Onset

Similarly, the different types of failure we expect to encounter can be classified as follows:

Sudden Failure: the unit fails in a way that is immediately noticeable. This is considered an easy fault to diagnose, but ultimately depends on the magnitude of failure. Examples include; sudden sensor failure; rupture of an actuator diaphragm; the severing of sensor or actuator signal/power lines due to construction or human error; the failure or incorrect programming of a controller responsible for some of the internal processes of the unit.

Incipient Failure: the unit fails slowly such that the change of dynamics is not noticeable over a short time-frame. A failure may be considered slow if the dynamics change due to the fault occurring slower than the normal (acceptable) drift in system dynamics. This is considered a difficult fault to diagnose. Examples include; shortage of coolant in a heat-exchange system due to a slow leak; development of hysteresis or stiction in valves; difficulty for a fan to generate duct pressure due to a dirty filter.

Pre-existing Failure: the unit does not operate similar to other units in its mode due to a pre-existing and possibly permanent fault. This is generally considered a difficult problem and is not usually considered by FDD techniques that rely on on-line parameter estimation or model fitting. Since other FDD methodologies do not consider unit-groups, there is no way to obtain a baseline. Examples include; a variable air-volume box that has to reheat the air while others on the floor are cooling due to it being located too close to a fan or the incorrect use of reduction sleeves; the mislabelling or misconnection of sensor outputs during installation, where a sensor intended to be connected to one unit is in fact connected to a different unit and a valve connected in reverse. The valve will still work, however the operating characteristics will be sub-optimal, or the valve may fail to fully close when under pressure and disconnected actuator pressure lines. It sometimes happens that a building technician will forget to reconnect a pressurized air tube or wire to an actuator after working on it. This failure could, under the correct conditions, go unnoticed for months. For example, if the valve defaults to open in a chronically underpowered situation, i.e. where the valve needs to be open most of the time, it is possible that there will be no deviation in performance until outside temperatures change significantly. While this is not a permanent fault, it can be classified as a pre-existing one if it occurred before data acquisition started.

4.1.3 Reasoning to Faults

There are different reasons for a particular fault but it is possible to find reason for a fault by analysing with a direction to fault. That could be possible in two directions either from top to down or bottom to up. In implementing a reasoning process, the problem which arises when constructing a system to diagnose faults can be understood in terms of the following two central difficulties:

1) How can the original reason for and location of a fault observed in the functioning of the whole is decided?
2) How can information about a specific fault in a precisely delimited process be combined with the impact of that fault in the functioning of the whole?

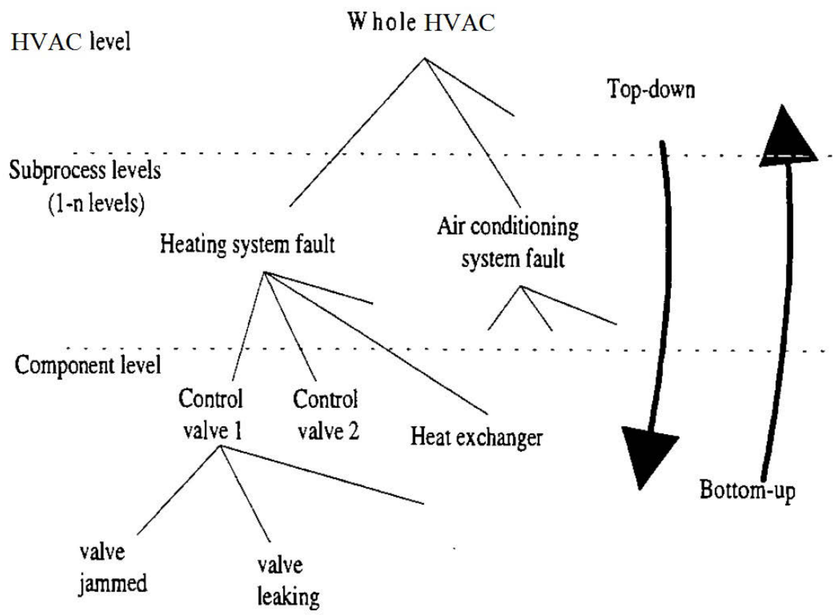


Figure 4.1 Top-Down, Bottom-Up Reasoning

The two problems can be presented in the following manner: When undesired operation is observed on the HVAC level, what is the cause of the problem on the level of subsystems or components? When a fault is observed on the component level, what is the seriousness of that fault in terms of HVAC performance on the level of the HVAC as a whole? In Figure 4.1 the problems are represented by arrows. The tail of each arrow indicates an observation of a fault or undesired operation, while the head indicates the result of the reasoning.

Instead the fault is detected in the HVAC or in the component level it can as well be detected in the subsystem level. In the last case one must be able to deduce the impact of the fault in the subsystem level in a higher level (HVAC level, for example), and on the other hand, be able to find the cause of the fault in a lower level (component level, for example).

4.1.3.1 The Top-Down Approach

In the top-down approach the observed phenomena becomes the objective function. The fact that the fault which is being examined has an undesired effect on the operation of the building is known right from the start. In this respect it is a more natural approach than the bottom-up approach. The top-down approach is well suited for use when there is a desire to locate a fault which has appeared in the system.

What becomes of a problem after the error has been observed, however, is in locating it in the many subsystems and constituent components of the HVAC.

How does localization of the error proceed, and to what degree should it be possible to explain the reason for its occurrence? It should be possible to arrange the reasoning so that it would proceed according to either the order of probabilities or some other corresponding order of importance. For example, if it is a question of how to reduce energy wasting, the first thing that should be considered for inspection is the most probable sub-process which might have been damaged or, alternatively, the partial system which consumes the most energy. If the fault is found there, work should continue according to some preliminary determined system within the subsystem to its components or something similar.

Localizing the fault in the hierarchical tree would thus proceed to the level possible using existing process data which has either been obtained from measurements or requested from the user. If there is no control of reasoning of this type, implementation of the reasoning would be too difficult. This leads in practice to a situation in which no resolution of the problem is obtained.

4.1.3.2 The Bottom-Up Approach

In the bottom-up approach an examination is made of the operation of individual components and small subsystems. A fault in a component or subsystem is not necessarily observable on the building level at the same time as on the lower levels. It can thus be said that bottom-up -approach can be used to predict the consequences which a fault in a specific individual process will have for the entire system.

Since the operation of all the components of the building cannot be monitored in practice, it is of essential importance that when the bottom-up approach is being used such components and faults are selected for investigation for which a failure is either probable or would have the greatest effect on the property or set of properties of the building which had been selected according to the order of importance selected.

4.1.3.3 Meta-Knowledge

A prior knowledge (knowledge available in advance) of the operation of the process system to be diagnosed is of essential importance when designing the operation in both the top-down and the bottom-up approaches. In top-down approach a priori knowledge is needed for designing the reasoning process and in bottom-up approach it is needed for selecting the most important components. Determining the fault without any such prior knowledge would require a considerable calculation or involved reasoning, and is impractical in large systems. A prior knowledge is used in the top-down approach to expedite the reasoning process or to select the reasoning path which is most probable or best corresponds to observations. In the bottom-up approach prior knowledge is used for selection of the components and sub-process to be examined.

Since the knowledge concerning the behaviour of the process consists of the set of rules for an expert system, the meta-knowledge can correspondingly consist of rules which control the use of the rules providing information about the use of the process. Meta-knowledge may be used to change the priority of some rules to correspond to the state of the process under examination, to completely prevent the use of certain rules, or to change the information and facts to be used.

Use of the meta-knowledge describing the control of reasoning depicted above was particularly well suited for use in the top-down approach. In the bottom-up approach the utilization of meta-knowledge in reasoning is not so obvious, and it might not even be proper to speak of using meta-knowledge for the control of reasoning. Instead, it is more appropriate to speak of the use of a priori knowledge in the construction of the system. If prior knowledge is used, the most important parts of the system can be selected in advance, and they can be provided with fault detection for the advance observation of faults. If such a method can be provided for each important part, the top-down approach is not needed at all. Instead, whenever a specific module leads to the detection of a fault, it is known that this will have an undesired effect on the operation of the overall system.

In the case of the bottom-up approach, prior knowledge consists of knowing which components are of central importance from the standpoint of the operation of the overall system, and what size or grade a fault has to have before it is regarded as undesirable.

4.1.4 Typical Faults Review on HVAC Sub-Sections

In this section a range of potential faults on different section of a HVAC system has been reviewed for a real case study. The case study is University of Technology Sydney (UTS) HVAC system which is considers as a large-medium HVAC system in size and capacity. In this study all data was taken from Building Management System (BMS) during years of 2010 and 2011. Also all potential faults are based on maintenance record and fault data recording of the BMS system.

4.1.4.1 Fault Source on Hot Water System (HWS)

This section deals with typical faults occurring hot water system on a typical HVAC system. The heating system is a typical hydraulic heating system used in medium to large size residential and non-residential buildings for space heating as well as for regular hot water heating. The areas under review are based on multi sections of building including tower and east wing. The heat generation plant includes several gas boilers but two gas boilers are undertaking in this study. The design departure temperature is around 90°C.

The distribution network is split into two circuits, one for the tower and one for the wing building. The temperature of each circuit is regulated using a three-way valve.

The control system includes: boiler control and sequencing, scheduling of occupancy periods, adaptive control of the flow temperature set-point of each secondary loop, adaptation of start/stop time prior to occupancy begin/end (optimum start/stop), flow temperature control observing the return flow temperature limitation signal, local temperature control of each zone by thermostatic valves.

From the description of the reference system a failure mode and effect analysis was performed. A restricted list including faults which appears at first sight as the most important was defined. For each list in this fault the process variable deviation due to the fault. The faults are ranked to determine the important ones and the areas where the effort of FDI system should be focussed. Some of regular faults in a heating system are but not limit to: Defects of instruments including level, temperature, pressure and flow transmitters; Leakage in pipes and tanks; Blockage of boiler pipes and heat exchangers; Valve defect including control valves, thermostatic valves, three-way valves; Water Pumps defect; Low zone temperature; Boiler low efficiency; Burner air damper defect; Burner ignition fault

4.1.4.2 Fault Source on Cold Water System (CWS)

This section deals with typical faults occurring in cold water sources. Vapour compression refrigeration equipment form the largest portion of the installed commercial refrigeration capacity. The basic vapour compression machine is composed of a compressor, a condenser, a thermal expansion valve, and an evaporator. The compressor, which is served by a lubrication subsystem, compresses the superheated refrigerant vapour. The compressed vapour is then condensed in the condenser, releasing its heat in a sink that can either be air or water. The condensed liquid is subsequently expanded at the expansion valve and evaporated at the evaporator to a superheated state. The heat for the evaporation is either supplied by air, water or a water-glycol solution, and the cycle is repeated.

An extensive list of vapour compression cycle machine faults, including their causes and symptoms is available. This section, however, will address the three faults that are deemed of highest priority due to their frequency and the adverse effects they have on equipment efficiency and mechanical integrity.

Lack of refrigerant is at the root of a number of serious failures encountered in the operation of vapour compression machines. Refrigerant leaks in refrigerating systems are typically due to small holes in the refrigerant tubing, loose fittings or poor connections, typically caused by corrosion, vibration and poor workmanship. The shaft seal of the compressor, the tube plates in the condenser and the service valves of the compressor top the list of sites for significant leaks.

Refrigerant loss is, in general, gradual over a period of time and as such it is difficult to detect before a critical situation arises. Typically, When the refrigerant charge is low, low suction pressure occurs and possibly a low discharge pressure. The low suction pressure, in turn, leads to low compressor capacity resulting in insufficient cooling. The low density of the vapour causes it to collect a higher level of heat per unit of refrigerant causing the refrigerant vapour to reach a higher and higher temperature. When this happens the compressor motor heats the vapour even more until, at the discharge of the compressor, the high discharge temperature may cause severe damage to the compressor.

Air in the refrigerating circuit is another of the more frequently encountered faults, particularly in low pressure centrifugal chillers. Air enters the refrigerant circuit during normal operation, equipment breakdown and maintenance work. Air, during normal operation, is admitted through loose fittings and poor connections, the same way refrigerant is lost. In the case of air, however, a negative pressure generally is required. Such a condition is present in the low pressure side of units operating with R-11 and R-123. Units operating using high pressure refrigerants do not, in general, experience gradual air intake but are more likely to have air present during ineffective evacuation and recharging of the refrigerant.

Air that is present in the refrigerating circuit usually collects in the condenser where it takes up space that would otherwise be occupied by refrigerant, thereby reducing the heat transfer surface. This leads to high discharge pressure since the air is non-condensable. In addition to this symptom, air in-leaks cause the admission of moisture into the circuit, the presence of which can lead to the blockage of the refrigerant flow control device by the formation of ice. This causes low suction pressure and possibly low discharge pressure, symptoms associated with reduced refrigerant flow. Finally, the presence of moisture can lead to acid formation that could damage the expansion valve and motor winding insulation.

Leak of refrigerant to oil is another possible fault in this section. There are two possible modes by which refrigerant can be found in the crankcase oil: the first, called refrigerant flood-back, occurs during the normal operation of the unit, while the second, called refrigerant migration occurs when the unit is off. In both cases serious damage can occur in the compressor since refrigerant will tend to remove the lubricating oil, thereby exposing moving components to excessive wear and causing premature failure of the compressor.

Liquid refrigerant flood-back occurs due to an over-supply of refrigerant to the evaporator or possibly, low evaporator load. In both cases there is not enough heat to evaporate all the refrigerant present in the evaporator causing some of it to be entrained in the vapour and carried to the compressor. The over-supply of refrigerant can be due to expansion valve stuck open; expansion valve bulb loose on suction line; capillary tube system overcharged; coil flooded after hot gas defrost and low evaporator load in which it may be due to the environmental conditions or due to evaporator fans being inoperative; air filters plugged and evaporator coil plugged.

Refrigerant migration occurs naturally due to the vapour pressure differential that exists between the refrigerant and the compressor oil. This problem is more prevalent during the off cycle when the oil temperature can fall due to a low temperature ambient conditions and/or a failed oil/crankcase heater. Consequently the higher vapour pressure of the refrigerant drives it to the compressor oil with which it is miscible. Any situation that drives the vapour pressure of the refrigerant up (higher ambient temperature for example) or that of the oil down worsens the situation.

Symptoms of liquid refrigerant flood-back manifest themselves in a similar manner as refrigerant overcharge. Depending on the external conditions and the particular cause for its presence, refrigerant flood-back may either result in high suction pressure, since if the ambient temperature rises more refrigerant will be available for evaporation, or low suction pressure, since not all the required refrigerant was evaporated as is the case with low evaporator loads. This could result in high discharge pressure and low discharge temperature respectively.

4.1.4.3 Fault Source on Air Handling Unit (AHU)

The air handling unit has outdoor and return air dampers, cooling and heating coils, an air filter section, a supply air fan, a return air fan, and air plenum sections. The supply air is ducted to three VAV boxes which supply conditioned air to three different zones. The return air fan takes air from the conditioned space and discharges it into the mixing section and/or the outside through a motorized exhaust air damper. Both supply and return fans are fitted with variable frequency controllers for regulating static pressure constant in air ducts.

The pressure independent VAV system attempts to maintain a constant static pressure at the VAV box inlets by sensing and controlling the static pressure in the supply duct. A static pressure controller with a PID algorithm sends a control signal to a variable frequency motor controller to vary the supply fan speed. The supply airflow rate is measured and the desired return airflow rate (supply air flow minus both the air flow through the exhaust air damper and the amount of air flow required for building pressurization) is calculated. The desired return flow rate is compared with the actual return rate and the difference (error signal) is used in a PID algorithm to set the return fan speed. The reference system also employs a dry-bulb type economizer cycle to save energy through the use of outdoor air for "free" cooling. Details on the VAV reference system, its control strategy, and the operation of the economizer cycle are given in chapter 3.

Faults in AHU can be divided to four main sections including air mixing section, filter and coil section, fan section and VAV box section. Faults on air mixing section can be consist of temperature or humidity sensors faults (include complete failure, offset, wrong scale, drifting or excessive noise); damper and actuator faults (contain stuck in position of return and exhaust dampers); control signal (no signal, incorrect signal).

Filter and coil also could be contain of faults including: temperature sensors (supply air, water entering or leaving preheating coil, water entering or leaving cooling coil, freeze protection); filter faults (partially clogged, incorrect/malfunctioning DP sensor/signal, leakage through or around); valve and actuator (preheat coil valve, cooling coil valve); stuck (mechanical failure, actuator/motor failure); water leakage; cavitation; partially blocked; too high/low pressure in hot/chilled water supply line; fouled coil; control signal (no signal, incorrect signal)

Faults on the fan section are normally in range of: fan (Failure, stuck); pressure sensor (failure, offset, wrong scale, drifting, excessive noise); control signal to fans (no signal/incorrect signal)

Some faults arte belong to VAV box which are contain: damper and actuator (stuck, air leakage past closed damper, faulty indicator of damper position); reheat coil (fouled coil, water leaks); piping (partially blocked); reheat valve and actuator (stuck, leakage and plugged); flow measurement; VAV box controller

4.1.5 Comparison with Other Fault Study

Further to this study, an investigation was carried out to find out typical faults in HVAC system based on a questionnaire to experts whose professions are ranging from system designers to maintenance engineers. Following conclusions were introduced:

In the twelve important faults selected by experts, five of them are faults originating at design stage and four are maintenance stage. Percentage of design stage faults is larger than that of maintenance stage faults. These design stage faults cannot be fixed by normal maintenance works.

The water proofing damage and the insulation damage are difficult to be detected, and are so tough that fixing these faults requires lots of time and cost.

The too small tank volume and those faults in the controller and control valve also have a bad influence upon energy conservation through insufficient stored heat in thermal storage tank, while the insulation damage and duration of the partial capacity control of heat pump cause energy loss directly.

The reasons related to room environment and energy loss earned the highest point (nearly 30 %). The reasons related to difficulty of detection and cost of fixing rank to the next. Causing secondary damage and credit loss for maintenance were not evaluated as being very important.

The water proofing damage of the first rank is selected mainly because of three reasons; the difficulty of detection, the need for recovery time and the cost of fixing. It should be noted that it took the first place in the important faults but that it does not include the environmental degradation and energy loss which are top reasons for the twelve important faults. The insulation damage which took the second place together with the too small tank volume is selected because of energy loss and difficulty of detection. The other important faults were selected because of environmental degradation and/or energy loss, and are considered to be easy to detect comparatively. Faults concerning the control of the storage system are the biggest in point numbers when summed up.

4.2 HVAC Sensitivity

Sensitivity is a condition where the faulty output begins to exhibit some detectable characteristics. For example, when gradually increasing the intensity of the fault injection, variables can discern the critical condition where a fault begins to occur or the fault becomes stable. By this method a verity range of sensitive variable can be detected on a system. The procedure begins with generating of an artificial fault which is applied on the system and then checking the result of any measurable variable on the monitoring system. However, applying artificial faults on any system have been some negative effects on the performance of the system or even go out of safety definition. So it is regular procedure to impediment an artificial fault on such systems especially for those which are in large scale. In many cases researcher develop a dynamic model of the system and then review this sensitivity over the model. By an appropriate dynamic model of any system or process now it is possible to impediment a wide range of artificial faults and check the categorize variables which are sensitive to a specific fault. Then any algorithm can be developed based on this analysis for fault detection. This is the main reason which we were extended a mathematical dynamic model of HVAC system in this study. Further to this analysis we test the proposed algorithm of fault detection on a laboratory scale of an HVAC system which is present in the last chapters. Although, this implementation is not defiantly in representative of a large scale of

HVAC system but can be a leading procedure for extension of fault detection on a BMS system for large scales.

Design and Modelling of Artificial Faults

Many energy systems are subject to faults or malfunctions and this is more important for such public energy system such as HVAC. The failure of important components in the HVAC system, such as actuators or sensors, can result in unsatisfactory performance, decreased availability, emergency shutdown or even significant damage to the system, humans and the environment. For these reasons, we are interest to check the model sensitivity to some special faults. HVAC system may suffer from many faults or malfunctions during operation but one commonly encountered fault is defined as supply fan fault that is prevalent fault in a HVAC system.

As we reviewed in many reports, there are different shapes for a fault during fault period. Some faults are happening suddenly in just a few time period against other type are remained for long times. There is also some graduate faults that increased with a small slope during time in opposite of sudden fault. The magnitude of the faults can be another significant factor in fault outline beside fault period time. Based on different reports for fan supply failure, six type faults evaluate for a supply fan speed.

As we can see in figure 4.2, a sudden fault is design with different magnitude. It introduces abrupt-temporal fault with drop-off of proposed magnitude to a certain value and then return to its normal value after a short period. These drop is represent for 20%, 50% and 100% of its nominal value. For a better understanding of this fault, it is mapped on fan speed that is normally called air supply fan fault. For example if fan speed drop due to it nominal 80% of its working point then that means a fault with magnitude of 20% happen during normal working condition. It is clear fan will be stopped completely if it is dropping to 100% of its normal work and that means the third part of the profile in figure 4.2 will be happen. For 50% of fall the second part of profile applied which means fan is working with half speed compare to its previous work. The time period of these faults are designed for two separate time ranges of 20 seconds and 300 seconds. The first and last falls (20% and 100%) are based on 300 seconds but middle fault (50%) designed based on 20 seconds of time period. This design is due to checking for influence of different fault time period on the system response. As we discussed in first of this chapter this type of fault (abrupt-temporal) accrues when electrical component in related with device (fan relay for example) start to be devastated. For a better address to the fault period impact a 20 seconds period fault called fault 1.1 and faults with 300 seconds period of time called fault 1.2 in this section.

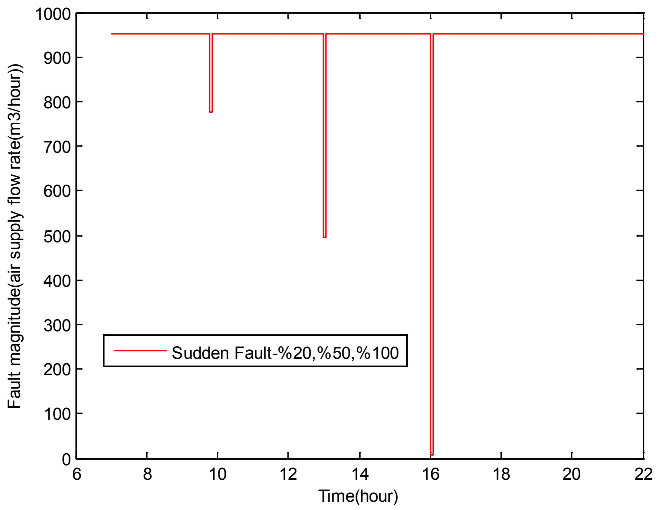


Figure 4.2 Profile for abrupt-temporal faults (Mapped on the air suppy fan air flow)

Figure 4.3 shows a profile for abrupt-permanent fault during a normal daily work of a system. This is a type of fault which accrues in many cases during work of a device and it means the device fall to its faulty zone during a short period but it never backs to its normal condition. This is a situation when device defect or become out of service. Again for a better description of this fault it is mapped on the fan supply during a daily work condition. As we can see in figure 4.3 there is a fall for fan speed from its normal value (normal working condition) to two level of 20% and 50% of nominal value and staying in new situation for ever. The red profile is based on 20% of drop for magnitude which is called fault 2.1 against black profile is representative of 50% drop of magnitude that is called fault 2.2. It is clear which this fault lead to stay fan in its final faulty value and this means fan keep working with faulty situation up to end of simulation. The time period for this fault is for 30 minutes which means fan drop to its faulty zone during this time. In this case the linear rate of decreasing is 6.4 (m3/hr) for red graph and 16 (m3/min) for black graph. Normally, these types of fault are introduced in related with mechanical devices and motors during their lifestyles.

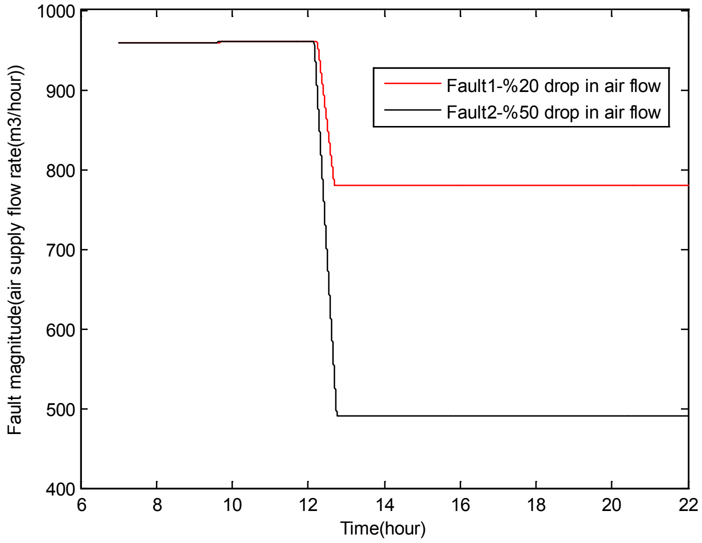


Figure 4.3 Profile for abrupt-permanent faults (Mapped on the air suppy fan air flow)

Figure 4.4 defines an incipient fault that is an important due to its graduate increasing. Despite two last faults this type of fault is reached its final value gradually. These cases of faults are usually hard to detect as it affection cannot detected on most variables. The fault is reached its final value during six hours of time. It is long periods of time compare to sudden faults which are getting its final values just in few seconds or minutes. As we can see in figure 4.4 two profiles are presented based on two different level of fault. The red curve is an incipient fault which is gradually decreased to 20% of its normal value during six hours. On the other hands the black curve is based on 50% of drop during these six hours of time. Same as before a fan supply is used for mapping of this fault to a better presentation. It is clear the air flow reduce from 960 (m3/hr) to 768 (m3/hr) during 360 minutes (six hours). That means the linear rate of this decrease is 0.5 (m3/min) which is very small compare to similar situation on previous fault (6.4 m3/min). Also a curve reduction used for this especial fault to presenting a nonlinear fault for a system that may be happen in many real cases. Same as before the ID for third type faults are fault 3.1 and fault 3.2 for 20% and 50% of magnitude respectively. Also this fault causes fan to force working in its faulty level for rest of the time.

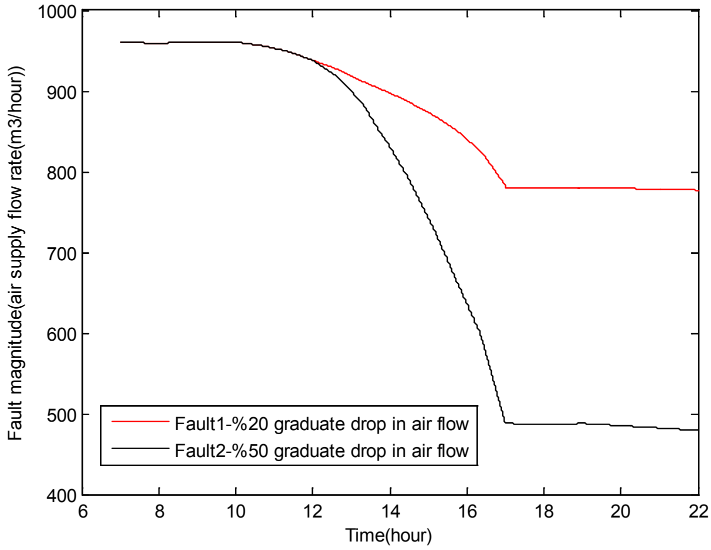


Figure 4.4 Profile for incipient faults (Mapped on the air suppy fan air flow)

4.2.1 Parameters Sensitivity against Fault Type

A HVAC model can be work for cold or hot weather condition but for simulation it is necessary to define the functionality of HVAC regarding cooling or heating mode. Cooling mode consider in this simulation due to reviewing chiller and cold water effect. It is also reported the total faults on hot seasons (HVAC cooling mode) are more than cold seasons (HVAC heating mode). This is probably due to problems through the chiller and cooling tower.

To start fault sensitivity it is require having both healthy and faulty model in parallel working. So a faulty model platform with capability for applying of the different faults prepare beside the healthy model. There are two set of model for this purpose as we can see in figure 4.5. The top model represents our healthy model in this study against bottom model which is based on a faulty system. A different range of faults including abrupt-sudden, abrupt-permanent and incident artificial fault model can be applied on this simulation. Both systems are controlled with a PI controller by a feedback from room temperature. It is possible to apply different faults from AHU, Cold Water System (CWS) or Hot Water System (HWS) by impediment proper change in bottom model. For this purpose it needs to define related variable of model which can influence by the define fault. For example a fault definition for air supply fan can be introduced in terms of change in rate of air flow volume. So air flow is defined as an input variable to faulty model (bottom configuration in figure 4.5) with possibility to change. This variable is defined as f_{mix} in the mathematical equations at pervious chapter. It is also possible to define supply pressure instead of air flow as these two characters are in relation.

Another example is presenting a fault for return damper in air handling unit (AHU) section. As this fault can effectively change the ratio between fresh airs and return air inside of AHU then this ratio can be a good candidate for representing of this fault. This ratio is indicated as r in mathematical equations so on the faulty model this ratio can be seen as manipulated input of system.

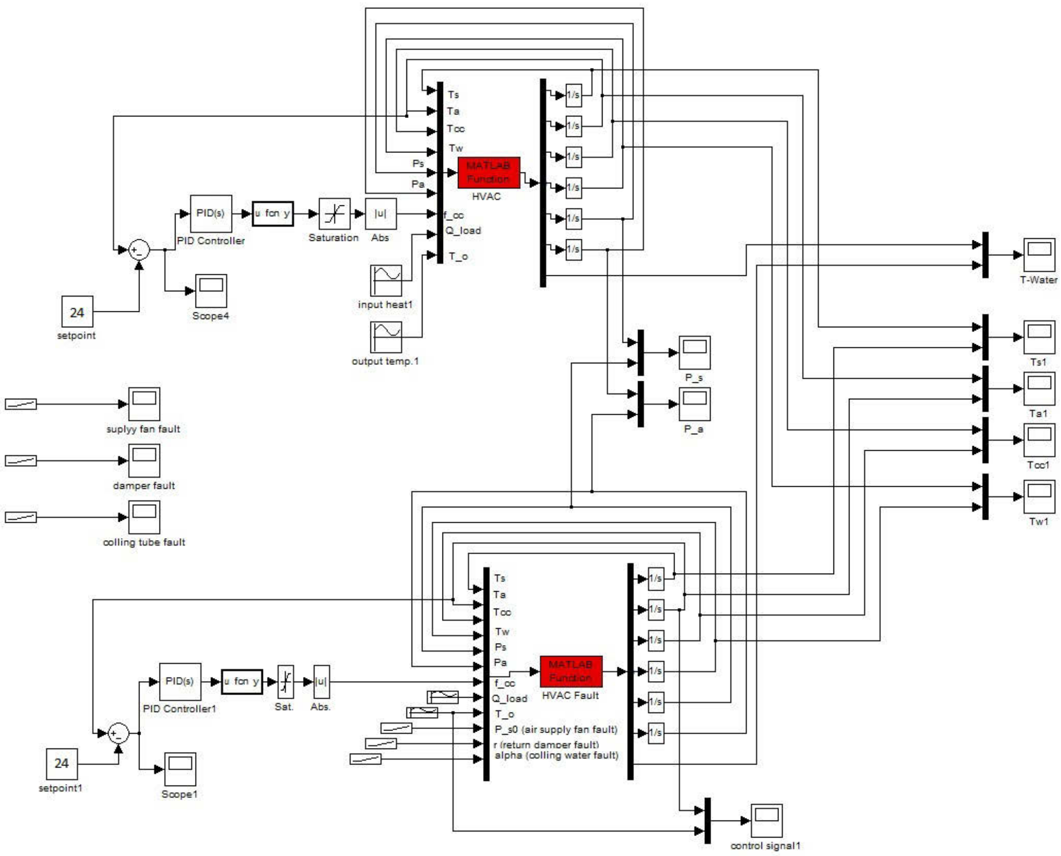


Figure 4.5 HVAC parallel healthy-faulty simulation

Fouling in the pipe lines are a very regular faults not only in HVAC system but also in any pipe network. However, a HVAC include different sections which are connected by pipelines. Also there are many heat exchanger and hot/cooling coil in HVAC process that are including different size of pipe. Depend on which pipe is considered for fouling a different variable can be taken. Here, for example, a cooling coil pipeline is considered for fouling and a proper variable is considered for simulation of this fault. It seems fouling in cooling coil can be strongly effect on cold water flow rate through the coil which is demonstrate as f_{cc} in the mathematical equations. This factor is also used as control variable for controlling of room temperature.

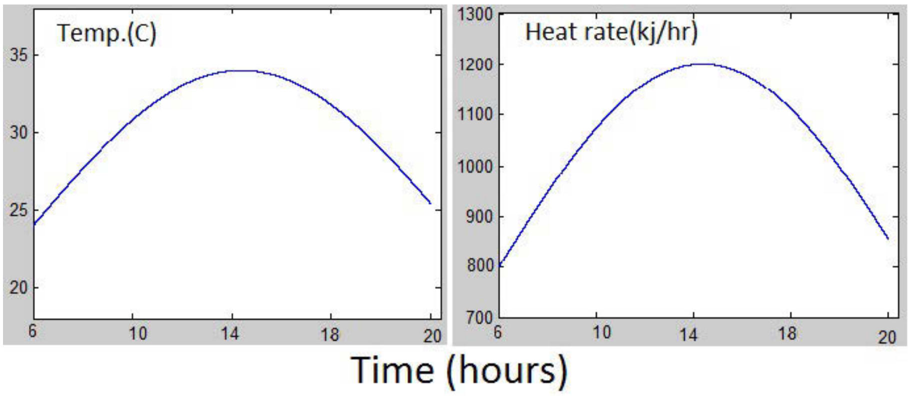


Figure 4.6 Outdoor temperature and heat rate profile between morning to night

The models are both nonlinear due to effect of outdoor condition is impediment on the mathematics. In the hot seasons environmental effects could be outdoor temperature and heat load form sun. These effects are considered in the models as input based on figure 4.6. A regular summer day with maximum temperature at afternoon and minimum temperature at morning and night has been taken as outdoor temperature effect to the system. Outdoor temperature considers changing from 24℃ to 34 C̊ during a summer day. Also the heat load rate changes from 800 (kj/hr) to 1200 (kj/hr) between morning and night. However, faulty model and healthy model are available in details of MATLAB code in appendix A.

4.2.1.1 Abrupt-Temporal Fault Analysis

In this stage we review different faults on the air fan supply and find the best sensitive variable over the test. First applied fault is abrupt-temporal fault which its profile was present in section 4.2.1. The result for each sensitive variable to each fault shows with compare to healthy model. The red graph is recording of healthy model versus back graph which is a representative of faulty model. The fault is type 1 including fault 1.1

and 1.2 which means fan supply speed reduces for two different short periods in three stages during a full simulation. In stage one fan speed reduce to 20% of working value in which the working value is 960 (m3/hr) and it reduced value is 768 (m3/hr). In stage two the reduction is 50% that means the final value is 480 (m3/hr). Finally stage three which is 100% reduction and it is equal to stop fan completely or value of air volume is zero. It should be noted that reduction periods are 20 seconds in all three stages. Based on the simulation for fault 1.1 three variables from 6 monitored variables are sensitive. These three sensitive variables are air supply temperature, cooling coil temperature and water out temperature.

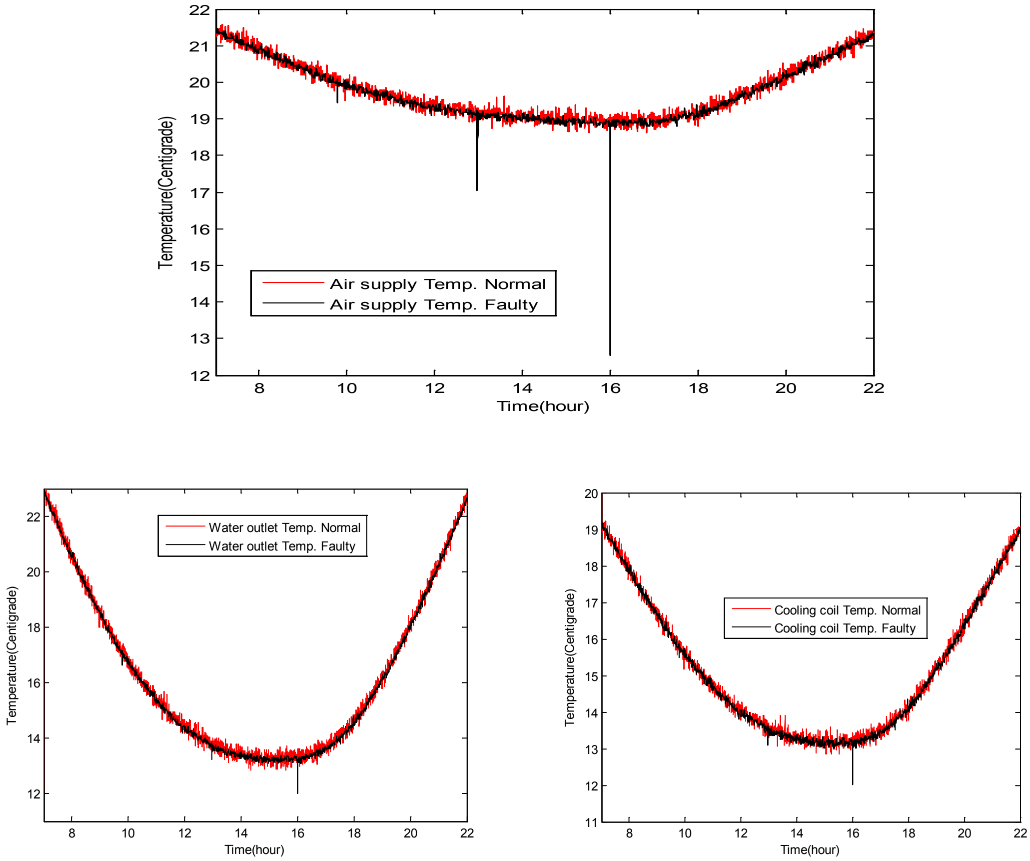


Figure 4.7 Sensitivity response of air supply teperature, cooling coil temperature and water outlet temperature to fault type 1.1

According to figure 4.7 a sudden drop of 1 ℃ temperature happens just in air supply temperature for 20% of fan speed reduction. This value is between 0.2 to 0.5℃ for two others variable which are not good for fault detection. Sensor accuracy and environment noises are usually in this range for temperature measurement within a HVAC system so it seems a range of temperature under 0.5 is not good for fault tracking. The drop off for air supply temperature is 2℃ in case of 50% reduction that is normally a good range for detection. However, this value is still under 0.5℃ for both cooling coil and outlet water temperature. In last stage (fan stop) all three variables become sensitive in different range from 1℃ in cooling coil and outlet water temperature to 7.5℃ in air supply temperature which means all three variable have a potential for fault detection.

Generally, air supply temperature, (T_s) is more sensitive compare to two others and this sensitivity repeated for all magnitude range by T_s. This can be seen in figure 4.7(Top graph) for three stage of reduction air volume over the fan supply. But cooling coil temperature (T_{cc}) and outlet water Temperature (T_{water_out}) are sensitive just for last magnitude of fault which means fan stoped for 20 seconds. Based on this sensitivity for abrupt-temporal fault air supply temperature is a useful variable for detecting of air supply fan stop as well as air supply fan speed reduction. However, cooling coil temperature and outlet water Temperature are also good variables to monitoring of air supply fan stop and large reduction in fan speed magnitude. If fault 1.1 occurs in a bigger period of time we expect other variables can response to this fault. This means if fault 1.2 replace with fault 1.1 then it is possible to see more variables reflex to fault. This is because HVAC system is a lag time system and many variables can find enough time to react with applied fault. Fault 2.1 is similar to fault 1.1 just a difference in period time of the fault. This period is 300 second against 20 second for fault 1.1 but including three stages with same amplitude in reduction. Our expected result is fulfilled in figures 4.8, 4.9 and 4.10 which is present result of model to this fault.

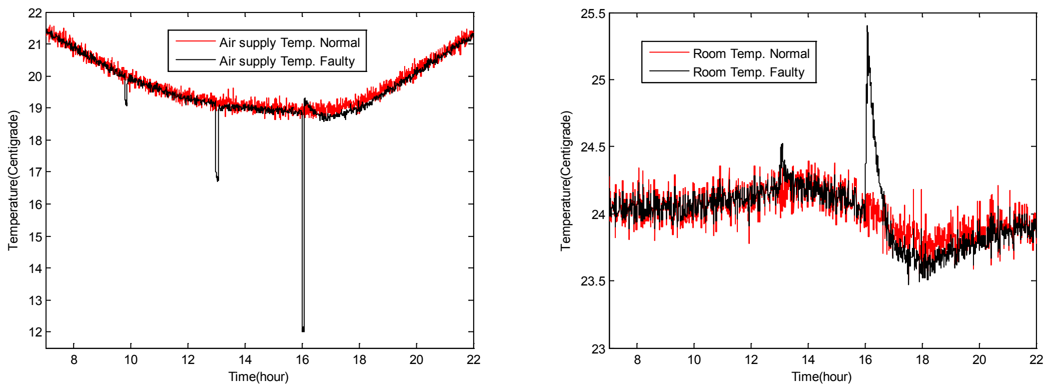


Figure 4.8 Sensitivity response of air supply teperature and room temparature to fault type 1.2

Figure 4.8 is model response of air supply temperature and room temperature to fault 1.2 in three stages. Same as last fault three stages are based on 20%, 50% and 100% of fan supply speed reduction for 300 seconds of period. It is clear air supply temperature is sensitive to all stages but room temperature is just sensitive to last stage (fan stop). Air supply temperature has reaction of 1℃, 2.5℃ and 8℃ drop off for these three stages respectively. In our analyse we found room temperature cannot react to small or medium range of this fault because control system is always trying to compensate any difference between actual room temperature and set pint by increasing in cold water flow rate. However, it has a significant reaction of 1.5℃ to this fault in last stage because fan is stop completely and with any flow rate the room temperature cannot adjusted.

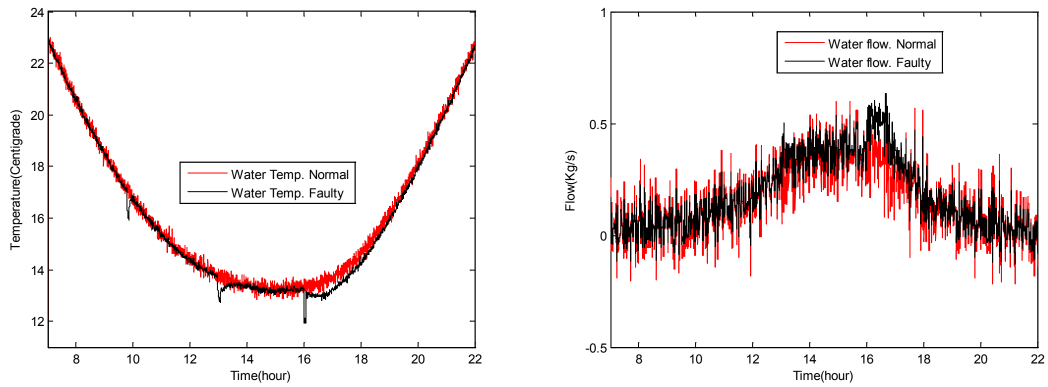


Figure 4.9 Sensitivity response of outlet water teperature and cold water flowrate to three stages of fault type 1.2

This fault also could effect on outlet water temperature in all three stages as we can see in figure 4.9. This variable changes 0.7℃, 0.9℃ and 1.2℃ to these three stages respectively so it seems this is a good parameter for monitoring of this fault beside air supply temperature. Measurement in water flow rate presented with a lot of noise due to applying a white noise with amplitude of 0.2. However, it seems there are two jumps in last two stages of fault regarding this amount of noise in measurement system. If the mean value of measurement used instead of real measurement then we can see jumps of 0.1 (kg/s) and 0.2(kg/s) for two last stage respectively. These amounts of jumps are enough for detecting a fault by a trained system so the water flow rate can be used as a good variable for medium to high range reduction of fan speed amplitude for a period of 300 seconds or over this period time.

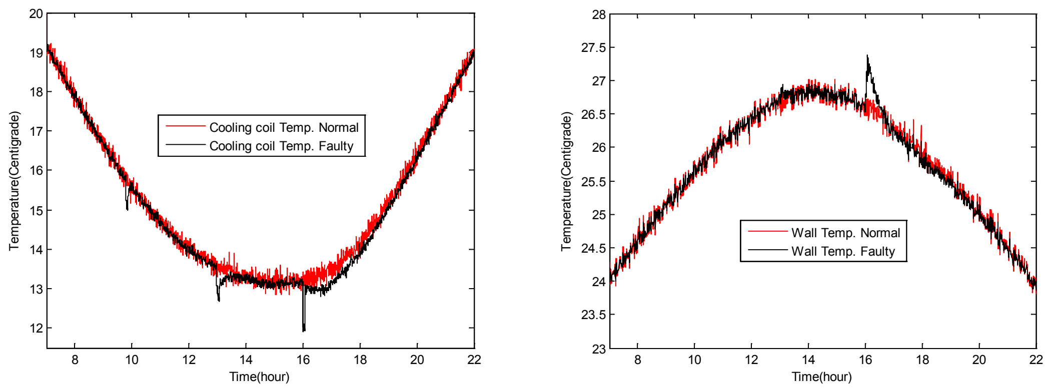


Figure 4.10 Sensitivity response of cooling coil teperature and wall temparature to three stages fault type 1.2

Another variable which could be effective to this type of fault is cooling coil temperature which normally reported in a BMS system. Figure 4.10 shows this sensitivity for this fault in three stage of fan speed reduction. It is clear this variable is sensitive to all three stages by a sudden dropping of 0.7℃, 0.9℃ and 1.3℃ for 20%, 50% and 100% of speed reduction for 300 seconds of period. It is clear cooling coil temperature can be account for analysing of this fault beside of other applicable variables. Wall temperature is not normally measured but if it is measured in some cases then it can be a useful parameter as auxiliary variable in last stage of this fault. This change is present in 0.7℃ of drop in case of fan stop.

Generally, air supply temperature, water outlet temperature and cooling coil temperature are good variables for detecting of this fault in all stages. We can see room temperature (T_a), wall temperature (T_{wall}) and cooling water flow rate (f_{cc}) are added to sensitive variable to this fault in last stages. Although new variables (T_a, T_{wall}, f_{cc}) are not sensitive to low magnitude faults but three main variables including T_s, T_{cc}, T_{water_out} had much reflection to low magnitude parts compare to fault 1.1. It is clear all features are sensitive to bigger magnitude in fault 2.1.

4.2.1.2 Abrupt-Permanent Fault Analysis

Another kind of fault which is known in many different numbers of systems is abrupt-permanent fault. This fault normally stays to its faulty value when it accrues through the system. When this fault happens in air supply fan then it could not reach its normal value after fault time. This fault happens in one stage but for two different modes; one 20% drop of fan speed and another in 50% of drop. Any of these drops has been stayed in its final value. For example in the first mode of fault speed reduce from 960 (m3/hr) to 762 (m3/hr) at a certain time and system will worked to end of simulation with this condition. The response of model to second fault type (with 20% of drop-fault 2.1) is shown in figures 4.11, 4.12 and 4.13 for six sensitive variables. In general view all variable are more sensitive to this fault compare to previous fault (abrupt-temporal). But in details it needs to review the graphs through the figures.

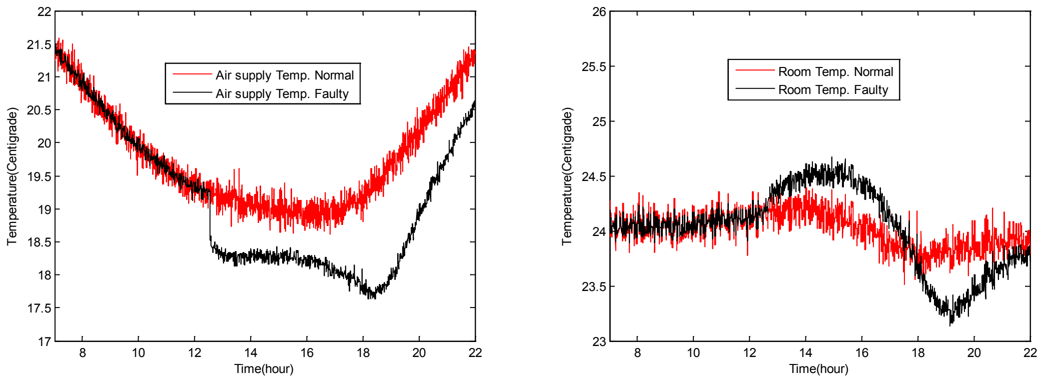


Figure 4.11 Sensitivity response of air supply teperature and room temparature to fault type 2.1

Figure 4.11 present the sensitivity of air supply temperature and room temperature with the permanent fault type for 20% of fan speed reduction. It is clear air supply fan follows a different pattern from healthy model by 1℃ of temperature drop off. On the other hands, room temperature tries to follows the set point with long time fluctuation. This fluctuation is between 0.5 to 0.7℃ and could be a good sign by fault detector algorithm to identify the fault. However, it seems using air supply temperature as major variable and room temperature as a cover temperature is a good idea for detection of this fault.

A similar pattern to air supply temperature happens for outlet water temperature in case of sensitivity to this fault as it has been shown in figure 4.12. As we can see outlet water temperature is trying to follows the healthy model but with a gap of 1℃. On the other graph concentrate on the flow rate of this water and it is a difference of 0.2 (kg/hr) in some points.

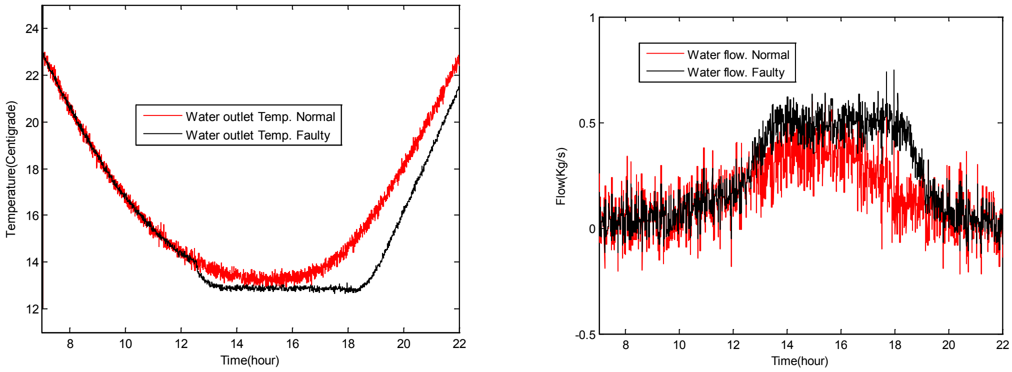


Figure 4.12 Sensitivity response of outlet water teperature and cold water flowrate to fault type 2.1

One interesting point of this graph is for last part of simulation where flow rate follows healthy model in closely. We know the last part of simulation is at night time where the outdoor temperature is mostly close to set point so control signal is not forced to follows the set point as there is not much difference between room temperature and outdoor forced temperature. Generally, it seems both variables have a good potential for detecting of this fault.

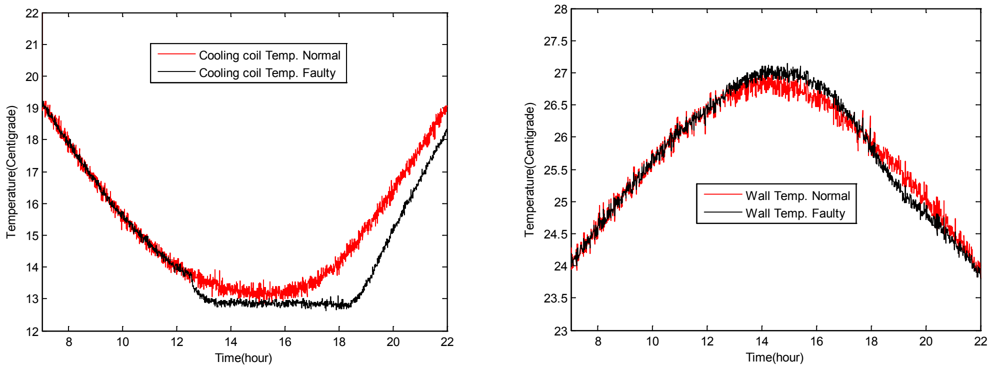


Figure 4.13 Sensitivity response of cooling coil teperature and wall temparature to fault type 2.1

Another potential for fault detection through the HVAC system was always cooling coils temperature as illustrated in figure 4.13. This variable is also follows similar pattern with its two close neighbours; air supply temperature and water outlet temperature. Maximum dropped temperature of 1.5℃ is happen at time of 6:30 pm for all of these three curves and this make a common point for these three variables. Wall temperature was also checked for any possibility of sensitivity but this variable cannot be used as a good factor for this analysis.

In total, as we can see in figures 4.11, 4.12 and 4.13 three variables consist of T_s, T_{cc} and T_{water_out} have a same response shape with different scale to this fault type. Same as before T_s is more sensitive compare to T_{cc} and T_{water_in}. Although T_{wall} is with less sensitivity but T_a shows a better reflection to fault. Another variable which is added to our database for this sensitivity is water cooling flow rate (f_{cc}) that is new option for fault detection of a abrupt-permanent fault.

A changing to magnitude of second fault type from 20% to 50% of maximum value can make a different response to variable sensitivity. This type of fault which normally called fault 2.2 is based on reduction of fan speed from its normal work value (960 m3/hr) to half of this value (480 m3/hr). The results of this analysis are shown in figures 4.14, 4.15 and 4.16 for studied parameters.

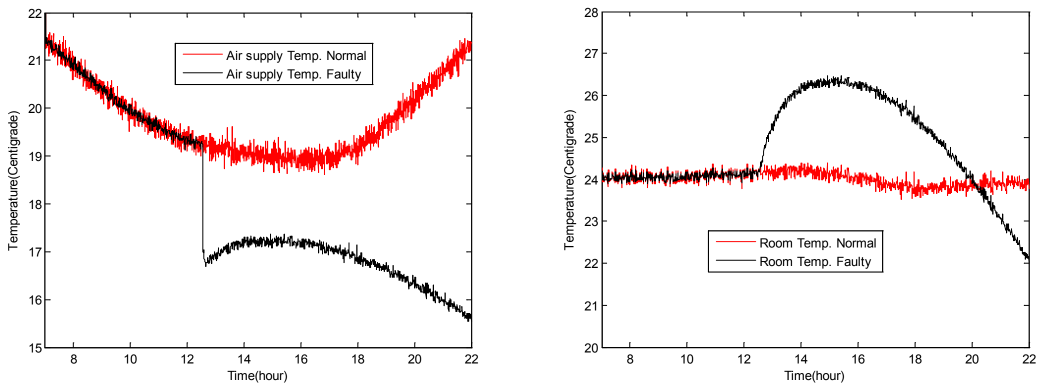


Figure 4.14 Sensitivity response of air supply temerture and room temerature to type 2.2

Figure 4.14 is an evidence for high sensitivity of variable to this fault as we can see a significant change over the air supply temperature. A drop of 2.5℃ accrues with faulty model and then the faulty model follows an inverse pattern of healthy model. As this drop is suddenly happen with air supply temperature then it is a good parameter to detect this fault as soon as it happens. However, room temperature also could not follow the set point and it gets a difference of 2.3℃ with set point in maximum point. This is an evidence for high sensitivity of these two variables to fault 2.2 and can be used for the learning patterns.

Another pattern is follows by two other variables including cooling coil temperature and outlet cold water temperature in figure 4.15. Both graphs have a sudden drop of 1℃ in time of fault and then they could not follow the healthy pattern. They actually stay fix on a constant value of 13℃ because HVAC control system tries to fix control room then use its maximum power to compensate this fault. But as we can see system internally have some limitation and this limit has been illustrate in these both graphs.

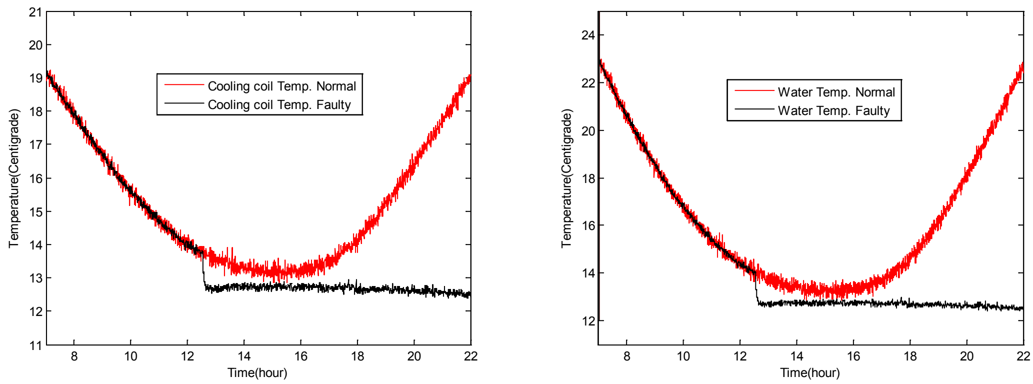


Figure 4.15 Sensitivity response of cooling coil teperature and cold water oulet temparature to fault type 2.2

Another evidence for HVAC system to use its maximum force is shown in figure 4.16. This evidence is on the right graph which is based on cold water flow rate analysis to this fault. It is clear flow rate stay in maximum value of 0.5 (kg/s) to compensate the 50% of reduction in fan speed in order to control signal force.

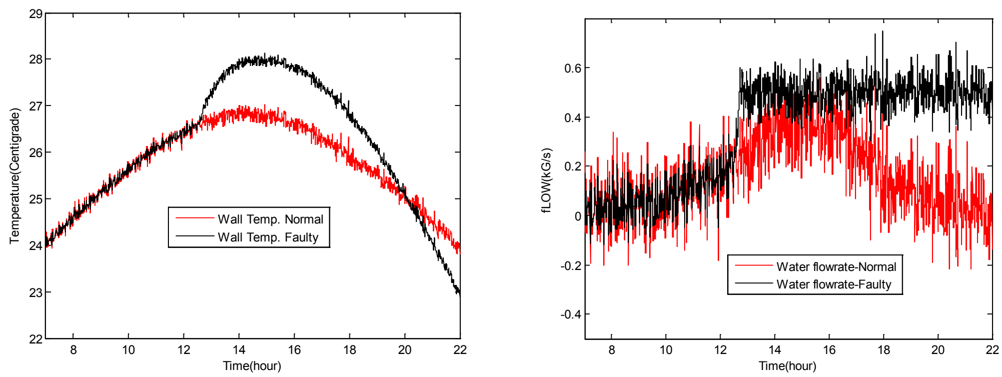


Figure 4.16 Sensitivity response of wall temerature and cold water flow rate to fault type 2.2

But it is not possible to compensate this fault by this amount of flow rate so other variable such as supply temperature change their profile to this force. The constant value of cooling coil temperature and cold water out temperature are in relation with constant value of cold water flow rate base on this fault. In another case wall temperature is reviewed in this figure (left graph) and shows this parameter is also react to this fault. This is the first case which wall temperature has been reacted to a special fault and is evidence to power of the fault.

As we can see in figures 4.14, 4.15 and 4.16 four variables are change with a different algorithm to bigger magnitude of abrupt-permanent fault (fault 2.2). Compare to lower magnitude fault (fault 2.1) this is more dangerous fault as lead to main damage in fan supply and could be detected easier than other faults. Also this fault can be detected by different feature sensitivity from fault starting time. A big gap between faulty and healthy signals is detected. Especial pattern for all features is the most important part in this analysis. Also we can use the wall temperature as sensitivity for power of different faults because of its resistance to react with soft faults.

4.2.1.3 Incipient Fault Analysis

Incipient (soft or slowly developing) failures are defined as small bias or drift errors that increases relatively slowly with time, which are difficult to detect and modelled as ramp (drift)-type changes. But in this part we attempt to simulate and get the sensitivity analysis for developing of fault detection algorithm. This fault called before in first of this chapter as fault type 3. There are two level of this fault; one based on 20% gradually reduction and other is for 50% of reduction in fan speed. Fault 3.1 as the incipient fault defines gradually reduction of fan speed from its steady value to 80% of its nominal value during 7 hours.

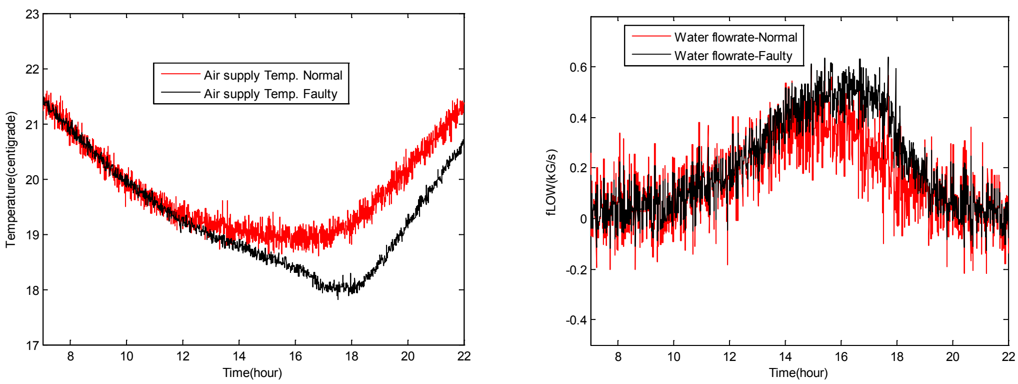


Figure 4.17 Sensitivity response of air supply temerature and cold water flow rate to fault type 3.1

Figure 4.17 shows analysing of air supply temperature and water flow rate sensitivity for an incipient fault with a low graduate drop of 20%. It is clear supply temperature is a good candidate for fault detection algorithm in this type of fault. It should be noted that this fault has not any sudden effect on the variables but the algorithm and pattern of faulty model is seriously important for novel fault detectors. A specific pattern which is follow in this graph is important especially around time of 4:00 pm. Another sensitive variable is cold water flow rate which is presented in right graph. There is a difference of 0.2 (kg/s) of change in maximum value between normal and faulty value for this flow rate. In this case both variables can be used for analysing of fault detection as they are sensitive to this fault.

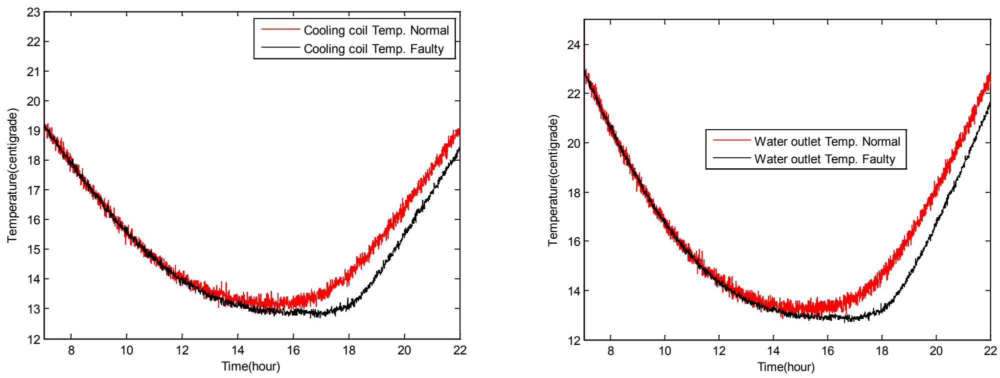
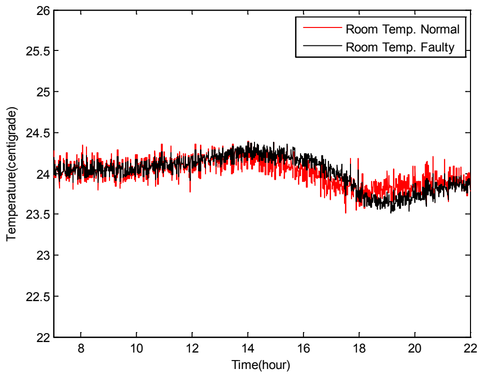


Figure 4.18 Sensitivity response of cold water oulet temerature and cooling coil temperature to fault type 3.1

A same pattern is follows by cooling coil temperature and water outlet temperature for this specific fault as shown in figure 4.18. The maximum difference accrues 1℃ which is in 6:00 pm. Both variable try to follow the healthy pattern but a constant difference happen for rest of simulation. Faults start at noon time of 12:00 but initial signs of fault are start four hours later at 4:00 pm. But in air supply temperature this sign was indicated at 2:00 pm which present they are some lag in time compare with air supply temperature.

Other variables which are sometimes sensitive to some faults are room temperature and wall temperature. In figure 4.19 we try to check this sensitivity for both of them to this fault but in this case none of them are sensitive to this particular fault. We can see room temperature try to stay in constant value of 24℃ which is set point in this case.



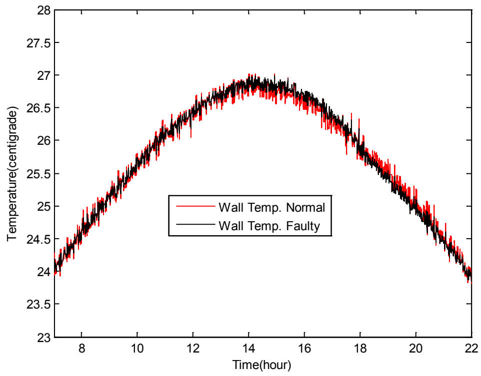
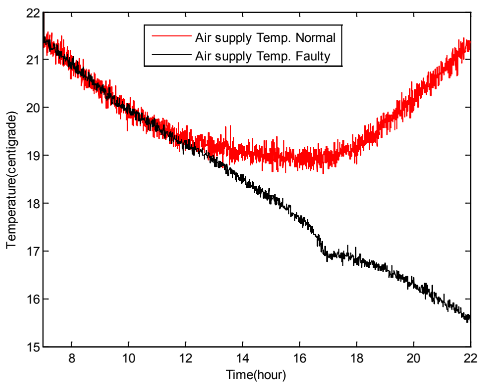


Figure 4.19 Sensitivity response of room temerature and wall temperature to fault type 3.1

As we can see our model detected this type of fault with four sensitive of variable consist of T_s, T_{cc}, T_{water_out} and f_{cc} in figures 4.17, 4.18 and 4.19. There are some differences between these responses to previous faults responses. These differences are in pattern style of features in low slope to fault shock. These characteristics will be useful for fault isolation in trainer algorithms.

In our last attempt for fault analysing we would like to test incipient failure of air fan supply with a higher rate drop as a description for fault 3.2. Higher rate of drop can be reach by lower final value of fan speed during same time. This is achieved by 50% of drop during 7 hours instead of 20% at same time. The response of model to this fault is clearly illustrated in figures 4.20, 4.21 and 4.22 for variables sensitivity.



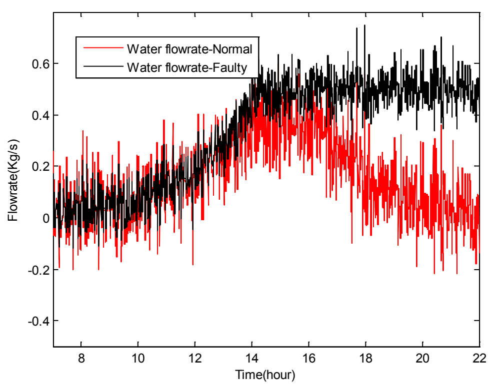


Figure 4.20 Sensitivity response of air supply temerature and cold water flow rate to fault type 3.2

Figure 4.20 which is based on air supply temperature and cold water flow rate analysis for an incident fault with final value of 50% of its nominal value. This time a big difference are shown in both graphs of air supply temperature and water flow rate. As we can see air supply temperature never could back to its original value and always takes far away from healthy model. We can see a big gap of 6℃ between healthy and faulty model at final time of simulation. On the other hand water flow rate try to compensate this fault by its maximum value but it could not reach any value more than 0.5 (kg/s) because of its limitation. However, it seems both variables have good characteristics to use in a training algorithm of fault detection specially air supply temperature due to its specific pattern reaction to this fault.

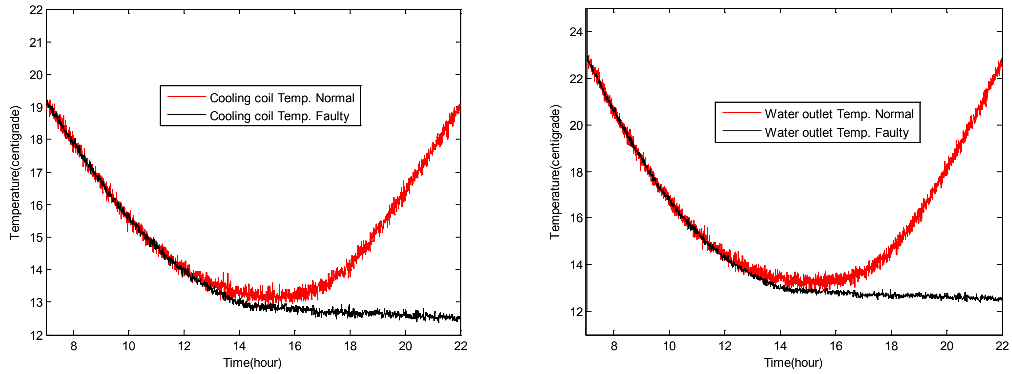


Figure 4.21 Sensitivity response of cold water oulet temerature and cooling coil temperature to fault type 3.2

Cooling coil temperature and water outlet temperature both follow a same pattern to this fault as illustrated in figure 4.21. Any of these two characters can be used for fault 3.2 due to their similarity in patterns. There is an interest point for a constant value which is compatible with the cold water flow rate reaction to this fault. As we can see both variables stay in 12.5℃ due to limitation of water flow rate which is stay on 0.5 (kg/s). These characteristics show us limitation of system and processes for compensating of some faults.

It seems room temperature and wall temperature can be sensitive to this fault as it account for a power full fault due to its strong effect. However, room environment characteristics shown in figure 4.22 both for wall and air temperature. We can see a significant change in room temperature compare to its set point and this because of lake in cold water flow rate which cannot compensate the supply fan fault. A big difference of 1.5℃ between room temperature and set point is an evidence for a predictable fault in the system. Also the pattern of room temperature is unique compare to other faults and can be used for diagnostic application in fault training. A difference in faulty record of wall temperature compare to its healthy model is an evidence for powered of the fault as we discussed before. Also it has a different pattern in compare with other faults which make it as a good potential for fault isolation.

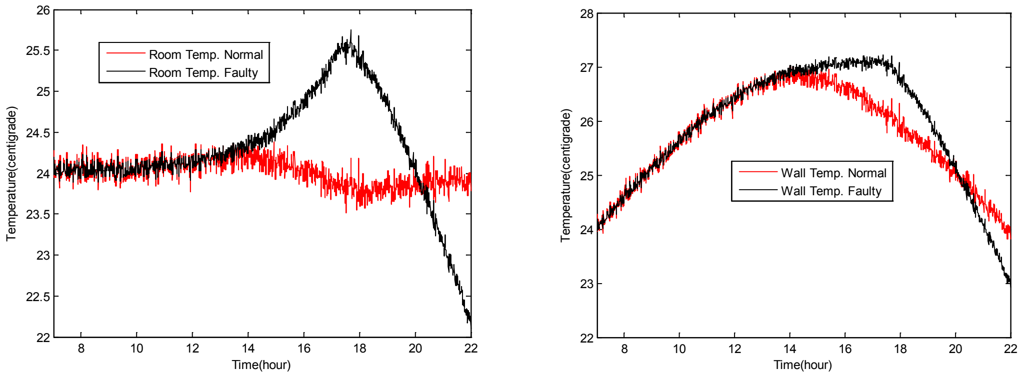


Figure 4.22 Sensitivity response of room temerature and wall temperature to fault type 3.2

It is perfectly detected by all six variables which have completely different response compare to healthy mode. These variations are demonstrated in magnitude as well as variable outline characteristics. Generally, we can consider all six variables as effective character for fault detection and isolation of fault type 3.2.

4.2.2 Conclusion

Three kinds of different pattern generated for major faults which are introduced based on fault review in past chapters. Then these patterns mapped to a famous and regular fault on HVAC system and then the effects have reviewed in next steps. Different types of faults applied for fan supply (as the critical part of AHU) in support of model validation for fault analysing purposes. Further to these three kinds of faults across fan supply the measuring noise is also considered by adding white noise with amplitude of 0.1 powers. These faults have been reviewed and related sensitive variable extracted as features for fault detector algorithm. By comparing the measured system states under normal and faulty conditions, it is found that the supply air, cooling coil and outlet water temperature are three sensitive features to all type of faults. Also some variables such as the room temperature, wall temperature and cooling water flow rate can be used as the auxiliary features to detect and isolating of the different faults.

Future research topics include more practical concerns and sub-system fault reviewing in details have been influenced by this simulation base on pre-fault analysing. In addition, this simulation could help us for any further sensor installation on a real HVAC case of fault detection. This important could reached by reviewing of all variables in the simulation model to find those which have potential to find an specific fault and then install new instrument for measure that variable if it is not installed previously. Algorithms such as principle component analysis (PCA), neural network and support vectors machine (SVM) in combination with this sensitivity analysis can be help to developing an automatic fault detection and isolation (AFDI). It is expected that indoor comfort level, energy-saving and safety of systems can be enhanced by integrating our analysis with an AFDI into the HVAC systems.

Chapter 5

5 Statistical Quantitative Diagnosis

5.1 Principle Component Analysis (PCA)

5.1.1 Introduction

This section attempts to introduce some promising multivariate statistical techniques that have been used for automated fault detection and diagnosis in different industries. Similar to HVAC systems, systems in chemical process operation and wastewater treatment also produce large amounts of real time process performance measurements at high frequency. Online statistical process control (SPC) is the primary tool traditionally used to improve process performance and reduce variation on key parameters. Due to the high number of measured variables, multivariate statistical techniques are used which provide capabilities to compress data and reduce data dimensionality so that essential information is retained and analysed more easily than the original huge data set. By dealing with all the variables simultaneously, multivariate methods extends traditional univariate control chart methods (Schein, 2003) for information extraction on the directionality of process variation. Principle component analysis (PCA) method is a standard multivariate technique to transform a number of related process variables to a smaller set of uncorrelated variables.

Kourti (2004) provided a good explanation for PCA method. When using a PCA method, principle components can be extracted by linearly combining the original input variables. PCA methods have been found to be useful in fault detection processes based on historical normal operating data. For fault diagnosis, however, they have been less powerful because of the non-casual nature of the data.

Lu et al. (2003) proposed a new process monitoring and diagnosis method that combined PCA to reduce the process dimension and wavelet analysis to extract the time-frequency process feature. A similarity measure was defined to compute the similarity degrees between pairs of process features for online fault diagnosis. Quantitative fault features were extracted from fault data by wavelet analysis method and saved in a database. The current fault is assumed to match an existing fault in the database if their fault features match. The method can not only detect abnormal conditions, but also differentiate faults with similar time domain characteristics, overcoming the major difficulty associated with the existing contribution plot method.

Yoon and MacGregor (2011) improved the PCA fault isolation method using additional data on past faults to supplement the models. This method extracted fault signatures that were vectors of movement of the fault in both the model space and the residual space. The directions of these vectors were then compared to the corresponding vector directions of known faults obtained from historical fault data. Fault isolation was then based on the observation of a joint plot of the angles between the vectors of the current fault and those of the known faults. Additional conclusion of this paper was that the number of principle components to be included in the PCA model will not result in any difference in the fault isolation step, or change in fault isolation ability.

5.1.2 Principle Component Analysis Fundamental

5.1.2.1 Data Reduction and Information Extraction

Principal Components Analysis is a favourite tool for data compression and information extraction. Generally, there is a great deal of correlated or redundant information in laboratory and process measurements. Essential information often lies not in any individual variable but in how the variables change with respect to one another; that is, how they co-vary. PCA provides combinations of variables that are more useful description of process conditions or events than individual variables. In this section, basic concepts about PCA are explained. Here, scalars are written as lowercase italics (x) and vectors as lowercase bold characters ($\mathbf{x}$). Uppercase bold characters ($\mathbf{X}$) represent matrices. The symbol "^{T}" refers to the transpose.

Let $\mathbf{X} \in \Re m \times n$ denote the raw data matrix with m samples (rows) and n variables (columns). The columns of **X** are normally first recommended for "autoscaling" (Ralston et al. 2001). Autoscaling means adjusting the value of X to have zero mean and unit variance by dividing each column by its standard deviation. This is because variables are measured with various means and standard deviations in different units. Autoscaling will put variables on an equal basis for the analysis (Ralston, 2011). Equation 3.1.1 gives the correlation matrix of **X**.

$$cov(X) = \frac{X^T X}{m-1} \tag{5.1}$$

Two algorithms, the NIPALS or singular value decomposition (SVD) algorithm (Wise and Gallagher 2006) are often used to decompose matrix X. The scaled matrix X is then decomposed as follows:

$$X = t_1 p_1^T + t_2 p_2^T + \cdots + t_k p_k^T + E \tag{5.2}$$

where t_i is scores vectors; p_i is loadings vectors, E is residual matrix. Therefore, PCA method reduces the original set of variables to k principal components and the remaining small variance factors are consolidated into a residual matrix (E). The scores vectors (t_i) form an orthogonal set ($t_i^T t_j$ for $i \neq j$) and contain information on how the samples (or conservations) relate to each other. The loadings vectors (p_i) are orthonormal ($p_i^T p_j = 0$ for $i \neq j$, $p_i^T p_j = 1$ for $i = j$) and contain information on how the variables (or measures) relate to each other. These variables (p_i) are the eigenvectors of the covariance matrix and the eigenvalue λ_i is a measure of the amount of variance corresponding to the eigenvector p_i.

$$\text{cov}(X) p_i = \lambda_i p_i \tag{5.3}$$

The loadings vectors (p_i) are considered as principal components, linear combination of the variables (columns of **X)**. The score vectors (t_i) represents the projection of each sample or observation onto the principal component axis. Therefore, for an *m* by *n* matrix **X**, at most n principal components could be computed. But due to correlation and noise, the first *k* principal components can accurately describe the major variances in the data already. There are many software that provide PCA tools. In this study, the PCA analysis is performed using the PLS_Toolbox 4.0 (Wise, 2006) with MATLAB.

5.1.2.2 Graphical Representation

The concept of principal components (PC) is shown graphically in Figure 5.1. Three dimensional data set lie mostly on a plane, thus the data is well described by a two PC model. The first PC represents the direction of the greatest variation in the data set, which is the major axis of the ellipse. The second PC aligns with the direction of second greatest variation and orthogonal to the first PC. In this case, a PCA model with two PC adequately describes all the variation in the measurements.

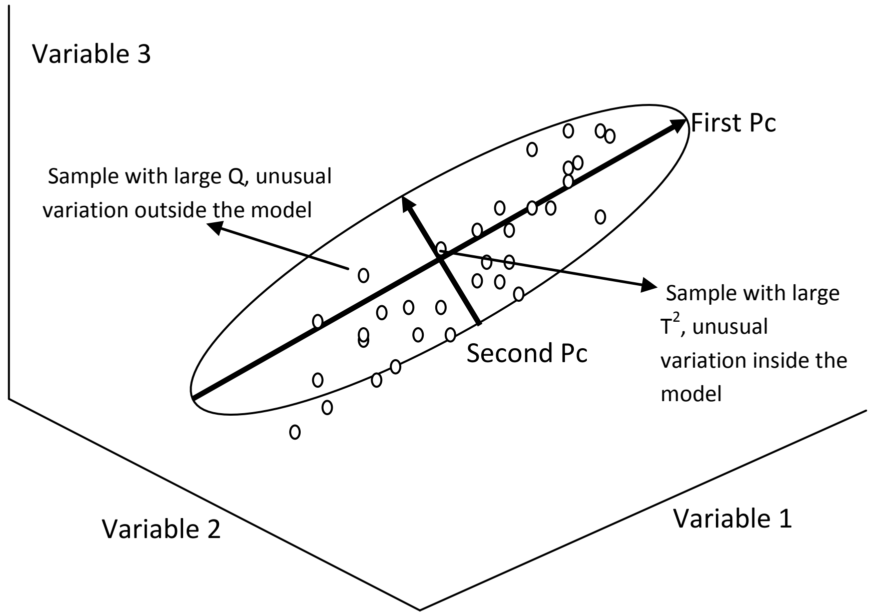


Figure 5.1 Graph presentation of PCA (Wise, 2006)

5.1.2.3 Statistics Associated with PCA Models

In general, linear static PCA based fault detection methods use two indices, Hotelling's T^2 statistic (sum of normalized squared scores) and the squared prediction error (SPE), also known as Q statistic. In Figure 5.1, T^2 is a measure of the distance from the multivariate mean to the projection of the operating point onto the 2 PCs. SPE is a measure of the distance off the plane formed by the 2 PC model. The T^2 limit defines an ellipse on the plane within which the operating point normally projects.

The SPE index is calculated using Eq. 5.4,

$$SPE = \|(I - PP^T)x\|^2 \leq \delta^2 \tag{5.4}$$

Where x is a new sample vector, δ^2 denotes the threshold (statistical confidence limit) for the *SPE* with a significance level α (Qin 2003).

The 100(1-α)% control limit for δ^2 is given by

$$\delta^2 = \left(\frac{s}{2m}\right)\chi_\alpha^2(2m^2/s) \tag{5.5}$$

where m and s are the sample mean and variance of the SPE values from the training data. T^2, known as Hotelling's T^2 statistic, is a measure of the variation in each sample within the PCA model. T_2 is defined as

$$T^2 = x^T P\,\lambda^{-1} P^T x \leq T_\alpha^2 \tag{5.6}$$

The 100(1-α)% control limit for T_α^2 is calculated by means of a *F*-distribution as

$$T_{\alpha}^{2} = \frac{k(m-1)}{m-k} F(k, m\text{-}1; \alpha) \quad (5.7)$$

where $F(k,\ m\text{-}1; \alpha)$ is a F-distribution with degree of freedom k and $m-1$ with level of significance α (MacGregor, 2009). Here m is the number of samples used to develop the PCA model and k is the number of principal component vectors retained in the model.

Once a fault is detected using the T^2 or SPE, the Q-contribution plot, which is a bar graph representing Q residual contribution (the significance of each variable on the index) versus variable number for certain sample, can be used to diagnose the fault. When the T^2or SPE breaks the threshold, the contribution of the individual variables to the T^2 or SPE can be identified, and the variable making a large contribution to the T^2 or SPE is indicated to be the potential fault source. In general, the Q-contribution plot helps to reduce the possible fault sources and thus focuses on a smaller range of measurements among all original variables.

5.1.3 Principle of Wavelet Analysis

5.1.3.1 Historical Review: From Fourier Analysis to Wavelet Analysis

The history of wavelets begins with the development of the traditional Fourier transform (FT) (Donald B., 2009), which is widely applied in signal analysis and image processing. Fourier transform breaks down a signal into the sum of infinite series of sines and cosines of different frequencies, in another word; FT is a mathematical technique for transforming our view of the signal from time-based to frequency-based. Fourier transform is very effective in problems dealing with frequency location. However, time information is lost during the process of transforming to frequency domain. This means that although we might be able to determine all the frequencies present in a signal, we do not know when they are present. In the time series process data, the most important part of the signal is the transient characteristics: drift, trends, and abrupt changes, and FT is not suited to detect them.

In an effort to improve the performance of the FT, the short time Fourier transform (STFT) has been developed in signal analysis. STSF compromises between the time and frequency based views of a signal by examining a signal under a fixed time window. The drawback of STSF is that the time window is fixed and same for all the frequencies. Many signals require a more flexible approach; the window size is required to vary according to the frequency.

Wavelet analysis or wavelet transform is close in spirit to the Fourier transform, but has a significant advance. It applies a windowing technique with variable-sized regions, a shorter time interval is used to analyse the high frequency components of a signal and a longer one to analyse the low frequency components of the signal. Wavelet analysis is very effective for dealing with local aspects of a signal, like trends, breakdown points, and self-similarity. Furthermore, wavelet analysis is capable of removing noise from signal and compress signal.

5.1.3.2 Wavelet Properties

Wavelets are a family of basis functions which are localized in the time and frequency domains, and are obtained from a single prototype wavelet, called mother wavelet or basic function Ψ(t), by scaling and translation (shifting). The wavelet family can be defined as

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\left(\frac{t-b}{a}\right) \tag{5.8}$$

where a and b represent the scale and translation parameters, respectively. In the discrete case, the scale and translation parameters are discretised as $a = 2^j$and $b = k2^j$. Ψ(t) can be rewritten as:

$$\Psi_{a,b}(t) = \Psi_{a,b}(t) = 2^{-j/2}\,\Psi(2^{-j}t - k) \tag{5.9}$$

where j and k denote the scale and translation parameters, respectively. The translation parameter determines the location of the wavelet in the time domain, while the scale parameter determines the location of the wavelet in the frequency domain. Given a function *x(t)*, the wavelet coefficients are obtained through the inner product operation:

$$W(a,b) = \int_{-\infty}^{+\infty} \Psi_{a,b}(t)x(t)dt \tag{5.10}$$

where j and k denote the scale and translation parameters, respectively. In general, wavelet analysis uses the wavelet functions which can be stretched and translated with a flexible resolution in both frequency and time. The flexible windows are adaptive to the entire time-frequency domain, which narrows while focusing on high frequency signals and widen while searching the low-frequency background. In this way, wavelet analysis allows the wavelets to be scaled to match most of the high and low frequency signal so as to achieve the optimal resolution with the least number of base functions.

The commonly used wavelet functions can be grouped into two main categories: continuous wavelets and discrete wavelets. The simplest discrete wavelet is the Haar wavelet, which is based on a box function. Daubechies wavelets with different orders are the most popular wavelets. In this study, Daubechies and Symlets wavelets are used for signal analysis. They are orthonormal with a compact support and capable of capturing smooth low-frequency features. Symlets are less asymmetric than Daubechies. There are other wavelet functions, including Coiflet, Mayer, Morlet, and Mexican Hat. The wavelets are chosen based on their shape and their ability to analyze the signal in a particular application.

5.1.3.3 Wavelet Transform Decomposition

The common application of wavelet transform is signal decomposition. MATLAB (2010) developed a recursive algorithm to decompose and reconstruct a signal using wavelet transform method. This method connects the continuous time multi-resolution to the discrete time filters. As shown in Figure 5.2, the wavelet transform can be used to decompose multivariate signals s into approximations a_1 and details d_1 coefficients at the first level.

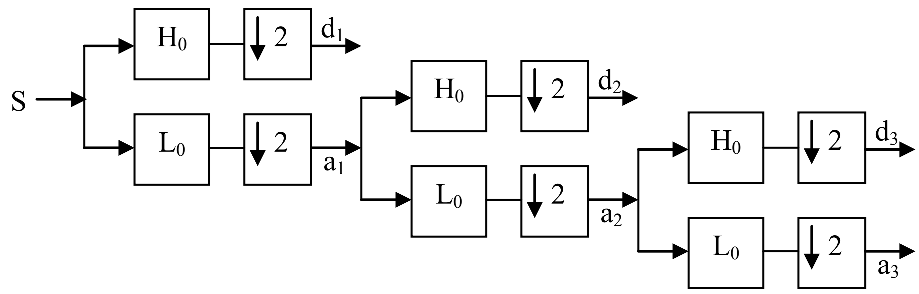


Figure 5.2 Wavelet decomposition tree (three level)

Application of the same transform on the approximations a_1 causes them to be decomposed further into approximations a_2 and details d_2 coefficients at the second level. The decomposition process can continue to a level L as long as the length of approximation coefficients in a_l is more than the length of coefficients in the wavelet filter. The wavelets transform works like a filter. After passing a signal through a wavelet transform filter, wavelet coefficients are generated. The wavelet transform contains a low pass filter (only obtaining low frequencies), which is denoted by L_0, and a high pass filter (only obtaining high frequencies), which is denoted by H_0. At each level, the original signal, s, passes through two both low and high pass filters and emerges as two signals, which is detail coefficients d_n, and approximation coefficients a_n. The terms “approximation” and “detail” are named by the fact that a_n. is the approximation of a_{n-1} corresponding to the “low frequencies” of a_{n-1}, whereas the detail d_n takes into account its “high frequencies”. Wavelet coefficients at various frequencies reflect the signal variations at those frequencies and corresponding times. Clearly, at the kth step of this partitioning procedure, the original signal, s, is expressed by Eq. 5.11.

$$s = a_k + d_k + d_{k-1} + \cdots + +d_1 \tag{5.11}$$

Which a_k representing the smooth signals referring to the time scale 2_k and d_n is the detail of time series with the time scale located in the interval [2_{k-1}, 2_k].

One of the most important applications of wavelet analysis is the compression of signals. The “true” signal tends to dominate the low-frequency area. The common approach to filtering is to remove the high-frequency components above a certain level since they represent the detailed information in the signal. The approximation coefficients resemble a moving average. The lower frequency approximation coefficient gives the identity of the signal and reflects the signal trend. The number of wavelet coefficients decreases by a factor of 2 at coarser scales. In this way, the original signal vector is smoothed and halved through the low pass filters. This reaches the objective of signal compression.

The discrete wavelet transform can be used to analyse or decompose signals. The inverse process of the discrete wavelet transform is called signal reconstruction, how those components can be assembled back into the original signal without loss of information. Whilst wavelet analysis involves filtering and down-sampling, the wavelet reconstruction process consists of filtering and up-sampling. Up-sampling is the process of lengthening a signal component by inserting zeros between samples. Wavelet analysis is still a new and emerging field; more possible unknown applications of wavelet analysis are waiting for exploring.

5.1.4 Pattern Matching in Historical Data

5.1.4.1 Introduction of Pattern Matching in Time-series Data

Large numbers of process measurements are stored in an EMCS database which provides a valuable source of process information. Building HVAC systems usually operate year-round and hundreds of variables are measured and recorded on a frequent basis, as often as every one to five minutes. However, the information contained in these databases has been underutilized for many reasons. The largest obstacle to extract useful information from historical data is simply the enormous amount of data that must be analysed. In this dissertation, a pattern matching methodology is developed for analysing AHU data. The proposed methodology provides a preliminary screening of historical data in order to locate periods of operation that are similar to current operating conditions. When an AHU is operated under similar weather and internal load conditions, the process variables, such as supply air flow rate, operation of dampers and valves, are also similar. If the past periods with similar operating conditions can be located from historic data, the data may provide useful information for detecting and diagnosing the current operation. For example, if under certain weather and internal conditions, OA damper position from historical data is always 40% open, but current OA damper position is 100% open. Then this information can serve as a warning sign that perhaps a economizer control fault or OA temperature sensor fault exists.

The ability to locate data with similar operating conditions will be advantageous for difficult AHU AFFD problems. Ideally, the new methodology should not require a priori knowledge of the operating systems and should require minimal computer time. A number of techniques have been developed for pattern matching in time-series data (Singhal, 2002a; Singhal, 2002b; Johannesmeyer, 2002). Automated pattern matching methodology requires only minimal information from the user. In particular, the user only has to supply the relevant process variables and their measurements for the duration of current examination data. This data is referred to as the snapshot data and serves as a template for searching the historical database. Furthermore, the historical data is sliced into data windows with fixed length. Thus, the data windows contain the same variables and the same number of samples as the snapshot data. Next, similarity factors (defined later) are calculated to characterize the degree of similarity between each historical data window and the snapshot. The historical datasets that have the highest values of the similarity factors are selected for a candidate pool, which is available for subsequent analysis to gain further insight in system operation. A primary research issue is how to calculate the similarity factors which are used to quantify the similarity between the snapshot data and historical data window.

5.1.4.2 PCA Similarity Factor

Krzanowski (2008) developed a method for quantifying the similarity of two data sets using a PCA similarity factor, S_{PCA}. Given two data sets contain the same n variables, respectively, a PCA model can be built for each data set. The corresponding PCA model has k principal components, where $k \leq n$. The similarity between the two data sets is then quantified by comparing the k principal components for each data set. The appeal of this approach is that the similarity between two data sets is quantified with one single number, S_{PCA}.

Consider a current snapshot data set ***S*** and a historical data set ***H*** with each data set consisting of *m* measurements of the same *n* variables. Let k_1 and k_2 be the number of principal components that describe at least 95% of the variance in data set ***S*** and ***H***, respectively. Let *k* is the maximum of k_1 and k_2, which ensures that *k* principal components describe at least 95% of the variance in each data set. Then by selecting only the first *k* principal components for each data set, subspaces of the ***S*** and ***H*** data sets can be constructed.

The corresponding ($n \times k$) principal component subspaces are denoted by ***L*** and ***M***, respectively. The matrices ***L*** and ***M*** are also the eigenvector matrices corresponding to the first *k* eigenvalues of the covariance matrices of ***S*** and ***H***, respectively. The PCA similarity factor compares these reduced subspaces and can be calculated from the angles between principal components (Krzanowski, 2008).

$$S_{PCA} = \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{k} cos^2\theta_{ij} \tag{5.12}$$

where θ_{ij} is the angle between the *i*th PC of data set ***S*** and the *j*th PC of data set ***H***. It can also be expressed in terms of the subspaces ***L*** and ***M*** as (Krzanowski, 2008):

$$S_{PCA} = \frac{trace(L^T M M^T L)}{k} \tag{5.13}$$

Because ***L*** and ***M*** contain the *k* most significant principal components for ***S*** and ***H***, S_{PCA} is also a measure of the similarity between data sets ***S*** and ***H***. The similarity level between two data sets ***S*** and ***H*** can be evaluated by the value of S_{PCA}.

Through comparing the spaces spanned by ***L*** and ***M***, the first *k* PCs for each ***S*** and ***H*** data sets are given equal weight. But because the amount of variance described by each of the *k* principal components varies significantly, this equal weighting may be inappropriate in many cases. A more informative measure of the similarity between two PCA models should take into account the variance explained by each principal component direction.

Consequently, a modified PCA similarity factor, S_{PCA}^{λ}, has been proposed that weights each principal component by the square root of its corresponding eigenvalue. The modified PCA similarity factor for two datasets ***S*** and ***H***, is calculated as,

$$S_{PCA}^{\lambda} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k}(\lambda_i^l \lambda_j^m) cos^2\theta_{ij}}{\sum_{i=1}^{k}\lambda_i^l \lambda_i^m} \tag{5.14}$$

where λ_l and λ_m are the eigenvalues of ***L*** and ***M***, respectively. The cosine squared of the angle between the *i*th principal component of ***L*** and the *j*th principal component of ***M*** is now weighted by the product of the *i*th eigenvalue of ***L*** and the *j*th eigenvalue of ***M***. Thus,

S_{PCA}^{λ}, quantifies the similarity between the $\boldsymbol{L}$ and $\boldsymbol{M}$ subspaces and lies between zero and one.

5.1.4.3 Distance Similarity Factor

Abstract distances between items or collections of items are important measure implemented in multivariate databases. Euclidean distance is a simple algorithm to calculate the geometric distance: square the difference in each dimension (variable), and take the square root of the sum of these squared differences. This distance measure has a straightforward geometric interpretation and is fast to calculate, but it has two basic limitations. First, the Euclidean distance is extremely sensitive to the scales of the variables involved. All variables are measured in the same units of length in geometric situations, but this is likely not the case with other data. Variables with many different scales would be dealt with in HVAC systems, such as temperature, pressure gauge, power, etc. The scales of these variables are not comparable. Second, the Euclidean distance is blind to correlated variables. Consider a data set containing four variables, where one variable is an exact copy of one of the others. Two variables are thus completely correlated. Although the copy will not bring any new information, Euclidean distance intends to weight the copied variable more heavily in its calculations than the other variables.

The Mahalanobis distance is a useful way of determining similarity measure of a single sample to a data set. It differs from Euclidean distance in that it takes into account the correlations among the variables and is scale-invariant. The problems of scale and correlation inherent in the Euclidean distance are no longer an issue for this measure.

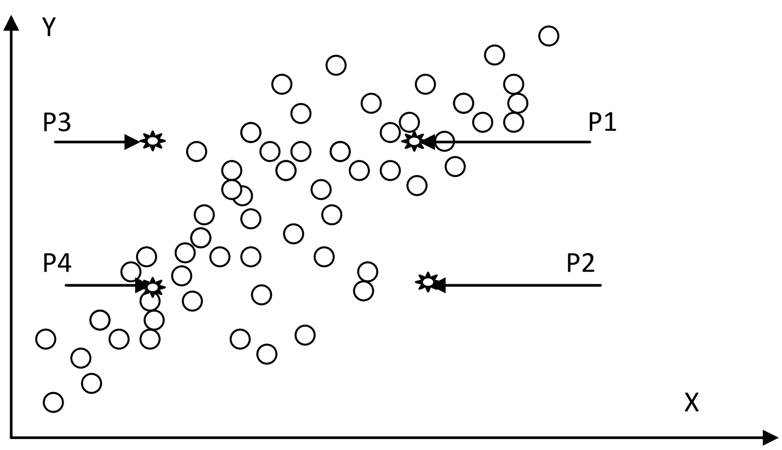


Figure 5.3 Similarity distance (scatter plot)

Consider the data graphed in the following Figure 5.3. The circle points make up of a data set, ***S***, the mean point of the data set is close to the centre of coordinates. Table 3.1 lists the calculation results of the Euclidean distance and Mahalanobis distance between four asterisk points and ***S***. It is easy to be observed in Figure 5.3 that points P2 and P3 are outliers of data set ***S*** comparing to points P1 and P4. The Mahalanobis distances calculated in Table 5.1 are in agreement with the estimation, but Euclidean distances are not able to discern among the four points. The example easily concludes that the Mahalanobis distance is more appropriate to measure the similarity than Euclidean distance by taking into account of the covariance of the data and the scales of the different variables.

Table 5.1 Results for Euclidean distance and Mahalanobis distance

	P1	P2	P3	P4
Coordinates	(1,1)	(1,-1)	(-1,1)	(-1,-1)
Euclidean distance	2.3173	2.0187	2.0047	1.7061
Mahalanobis distance	1.3294	20.2278	20.0652	0.9783

The Mahalanobis distance is a very useful way of determining the "similarity" of an observation sample to a snapshot dataset. The centres of the snapshot dataset, $\overline{x_S}$, is defined as the sample means

$$\overline{x_S} = \frac{1}{m_S}\sum_{i=1}^{m_S} x_i, \qquad x_i \in S \tag{5.15}$$

where m_S is the number of observations in datasets *S*, and x_i is an observation in *S*. The Mahalanobis distance Φ from the centre of the snapshot dataset, $\overline{x_S}$ to the observation sample x_H is calculated as (Mardia, 2006)

$$\Phi = \sqrt{(x_H - \overline{x_S})^T \sum\nolimits_S^{*-1} (x_H - \overline{x_S})} \tag{5.16}$$

where matrix $\sum_S^{*-1}$ is the pseudoinverse of $\sum_S$, the covariance matrix of dataset *S*. It can be calculated using singular-value decomposition. The distance similarity factor is defined as the probability that x_H is at least a distance Φ from $\overline{x_S}$. Assuming a Gaussian probability distribution:

$$S_{dist} = d\sqrt{\frac{2}{\pi}}\int_{\Phi}^{\infty} e^{-z^2/2}dz = 2 \times \left[1 - \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\Phi} e^{-z^2/2}dz\right] \tag{5.17}$$

The number of singular values used to calculate the pseudo-inverse is specified to be the minimum number of principal components needed to explain at least 95% of the variance in each dataset. The novelty of the distance similarity factor is that it assigns a probability value lying between 0 and 1 to the Mahalanobis distance between an observation samples to a snapshot dataset by using the Gaussian probability distribution.

5.1.4.4 Moving–Window Approach for Selection of the Candidate Pool

In Sec. 5.1.4.2, the similarity between the snapshot and historical data can be characterized by the modified PCA similarity factor, S_{PCA}^{λ}. The historical data set must be divided into windows with the same size as the snapshot data for the application of this pattern similarity methodology. Therefore, a systematic method for defining windows of data in the historical database must be developed. The method chosen will have a significant impact on the success of the search.

A simple method of selecting windows of historical data is to use a constant window size moving through the historical data. The window size is equal to the length of the snapshot data. The historical data set is then divided by placing windows side by side along the time axis resulting in equal length, non-overlapping segments of data. For example, a window size of 100 minutes moving through 1000 minutes of historical data results in ten data segments. However, there is shortcoming associated with this approach. If the phenomenon of interest starts in the middle of one window and ends in the middle of next window, its effect could be “averaged out”. Thus, the occurrence could be missed even though it is relevant.

In order to locate phenomena at various starting points in the historical database, a sliding window could be adopted. The similarity of the data contained in the window could be compared to the snapshot data as the window slides back through the historical database. The rate at which the window is moved through historical data determines the accuracy of pattern matching and the computational load required. To reduce the computational load, the moving window can be moved by w observations at a time, so that w new observations replace the w oldest observations each time. The integer w is referred to as the window movement rate. Choosing w to be one-tenth to one-fifth of the length of the snapshot data window provides a satisfactory trade-off between the accuracy of pattern matching and the computational load required.

Furthermore, the similarity factors are sorted in decreasing order to create a rank-ordered list of the historical data windows. The historical data windows with the largest values of similarity factors are then collected in a candidate pool. To avoid redundant counting of data windows that represent the same period of operation in the simulated historical database, in case that historical data windows that are "too close" to each other, the historical data window with the largest values of the similarity factors is chosen among the overlapped historical data windows. This restriction results in the selection of data windows that are at least *m* observations apart, where *m* is the length of duration in the snapshot data.

5.1.5 PCA Model Development

In this study, PCA models are developed using PLS Toolbox in MATLAB Version 2010b. Basic steps to building a PCA model using PLS Toolbox are: 1) data loading, which means importing data into the MATLAB environment; 2) autoscaling; 3) determining the number of principle components (PCs). Autoscaling (mean-centres and scales the columns to unit variance) is a necessary step when data contains variables that have different variance and units. For example, a data set may contain temperature, pressure, and humidity measurements, each of which has its own average value and variance. The autoscaling approach enables all variables on an equal basis in the analysis.

After data are autoscaled, the next step is to determine the number of PCs needed in the model. Rule of thumb and knowledge about the data are generally needed here. Eigenvalues of a dataset are useful when deciding the number of PCs, because eigenvalues are the number of independent variables each PC contains. A good rule of thumb is that any PC with an eigenvalue less than 1 is probably not describing any systematic variance and is not likely to be included in the model.

Table 5.2 Percent variance captured by PCA model

Principle component	Eigenvalue of cov(X)	% Variance this PC	% Variance cumulative
1	4.63	46.25	46.25
2	2.59	25.92	72.17
3	1.25	12.51	84.68
4	0.827	8.27	92.95
5	0.416	4.16	97.11
6	0.137	1.37	98.48
7	0.0838	0.84	99.32
8	0.0312	0.31	99.63
9	0.0268	0.27	99.9
10	0.0103	2 0.10	100

To demonstrate the development of a PCA model, 770 measurement samples for 12 variables are selected from an experiment conducted at the University of Technology Sydney (UTS) HVAC laboratory test facility between 8:00 to 18:00 during a sunny week.

The PCA model is built using twelve typical AHU variables including Outside air temperature, Mix air temperature, Supply air temperature, Supply air duct static pressure, Supply airflow rate, Supply fan total power meter, Return fan total power meter, Supply fan speed signal, Return fan speed signal, Mixing box damper position signal, Heating coil valve position signal and Cooling coil valve position signal. Table 5.2 is percent variance captured by the PCA model, the best choice in this case is to select three PCs because three PCs cumulatively cover 84.68% of the total variance and the third PC captures only 12.51% of the variance.

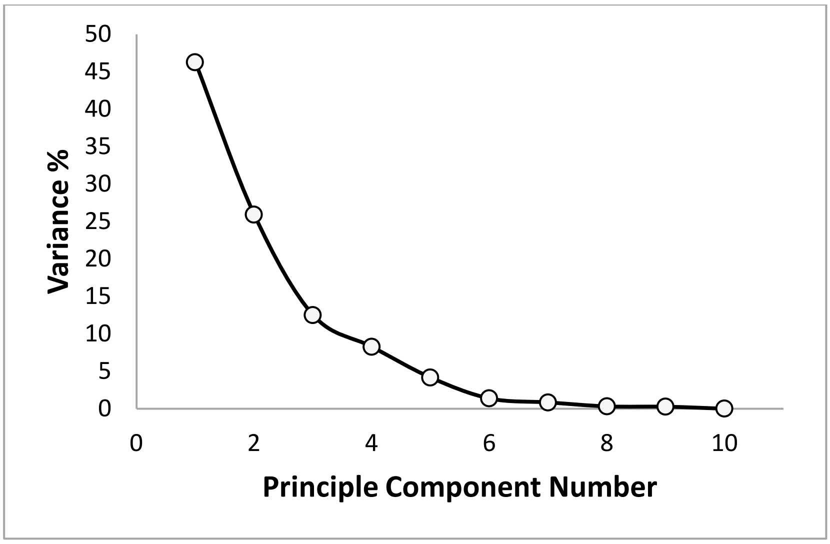


Figure 5.4 Eigenvalues and cross validation for AHU

Another approach is to plot the relationship between the eigenvalues and PCs, shown in Figure 5.4 (from PCA model shown in Table 5.2). By looking for a sudden jump in the eigenvalue, a “knee” in the line, one can see how the ratio of successive eigenvalues varies. A “knee” occurs at PC number three for the data presented in Table 5.2, indicating that retaining three PCs in the model is a good choice.

It is also possible to choose the number of PCs based on a cross-validation procedure. In this procedure, a continuous block of 10% of the training data is set aside in turn and the remaining 90% of the data is used in modelling. A PCA model is then built on all but the left out training data. The model is then used to estimate variables with 10% of the training data.

5.2 Support Vector Machine (SVM) and Classification

5.2.1 Introduction

Support vector machine (SVM) was originally introduced by Vapnik (1998) and co-workers in the late 1990s. While traditional statistical theory keeps to empirical risk minimization (ERM), SVM satisfies structural risk minimization (SRM) based on statistical learning theory (SLT), whose decision rule could still obtain small error to independent test sampling. SVM mainly has two classes of applications, classification and regression. In this paper, application of classification is discussed.

SVM have been extensively researched in the data mining and machine learning communities for the two last decades and actively applied to applications in various domains. SVM are typically used for learning classification and regression functions, for which two first concepts are called classifying (SVM), support vector regression (SVR) respectively. Cristianini (2000) provided a good explanation for SVM method and kernel application. This section introduces these general concepts and techniques of SVM for learning classification and regression functions. SVM models are a close cousin to classical multilayer perceptron neural networks. Using a kernel function, SVM's are an alternative training method for polynomial, radial basis function and multi-layer perceptron classifiers in which the weights of the network are found by solving a quadratic programming problem with linear constraints, rather than by solving a non-convex, unconstrained minimization problem as in standard neural network training. The main advantages of the SVM approach are as follows.

• SVMs implement a form of structural risk minimization. They attempt to find a compromise between the minimization of empirical risk and the prevention of over fitting.

• The problem is a convex quadratic programming problem. So there are no nonglobal minima, and the problem is readily solvable using quadratic programming techniques.

• The resulting classifier can be specified completely in terms of its support vectors and kernel function type.

Usually, SVMs are trained using a batch model (offline method). Under this model, all training data is given a priori and training is performed in one batch. If more training data is later obtained, or we wish to test different constraint parameters, the SVM must be retrained from scratch. But if we are adding a small amount of data to a large training set, assuming that the problem is well posed, then it will likely have only a minimal effect on the decision surface. Resolving the problem from scratch seems computationally wasteful. An alternative is to “warm-start” the solution process by using the old solution as a starting point to find a new solution. This approach is at the heart of active set optimization methods and, in fact, incremental learning is a natural extension of these methods (online training). While many papers have been published on SVM training, relatively few have considered the problem of incremental training.

5.2.2 Introduction to SVM

For easy understanding of SVM, first we refer to a classification problem of two dimensions. Consider we have two types (class) of data that are not mixed together in a 2-D space. Figure 5.5 shows these data with fill circle and non-fill circle. If we try to separate them with a line, so we have several choices of lines for separation but important thing is which one is the best. It is related to us, that what we like to do. If we define a classification problem, so the desired subject is to separation data with maximum distance together. It means finding the best line with maximum distances from data.

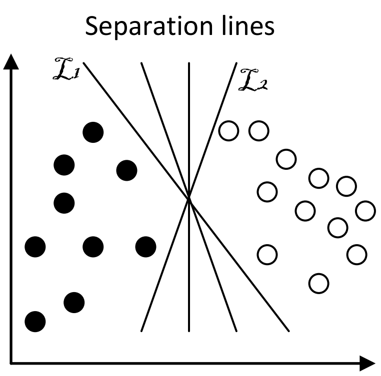


Figure 5.5 Two group of data in 2-D

But when we choose a line with maximum distance from one data, other distances between line and other data are not in maximum distances. This constrains lead to an optimization problem to find the best line for maximum possible distances between line and data. First try for a good separation is to find some of data which are closed to other group. As we can see in figure 5.6 the best selection with above consideration are a,b for solid circles and α,β for non-fill circles.

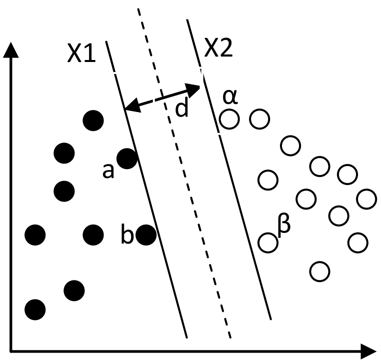


Figure 5.6 Best separation line selections

In the next step we could select a closest line to each group with consider to maximum distance to other group. In fact we try to find two lines with maximum vertical distance together which each line has minimum distance (d) to each group. It seems the lines X1 and X2 are good candidate for this purpose. The final section is to find a line between X1 and X2 which is the best line for separation.

Our previous example was very simple state for classification. In fact in most cases our data are combined together with some complexity. For example in figure 5.6 you could see two types of data with more complexity of combining. It seems we could not separate them with a line and we need some curve function to separation. With selection a curve instead of line the separation problem guide to nonlinear problem. A method which we could used for getting a linear separation is mapping of data with some function to another state which we could separation our data in new state with a linear function. It is clear in figure 5.7 some data which define with closed curve are mapped with function of φ to a comfortable state of linear separation.

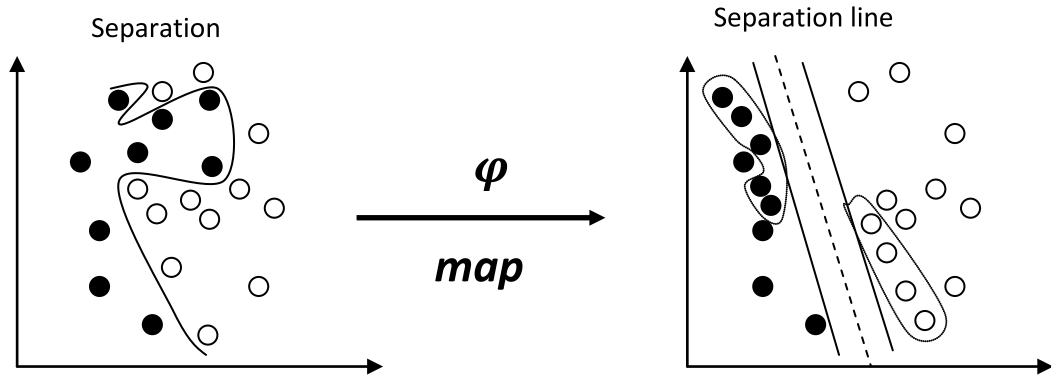


Figure 5.7 Mapping nonlinear separations to linear

Due to complexity of separation in some cases we have to permit to separation algorithm to have a minimum error for separation. This permit to algorithm for achieved more distance for separation rather than with zero error. Figure 5.8 shows a simple define of penalty which allowed to one data (a) having an error form separation point.

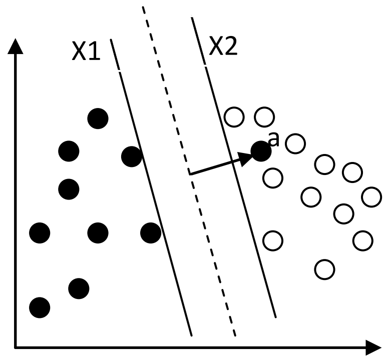


Figure 5.8 Error cost function for separation

To allow to this flexibility in separating the categories, we could define a cost parameter, C, that controls the tradeoff between allowing training errors and forcing rigid of lines X1 and X2 which lead to some misclassifications. Increasing the value of C increases the cost of misclassifying data and forces the creation of a more accurate model that may not generalize well.

Above example we try to define a classification problem with a mathematical algorithm. In fact this algorithm defines as Support Vector Classification Machine (SVCM). In SVCM some data (in our example a,b and α,β) which used for finding two lines X1 and X2 called Support Vectors. The dashed line between X1 and X2 called hyperplane. Also vertical distance (d) between two lines X1 and X2 called margin and mapping function φ called as kernel function.

5.2.3 Incremental Learning development through the SVM

The initial form of support vector classification machine (SVCM) is a binary classifier where the output of learned function is either positive or negative (Bartlett; Bennett, 2000). A multiclass classification can be implemented by combining multiple binary classifiers using pairwise coupling method. This section explains the motivation and formalization of SVCM as a binary classifier, and the two key properties-margin maximization and kernel trick.

Binary SVCMs are classifiers which discriminate data points of two categories. Each data object (or data point) is represented by a n-dimensional vector. Each of these data points belongs to only one of two classes. A linear classifier separates them with a hyperplane (Chew, 2001). Figure 1 shows two groups of data and separating hyperplanes that are lines in a two-dimensional space. There are many linear classifiers that correctly classify (or divide) the two groups of data such as L1 and L2 in Figure 1. As we can see in Figure 2, SVCM can achieve maximum separation between the two classes; it picks the hyperplane which has the largest margin. The margin is the summation of the shortest distance from the separating hyperplane to the nearest data point of both categories. Such a hyperplane is likely to generalize better, meaning that the hyperplane correctly classify "unseen" or testing data points. It is clearly illustrate in Figure 5.5 that SVCM does the mapping from input space to feature space to support nonlinear classification problems. The kernel trick is helpful for doing this by allowing the absence of the exact formulation of mapping function which could cause the issue of curse of dimensionality. This makes a linear classification in the new space (or the feature space) equivalent to nonlinear classification in the original space (or the input space). SVM gets help by mapping input vectors to a higher dimensional space (or feature space) where a maximal separating hyperplane is constructed (Chew, 2003).

5.2.3.1 Hard-Margin SVCM Classification for Incremental Learning

To understand how SVCMs compute the hyperplane of maximal margin and support nonlinear classification, the hard-margin SVCM explained where the training data is free of noise and can be correctly classified by a linear function. The data points S in Figure 5.6 (or training set) can be expressed mathematically as follows:

$$S = \{(x_1, y_1), \dots, (x_n y_n)\} \quad (5.18)$$

Where x_i is an n-dimensional real vector, y_i is either 1 or -1 denoting the class to which the point xi belongs. The SVCM classification function F(x) takes the form:

$$F(\boldsymbol{x}) = \boldsymbol{w}.\boldsymbol{x} - b \quad (5.19)$$

w is the weight vector and b is the bias, which will be computed by SVCM in the training process. First, to correctly classify the training set, F(0) (or w and b) must return positive numbers for positive data points and negative numbers otherwise, that is, for every point xi in S, $\mathrm{w.x} - \mathrm{b} > 0 \ if\ \mathrm{y_i} = 1$ and $\mathrm{w.x} - \mathrm{b} < 0 \ if\ \mathrm{y_i} = -1$ These conditions can be revised into:

$$y_i(w.x - b) > 0 \quad \forall (x_i, y_i) \in S \quad (5.20)$$

If there exists such a linear function F that correctly classifies every point in S or satisfies Eq.(5.20), S is called linearly separable (Sch¨olkopf, 2008). Second, F (or the hyperplane) needs to maximize the margin. Margin is the distance from the hyperplane to the closest data points.

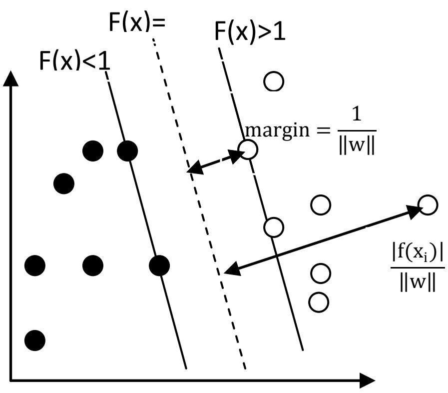


Figure 5.9 the hyperplane maximizing the margin in a two-dimensional

An example of such hyperplane is illustrated in Figure 5.9. To achieve this, Eq.(5.20) is revised into the following Eq.(5.21).

$$y_i(w.x - b) \geq 1 \quad \forall (x_i, y_i) \in S \tag{5.21}$$

Note that Eq.(3.4) includes equality sign, and the right side becomes 1 instead of 0. If S is linearly separable, or every point in S satisfies Eq.(5.20), then there exists such a F that satisfies Eq.(5.21). It is because, if there exist such w and b that satisfy Eq.(5.20), they can be always rescaled to satisfy Eq.(5.21) (Sch¨olkopf, 2007). The distance from the hyperplane to a vector x_i is formulated as $\frac{|f(x_i|}{||w||}$. Thus, the margin becomes

$$margin = \frac{1}{||w||} \tag{5.22}$$

because when x_i are the closest vectors, F(x) will return 1 according to Eq.(5.22). The closest vectors, that satisfy Eq.(5.21) with equality sign, are called support vectors. Maximizing the margin becomes minimizing ||w||. Thus, the training problem in SVCM becomes a constrained optimization problem as follows.

$$Q(\boldsymbol{w}) = \frac{1}{2} ||\boldsymbol{w}||^2 \tag{5.23}$$

$$subject\ to \qquad y_i(\boldsymbol{w}.\boldsymbol{x} - b) \geq 1 \quad \forall (\boldsymbol{x}_i, y_i) \in S \tag{5.24}$$

The factor of 1/2 is used for mathematical convenience. The constrained optimization problem (5.23) and (5.24) is called primal problem. It is characterized as follows:

• The objective function (5.23) is a convex function of w.

• The constraints are linear in w.

Accordingly, for solve the constrained optimization problem using the method of Lagrange multipliers (Sch¨olkopf, 2002). After solving in Lagrange condition we get:

$$\mathrm{F}(\mathbf{x}) = \sum_{\mathrm{i}} \alpha_{\mathrm{i}} \mathrm{y}_{\mathrm{i}} \mathbf{x}_{\mathrm{i}}.\mathbf{x} - \mathrm{b} \tag{5.25}$$

Where the auxiliary nonnegative variables α are called Lagrange multipliers.

5.2.3.2 Soft-Margin SVCM Classification

The discussion so far has focused on linearly separable cases. However, the optimization problem (5.23) and (5.24) will not have a solution if D is not linearly separable. To deal with such cases, soft margin SVCM allows mislabelled data points while still maximizing the margin. The method introduces slack variables, ξi, which measure the degree of misclassification. The following is the optimization problem for soft margin SVCM.

$$minimize: Q1(w, b, \xi_i) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \quad (5.26)$$

$$subject\ to \quad y_i(\boldsymbol{w}.\boldsymbol{x} - b) \geq 1 - \xi_i, \ \forall(\boldsymbol{x}_i, y_i) \in S \quad (5.27)$$

5.2.3.3 Kernel Trick for Incremental Learning of Nonlinear Classification

If the training data is not linearly separable, there is no straight hyperplane that can separate the classes. In order to learn a nonlinear function in that case, linear SVCMs must be extended to nonlinear SVCMs for the classification of nonlinearly separable data (Scholkopf, 2008).

The process of finding classification functions using nonlinear SVCMs consists of two steps. First, the input vectors are transformed into high-dimensional feature vectors where the training data can be linearly separated. Then, SVCMs are used to find the hyperplane of maximal margin in the new feature space. The separating hyperplane becomes a linear function in the transformed feature space but a nonlinear function in the original input space.

Let **x** be a vector in the *n*-dimensional input space and φ be a nonlinear mapping function from the input space to the high-dimensional feature space. The hyperplane representing the decision boundary in the feature space is defined as follows.

$$\boldsymbol{w} \cdot \varphi\ (\boldsymbol{x}) - b = 0 \quad (5.28)$$

where **w** denotes a weight vector that can map the training data in the high dimensional feature space to the output space, and b is the bias. Using the φ function, the weight becomes:

$$F(\boldsymbol{x}) = \sum_i \alpha_i y_i \varphi(\boldsymbol{x}_i).\varphi(\boldsymbol{x}_j) - b \quad (5.29)$$

Note that the feature mapping functions in the optimization problem and also in the classifying function always appear as dot products, e.g., $\varphi\ (x_i) \cdot \varphi\ (x_j)$ is the inner product between pairs of vectors in the transformed feature space. Computing the inner product in the transformed feature space seems to be quite complex and suffer from the curse of dimensionality problem. To avoid this problem, the kernel trick is used. The kernel trick replaces the inner product in the feature space with a kernel function K in the original input space as follows.

$$K(u,v) = \varphi\ (u) \cdot \varphi\ (v) \tag{5.30}$$

$$F(\boldsymbol{x}) = \sum_{i} \alpha_i y_i \boldsymbol{K}(\boldsymbol{x}_i . \boldsymbol{x}_j) - b \tag{5.31}$$

Since K is computed in the input space, no feature transformation will be actually done or no φ will be computed, and thus the weight vector $\mathbf{w} = \alpha_i y_i \varphi(\mathbf{x}_i)$ will not be computed either in nonlinear SVCMs.

Note that, the kernel function is a kind of similarity function between two vectors where the function output is maximized when the two vectors become equivalent (Smola, 2000). Because of this, SVCM can learn a function from any shapes of data beyond vectors (such as trees or graphs) as long as we can compute a similarity function between any pairs of data objects. Further discussions on the properties of these kernel functions are out of the scope. We will instead give an example of using polynomial kernel for learning an XOR function in the following section.

5.2.4 Combine Incremental-Decremental Learning

The main advantages of SVM include the usage of kernel trick (no need to know the non-linear mapping function), the global optimal solution (quadratic problem), and the generalization capability obtained by optimizing the margin (Cerqueira, 2008). However, for very large datasets, standard numeric techniques for QP become infeasible. An on-line alternative, that formulates the (exact) solution for l+1 training data in terms of that for l data and one new data point, is presented in online incremental method. Training an SVM incrementally on new data by discarding all previous data except their support vectors, gives only approximate results (Syed, 2009). Cauwenberghs (2001) consider incremental learning as an exact on-line method to construct the solution recursively, one point at a time. The key is to retain the Kuhn-Tucker (KT) conditions on all previous data, while adiabatically adding a new data point to the solution. Leave-one-out is a standard procedure in predicting the generalization power of a trained classifier, both from a theoretical and empirical perspective.

Giving n data, $S = \{x_i, y_i\}$ and $y_i \in \{-1, +1\}$ where x_i represents the condition attributes, y_i is the class label (correct label is +1 and faulty label is -1), and i is the number of data for train. The decision hyperplane of SVM can be defined as (w,b), where w is a weight vector and b a bias. Let w_0 and b_0 denote the optimal values of the weight vector and bias. Correspondingly, the optimal hyperplane can be written as:

$$w_0^T + b_0 = 0 \tag{5.32}$$

To find the optimum values of w and b, it is required to solve the following optimization problem:

$$min_{w,b,\xi} \quad \frac{1}{2} w^T w + C \quad \textit{Subject to} \quad y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i \tag{5.33}$$

Where ξ is the slack variable, C is the user-specified penalty parameter of the error term (C>0), and φ is the kernel function. SVM can change the original non-linear separation problem into a linear separation case by mapping input vector on to a higher feature space. On the feature space, the two-class separation problem is reduced to find the optimal hyperplane that linearly separates the two classes transformed in to a quadratic optimization problem. Depend on problem type, several kernel functions are used.

Two best kernel functions for classification problems are Radial Basis Function (RBF) and Gaussian function regard to nonlinearity consideration.

$$K(x_i, x_j) = exp\{-\gamma \|x_i - x_j\|^2\}, \ \gamma > 0 \quad RBF \tag{5.34}$$

In SVM classification, the optimal separating function reduces to a linear combination of kernels on the training data, $f(x) = \sum_j \alpha_j y_j K(x_j, x) + b$,with training vectors $\boldsymbol{x}_i$ and corresponding labels $\boldsymbol{y}_i$ in the dual formulation of the training problem, the coefficients $\boldsymbol{\alpha}_i$ are obtained by minimizing a convex quadratic objective function under constraints.

$$min_{0<\alpha_i<C} \quad W = 1/2 \sum_{ij} \alpha_i Q_{ij} \alpha_j - \sum_i \alpha_i + b \sum_i y_i \alpha_i \tag{5.35}$$

With Lagrange multiplier (and offset) b, and the symmetric positive definite kernel matrix $Q_{ij} = y_i y_j \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ the first-order conditions on **W** <reduce to the Kuhn-Tucker (KT) condition:

$$\frac{\partial W}{\partial \alpha_i}\begin{cases} > 0; & \alpha_i = 0 \\ = 0; & 0 < \alpha_i < C \\ < 0; & \alpha_i = C \end{cases} \qquad \frac{\partial W}{\partial b} = 0 \tag{5.36}$$

The margin vector coefficients change value during each incremental step to keep all elements in equilibrium. i.e., keep their KT conditions satisfied. It is naturally implemented by decremental unlearning, adiabatic reversal of incremental learning, on each of the training data from the full trained solution. Incremental learning and, in particular, decremental unlearning offer a simple and computationally efficient scheme for on-line SVM training. It has also exact leave-one-out evaluation of the generalization performance on the training data.

5.3 Artificial Neural Network (ANN)

5.3.1 Multilayer Perceptron Trained with Back-propagation (MLP)

An artificial neural network (ANN) is an information-processing paradigm inspired by the way the biological nervous system (the brain) process information. It is comprised of massively parallel arrays of nonlinear interconnected processing elements (neurons) working in unison to solve specific problems. ANN, like people, learns by example. An ANN is configured for a specific application like pattern recognition through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between neurons.

Feed-forward neural networks (FFNN) are suitable structure for nonlinear separable input data. In FFNN model the neurons are organized in the form of layers. The neurons in a layer get input from the previous layer and feed their output to the next layer. In this type of networks connections to the neurons in the same or previous layers are not permitted. Figure 5.10 shows the architecture of the system for face classification. The back-propagation training algorithm is an iterative gradient algorithm designed to minimize the mean square error between the actual output of a multilayer fee-forward perceptron and the desired output (Lippmann, 2004). According to Lippmann (2004), the BPNN function according to the following steps

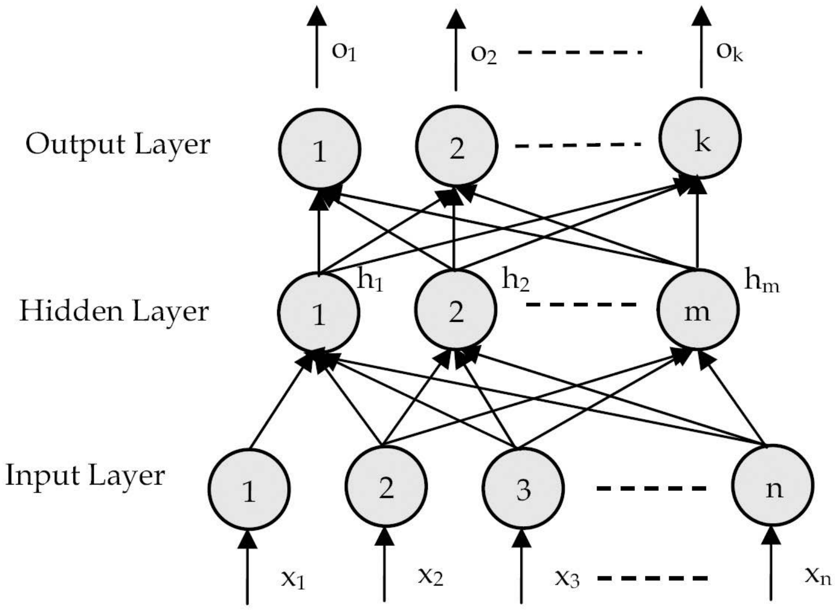


Figure 5.10 the architecture of FFNN for classification space

- Initialize weights and offsets to small random vales.
- Present input and desired output.
- Calculate actual outputs.
- Adapt weights using a recursive algorithm starting at the output nodes and working back to the first hidden layer according to the following equation:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_i x_i^{'} \qquad (5.37)$$

Where $w_{ij}(t)$ is the weight from hidden node i or from an input node j at time t .

η is the training rate coefficient that is restricted to the range [0.01,1.0].

x_j is the output at neuron j in the input layer, and

δ_i is an error term for node j . If node j is an output node then:

$$\delta_i = (t_i - o_i) o_i (1 - o_i) \qquad (5.38)$$

Where t_i is the target output of node i

o_i is the actual output.

Internal node threshold are adapted in a similar manner by assuming they are connection weights on links from auxiliary constant-valued inputs. For hidden nodes the following weight adaptation equation applies.

$$\delta_{Hi} = x_i (1 - x_i) \sum_{j=1}^{k} \delta_j w_{ij} \tag{5.39}$$

and

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_{Hi} x_i^{'} \tag{5.40}$$

x_i is the output at neuron i in the input layer, and summation term represents the weighted sum of all δ_j values corresponding to neurons in output layer that obtained.

Chapter 6

6 Experimental Result

6.1 SVM Parameter Setting

6.1.1 Introduction

During the past decades, the system maintenance strategy has experienced three development stages (Yoshimura, 1999): breakdown maintenance, time-based maintenance and condition-based maintenance. Presently the condition-based intelligent preventive maintenance is gaining more and more interests for HVAC systems. Building Energy Management Systems (BMS) were developed to monitor and control the HVAC systems, whilst a number of Fault Detection and Isolation (FDI) methods and applications were assembled by International Energy Agency (IEA) to detect and prevent the faults further (Hyvarinen, 1996; Dexter, 2001). Generally, the FDI methods can be divided into two types (Liang, 2007; Salsbury, 2001; Zhou, 2009): the feature-based method (Du, 2006), the model-based method (Simani, 2003). Model-based techniques either use mathematical model or a knowledge model to detect and isolate the faulty modes. These techniques include but not limited to observer-based approach (Zhang, 2009), parity-space approach (Kim, 2005), and parameter identification based methods (Dote, 2002). Henao (2003) reviewed fault detection based on signal processing. This procedure involves mathematical or statistical operations which are directly performed on the measurements to extract the features of faults.

Although these methods have been applied in a number of industrial processes with good performance (Chen, 2007), their application in the HVAC system is still at the research stage in laboratories and most of the studies focus on the vapour compression system. For example, Kim (2005) investigated the effect of four artificial faults on the performance of a variable speed vapour compression system by using a rule-based fault classification method. Tassou and Grace (2005) also presented a fault diagnosis and refrigerant leak detection method for vapour compression refrigeration systems by using neural network and expert system. Furthermore, several studies were also presented to deal with the faults in the air handling unit (AHU) (Dexter, 2001; Lee, 2004; Pakanen, 2003; Yoshida, 1999; Yoshida, 2001) and sensors (Hou, 2006; Wang, 2004) respectively. At the same time, some advanced algorithms, such as transient pattern analysis (Cho, 2005), multi-step fuzzy model-based approach (Dexter, 2001), general regression neural networks (Lee, 2004), and feed forward control scheme (Salsbury, 2001), etc. were also utilized to detect and diagnose the faults in the HVAC systems.

Intelligent methods such as Genetic Algorithm (GA), Neural Network (NN) and Fuzzy Logic had been applied during last decade for fault detection. Neural network has been used in the variety range of systems for fault detection even for HVAC systems (Du, 2009). Lo (2007) proposed intelligent technique based on fuzzy-genetic algorithm (FGA) for automatically detecting faults on HVAC systems. However, many intelligent methods such as NN often require big data set for training. Some of them are not fast enough to realize real time fault detection and isolation. This paper investigates methods with real time operation capability and requiring less data. Support Vector Machine (SVM) has been extensively studied in data mining and machine learning communities for the last two. SVM is capable of both classification and regression. It is easy to formulate a fault detection and isolation problem as a classification problem.

SVM can be treated as a special neural network. In fact, a SVM model is equivalent to a two-layer, perceptron neural network. With using a kernel function, SVM is an alternative training method for multi-layer perceptron classifiers in which the weights of the network are identified by solving a quadratic programming problem under linear constraints, rather than by solving a non-convex unconstrained minimization problem as in standard neural network training.

Liang (2007) studied FDI for HVAC systems by using standard SVM (off-line). But there are two weak points on his application based on offline SVM method. First one is the low speed of training system by his proposal SVM as offline method used batch data for training. The second one is refer to his offline algorithm of training which need to offline application. In this chapter, incremental-decrimental SVM based on on-line methodology has been applied. As we discussed in chapter 2, it is required to solve a quadratic programming (QP) for the training of a SVM. However, standard numerical techniques for QP are infeasible for very large datasets which is the situation for fault detection and isolation for HVAC systems. By using online SVM, the large-scale classification problems can be implemented in real time configuration under limited hardware and software resources. Furthermore, this paper also provided a potential approach for the implementation of FDI under unsupervised learning frame work.

Based on the model structure which developed on chapter 4, we constructed a HVAC model by using MATLAB/Simulink and identified the variables which are more sensitive to commonly encountered HVAC faults. Finally, the effectiveness of the proposed online FDI approach has been verified and illustrated by using Simulink Simulation Platform. The chapter is organized as follows. The result shows our proposed algorithm work properly with high speed and good accuracy.

Now it is well recognized that FDI is very important in ensuring the safety of the HVAC systems, improving user comfort, improving energy efficiency, and reducing operating and maintenance costs. However, efficient FDI methods for HVAC systems still remain as a challenge and commercial FDI systems are only beginning to emerge in recent years.

6.1.2 Parameter Setting and SVM Test

It is well known that SVM generalization performance (estimation accuracy) depends on a good setting of meta-parameters parameters C, σ and the kernel parameters. The problem of optimal parameter selection is further complicated by the fact that SVM model complexity (and hence its generalization performance) depends on all three parameters.

Existing software implementations of SVM regression usually treat SVM meta-parameters as user-defined inputs. In this section we focus on the choice of C and σ, rather than on selecting the kernel function. Selecting a particular kernel type and kernel function parameters is usually based on application-domain knowledge and also should reflect distribution of input (x) values of the training data. For example, in this research we used SVM classification using radial basis function (RBF) kernels where the RBF width parameter should reflect the distribution/range of x-values of the training data.

Parameter C determines the trade-off between the model complexity (flatness) and the degree to which deviations larger than σ are tolerated in optimization formulation. For example, if C is too large (infinity), then the objective is to minimize the empirical risk only, without regard to model complexity part in the optimization formulation. Parameter σ controls the width of the σ-insensitive zone, used to fit the training data. The value of σ can affect the number of support vectors used to construct the regression function. The bigger σ, the fewer support vectors are selected. Hence, both C and σ-values affect model complexity (but in a different way). Existing practical approaches to the choice of C and σ can be summarized as follows:

Parameters C and σ are selected by users based on a priori knowledge and/or user expertise. Obviously, this approach is not appropriate for non-expert users. Based on observation that support vectors lie outside the ε-tube and the SVM model complexity strongly depends on the number of support vectors, Schölkopf et al (2002) suggest to control another parameter ν (i.e., the fraction of points outside the σ -tube) instead of σ. Under this approach, parameter ν has to be user-defined.

Similarly, Tax (2001) propose to choose σ- value so that the percentage of support vectors in the SVM regression model is around 50% of the number of samples. However, one can easily show examples when optimal generalization performance is achieved with the number of support vectors larger or smaller than 50%. Chew (2001) and Smola (2000) proposed asymptotically optimal σ- values proportional to noise variance, in agreement with general sources on SVM. The main practical drawback of such proposals is that they do not reflect sample size. Intuitively, the value of σ should be smaller for larger sample size than for a small sample size (with the same level of noise).

Selecting parameter C equal to the range of output values is another technique for SVM parameter setting. This is a reasonable proposal, but it does not take into account possible effect of outliers in the training data. In some cases using cross-validation for parameter choice is best choice. This is very computation and data-intensive. Several recent references present statistical account of SVM regression where the σ- parameter is associated with the choice of the loss function (and hence could be optimally tuned to particular noise density) whereas the C parameter is interpreted as a traditional regularization parameter in formulation that can be estimated for example by cross-validation.

As evident from the above, there is no shortage of (conflicting) opinions on optimal setting of SVM regression parameters. Under our approach we propose:

- Analytical selection of C parameter directly from the training data;
- Analytical selection of σ- parameter based on (known or estimated) level of noise in the training data.

In this section we suggest the σ-insensitive loss, in the sense that SVM classification (with proposed parameter selection) consistently achieves superior prediction performance vs other (robust) loss functions, for different noise densities.

As we discussed above, the parameters of online SVM classifier should be trained first. First parameter is maximum penalty. By define the maximum penalty of 10 we could achieved best margin for SVM. Based on the testing of different kernel functions, RBF function chooses as the best kernel function with $\sigma = 1.25$ .The process of finding classification functions using nonlinear SVMs consists of two steps. First, the input vectors are transformed into high-dimensional feature vectors where the training data can be linearly separated. Then, SVMs are used to find the hyperplane of maximal margin in the new feature space. The separating hyperplane becomes a linear function in the transformed feature space but a nonlinear function in the original input space. Note that, the kernel function is a kind of similarity function between two vectors where the function output is maximized when the two vectors become equivalent.

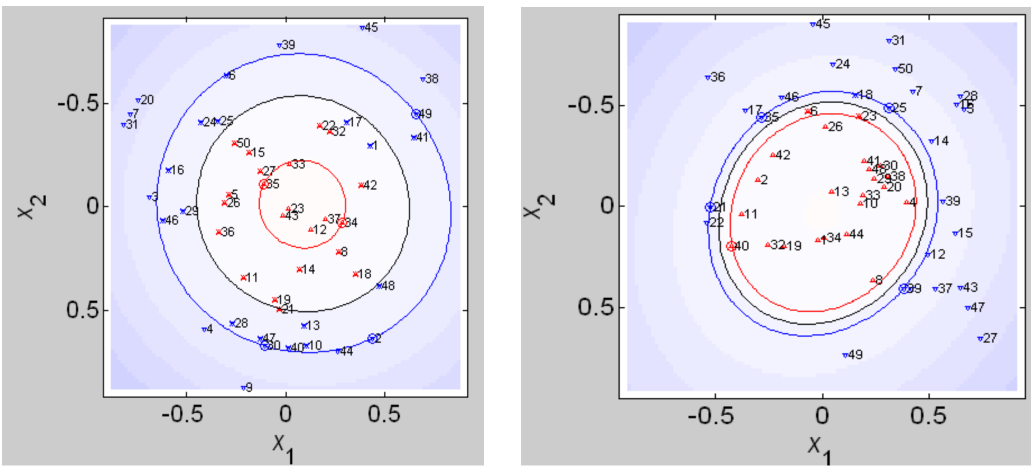


Figure 6.1 SVM classification result on a series of random numbers based on circle condition on two dimensions

Because of this, SVM can learn a function from any shapes of data beyond vectors (such as trees or graphs) as long as we can compute a similarity function between any pairs of data objects. Two regular kernel functions for classification problems are Radial Basis Function (RBF) and Gaussian function regard to nonlinearity consideration.

Figure 6.1 shows the result of our coding for online SVM based on incremental algorithm that was explained. Purpose is separating inside and outside of a circle for testing the code and algorithm. 50 random number between (1) and (-1) was generating as X1 and X2 then terms of y was labelled based on circle equation(X12+X22=0.25). y is labelled healthy (with sign of 1) for all X1 and X2 which working with X12+X22<0.25 condition and labelled faulty (with sign of -1) for X12+X22>0.25 . We train the algorithm with this 50 point and then test it on whole page including 10000 points. The points are indicated on this figure are those which are useful for training and specified as vectors. Black circle is present separator line and distance between red and blue circle is margin. Margin can be change by defining of the penalty factor (C) in the program. Coding and simulation is done in MATLAB and SIMULINK environment (version 2010b) software.

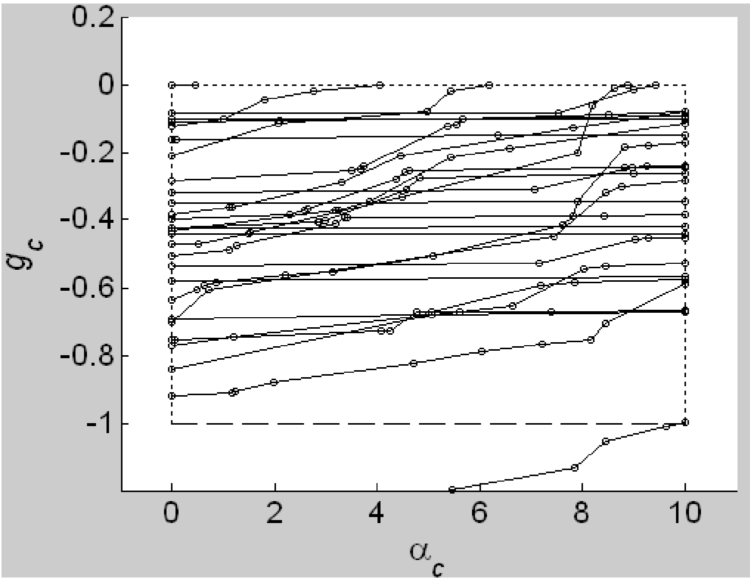


Figure 6.2 margin vector coefficients change against weight during each incremental step

According to figure 6.2 the margin vector coefficients change value during each incremental step to keep all elements in equilibrium. i.e., keep their KT conditions satisfied. It is naturally implemented by decremental unlearning, adiabatic reversal of incremental learning, on each of the training data from the full trained solution.

Incremental learning and, in particular, decremental unlearning offer a simple and computationally efficient scheme for on-line SVM training.

The margin vector coefficients change value during each incremental step to keep all elements in equilibrium. i.e., keep their KT conditions satisfied. It has also exact leave-one-out evaluation of the generalization performance on the training data.

6.2 Test and Algorithm Validation

6.2.1 Algorithm Proposal

There are huge amounts of data in any large scale system such as big buildings with several AHU, heating and cooling system. Most of these data are stored in a database of Building Management System (BMS) and are good potential to analysing for future fault detection.

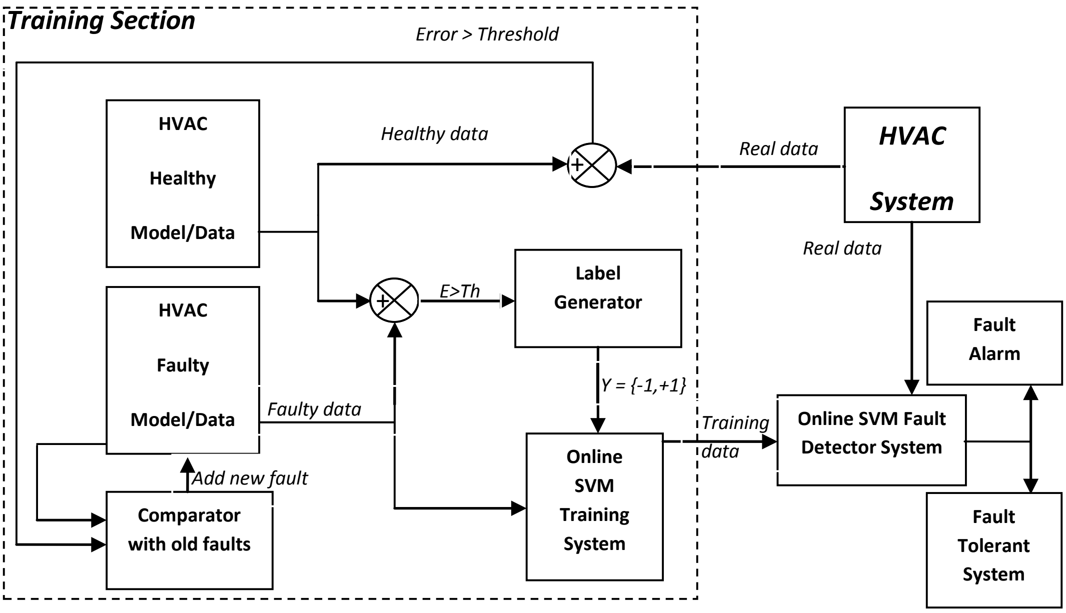


Figure 6.3 Schematic of semi unsupervised fault detection with online SVM for an HVAC system

For large datasets, standard SVM techniques (off line SVM) become infeasible. This motivates the usage of incremental-decremental SVM (online SVM) for such large database with huge amount of data. This also inspires us to design a fast and accurate algorithm for fault detection over the large scale system with inherent time delay.

Figure 6.3 shows the proposed fault detection scheme by using incremental-decremental support vector machine classification. The main purpose of the system is to detect unknown faults by monitoring key HVAC variables during system operation. In this algorithm new faults can be detect (as unknown new faults) by comparing with the outputs of the healthy model and the real system. If detected fault was similar to old fault, it will be categorized by algorithm as existing faults. Otherwise, this data is sent to online SVM trainer for training for the new fault. Finally new fault will be isolate by this online SVM as a known fault. The incremental procedure is reversible and decremental unlearning of each training sample produces an exact leave- one-out estimate of faults with using all HVAC data during its operating. The main advantage of this algorithm is usage of only a range of useful data (including healthy data, old faults, and new faults) instead of whole data sets. Based on this online training procedure, semi-unsupervised fault detection can be implemented.

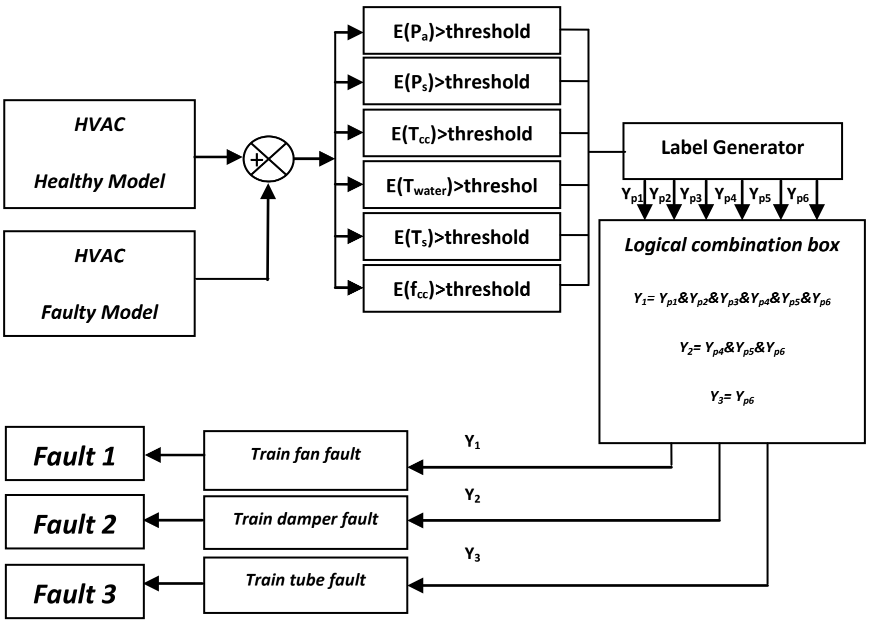


Figure 6.4 Schematic of label generation algorithm for training system of each fault

Figure 6.4 shows the structure of the label generation algorithm. This model is evaluated based on outputs of experimental data of the UTS HVAC system. The label of y is set as +1 (Fault free) when the error is smaller than a given threshold and it is set as -1 (faulty) when the error is bigger than that threshold. Labels including $y_{p1},\ldots,y_{pn}$ are generated with *n* variables for each fault. For each fault all labels should be combined together in a proper logic to generate one label as one fault needs just one label for training. A combination and configuration of different labels can be used for generate final label. There are other ways to configure between different labels and it depends on complexity of variable. For example if data are chosen from a BMS system with multi impacts and effects then using fuzzy logic algorithm can be the best option. The membership function can be used for any single label and the rules help us to generated final label based on this membership functions. In this research, a specific fault can be happen if all errors of sensitive parameters can passed from their thresholds. However, a simple AND, OR logic applied for generating final label in this section due to simple experimental structure of HVAC system with one AHU, cooling and heating systems.

6.2.2 Test of Developed SVM with Mathematical Model

6.2.2.1 Artificial Fault Generation

HVAC system may suffer from many faults or malfunctions during operation. Three commonly encountered faults are defined in this simulation:

- Supply fan fault
- Return damper fault
- Cooling coil pipes fouling

Usage of these faults is just to test the performance of proposed fault detector system. Incipient faults applied to test the proposed fault detector due to its difficulty to detection.

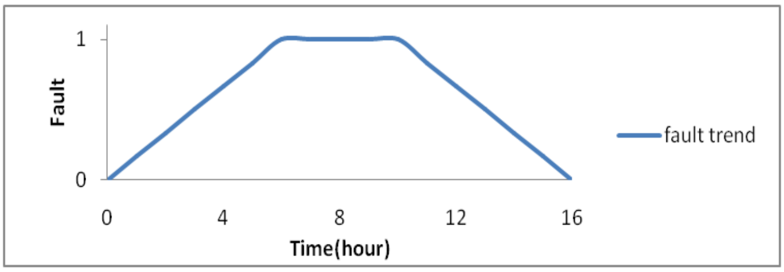


Figure 6.5 Fault trend during one day

Four different models consist of one healthy model and three faulty models are generated. Figure 6.5 shows general profile of the fault during one day. The amplitude of the faults gradually increased for 6 hours to reach their maximum values then they stay in this state for 4 hours and returns gradually to normal condition during the last 6 hours. This fault profile has been applied for each fault with some minor changes. In air supply fan fault this profile is used with gain of 10.

In damper fault it is used with gain of -2 and shift point of +4. For pipe fault it is used with gain -0.3 and shift point of +1. The most sensitive parameters have been identified for each fault. For the air supply fan fault the major most sensitive parameter is the air supply pressure that changes between 0 to 10 Pascal during fault period. The mixed air ratio is selected as the indicator of the damper fault that decreases from 4 to 2. Cooling water flow rate decrease from f_{cc} to $0.7f_{cc}$ in cooling coil tube fault.

6.2.2.2 Parameter Sensitivity

The sensitivity of variables respect each fault is analysed in this subsection. Figure 6.6 shows sensitivity of cooling water flow rate respect to different faulty modes. It is clear cold water flow rate has been sensitive to all three faults (damper fault, supply fan fault and cold water tube fault).

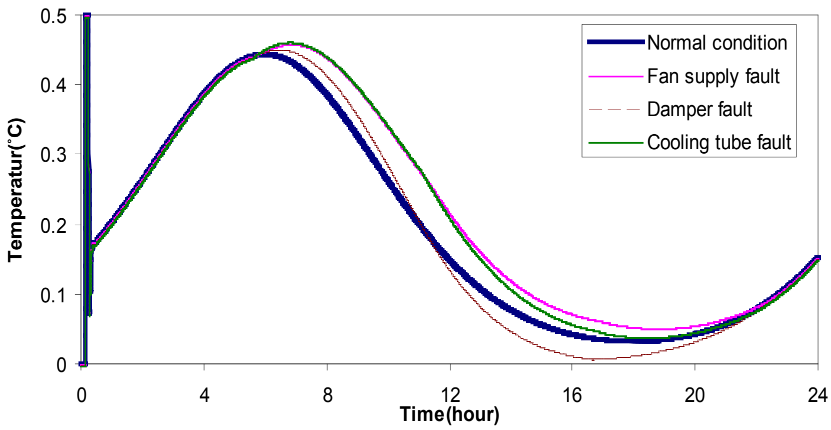


Fig. 6.6 Cooling water flow rate sensetivity agains three artificial faults

This sensitivity was different for any of these artificial faults with sensitivity of 0.03 kg/s for damper fault, 0.07 kg/s for supply fan fault and 0.08 kg/s for tube fouling fault.

As we can see the damper fault has minimum influence on the cold water flow rate against tube fouling has maximum effect. So base on this sample analysis we can consider more cold water flow rate weight in contribution of two last faults. This could be helpful when there are many sensitive variables to specify fault. In this case we can give different weights to different variables base on a fuzzy logic algorithm to make different contribution.

Other analysis shows cooling coil temperature and outlet water temperature are sensitive to both supply fan fault and return damper fault. Based on this sensitivity analysis, six parameters consisting of air supply, room pressure, air supply temperature, cooling coil temperature, outlet water temperature, and water flow rate are used for training of supply fan fault. Also three parameters consist of cooling coil temperature, outlet water temperature and water flow rate are applied for training of return damper fault but just water flow rate used for training of third fault (Cooling coil pipes fouling).

6.2.2.3 Analysis of Result

Since the SVM classifier presented in last section can only be used to deal with two-class case, a multi-layer SVM framework has to be designed for the FDI problem with various faulty conditions. In order to use online SVM classification method to achieve a better isolation performance, three faulty models are used in the isolation section. A four-layer SVM classifier is designed, in which the normal and three different HVAC fault conditions are all taken into consideration. Furthermore, it should be pointed out that other unknown faulty conditions can be placed in the upper layer of the FDI system. The kernel function must be properly selected for SVM classifier in order to achieve high classification accuracy. In general, linear function, polynomial function, radial basis function (RBF), sigmoid function, and Gaussian function can be adopted as the kernel function. In this section, Gaussian function is used as it has excellent performance in the simulation.

In this research, two tests are conducted systematically. The diagnosis results and corresponding characteristics of the SVM classifiers are shown in Figures 6.7, 6.8 and 6.9. Test 1 is designed to investigate the SVM classifier performance on known incipient faults. The steady-state data is used to build the four-layer SVM classifier: as mentioned in previous section, the data within the threshold under the normal condition indicate fault free, and the data beyond the threshold indicate faults 1 to 3. For each normal/faulty condition, two days data (20 hours data per day between 2am and 22pm) are used. Therefore, a total of 4 times 40 hour samples are collected. Half of the data for each condition are used as the training data, whilst the rest are used as the testing data for fault diagnosis. In Figure 6.7, the label changes from +1 (non-faulty situation) to -1 (faulty situation) when the fault is detected. For simplicity Figure 6.5 only shows two variables when there is fault 1 in the data: the cooling coil temperature and the water outlet temperature. It is clear that the HVAC faults can be diagnosed 100% by using the SVM classifier for the testing data.

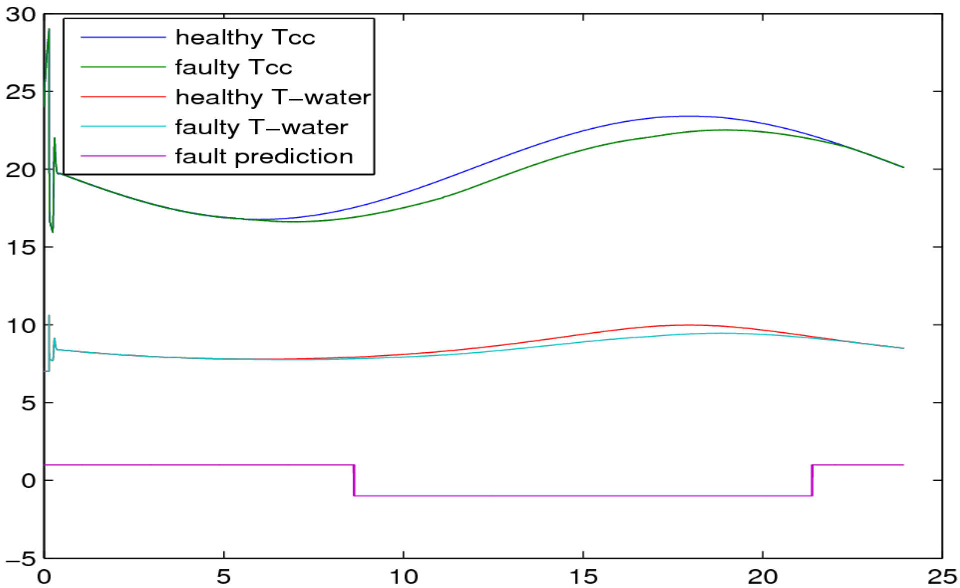


Figure 6.7 Incipient supervised fault detection based on online SVM algorithm

As mentioned earlier, our proposed algorithm is able to detect unknown faults in the sense of semi-unsupervised manner. To testing semi-unsupervised performances, an unknown sudden fault (at time of 9 h) combined with previously introduced incipient faults are imposed on system at second test. The detection results are shown in Figure 4.16. It is clearly indicated that the margin changes from high level to low level when detecting incipient fault. For unknown fault this change is dramatic as unknown fault is abrupt type. To efficiently optimizing training process, samples in each normal/faulty condition should be applied. A group containing of maximum faulty training samples is selected, and applied for training.

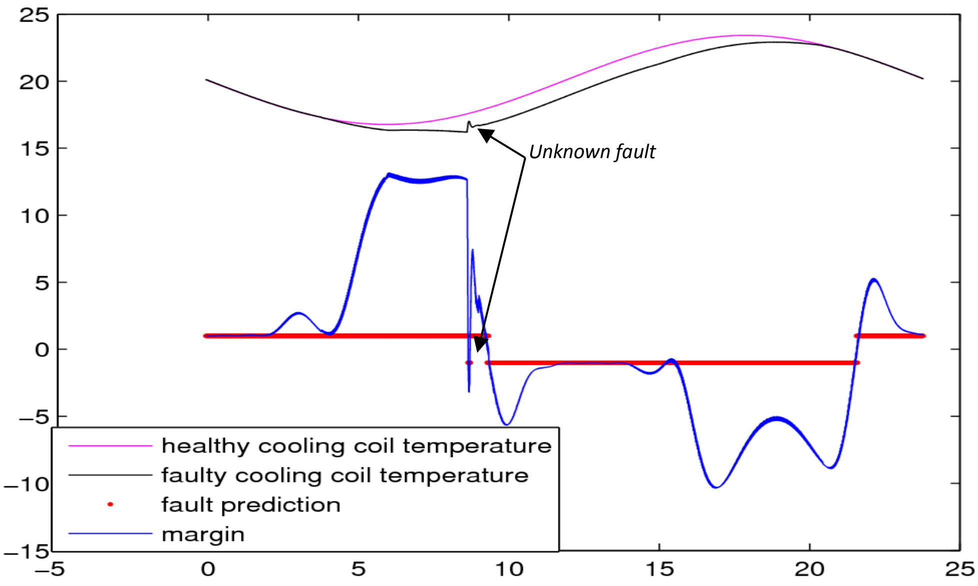


Fig.6.8 Sudden unsupervised fault detection based on online SVM

From the Figure 6.8, it is found that the designed SVM classifier can identify the HVAC unknown fault accurately. Based on the simulation result, it is found that by using the proposed approach, the unknown faults of HVAC system can also be detected efficiently. As we can see an unexpected fault happens between times of 5 to 10 hours between a known long duration faults. This fault could be detected as soon as sensed by the algorithm even it was working on a supervised detection algorithm. A fluctuation of margin also indicated an unexpected fault during supervised working. This is because online SVM in our algorithm used incremental learning and any rapid change in one variable compare with other training variables. If this change is compatible with other variables change then this could be account as a potential of supervised faults. If this is a change in just one variable with incompatibility or less compatibility of other variables then this could be account as set point change or new potential faults.

Cooling coil temperature has been shown in figure 6.8 for avoid any complexity of figure to understanding of the fault situation. However, there are other sensitive variables for specific fault which are used for training purposes. These variables are limited to those variables which are used for sensitivity analysis in last chapter. As we know mathematical model was used in this section to confirm the performance of our fault detector algorithm based on incremental online SVM. The result shows this confirmation in two last figures.

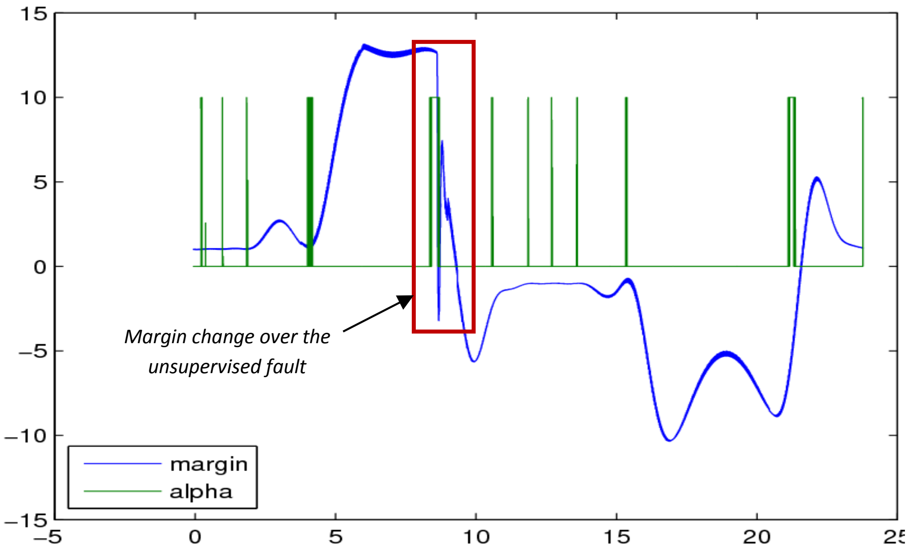


Fig.6.9 Training coefficient via margin change for unsupervised algorithm

We can also follow other SVM parameters for both supervised and unsupervised fault detection. These could be including margin in one side and other parameters at other side. One of these parameters is alpha ($\boldsymbol{\alpha}_i$) coefficient which is change during training and detection. As figure 6.9 illustrates alpha has been specific change during unexpected fault respect to margin change. This change is wider in time scale compare to other changes during both training and testing section. As mentioned in previous section, this coefficient should be confine between zero and maximum penalty as shown in this figure.

This section focuses on the fault detection and isolation of HVAC system under mathematical model working conditions. An online SVM FDI classifier has been developed which can be trained during the operating of the HVAC system. Different with off line method, the proposed approach can even detect new unknown faults for the training of the classifier in real time. Furthermore, this online approach can more efficiently train the FDI modular by throwing out unnecessary data (leave out vectors) and just used a series of data with high priority regarding to classification. Due to these properties, the proposed algorithm can be implemented in a semi-unsupervised learning frame work. Simulation study indicates that the proposed approach can efficiently detect and isolate typical HVAC faults. In the next step, we will validate the proposed approach by using real experimental data.

6.2.3 Test Developed SVM with Laboratory Experimental Data

6.2.3.1 HVAC Laboratory of UTS

Object is a HVAC laboratory scale system which is used for application and commissioning of new technology on the HVAC.

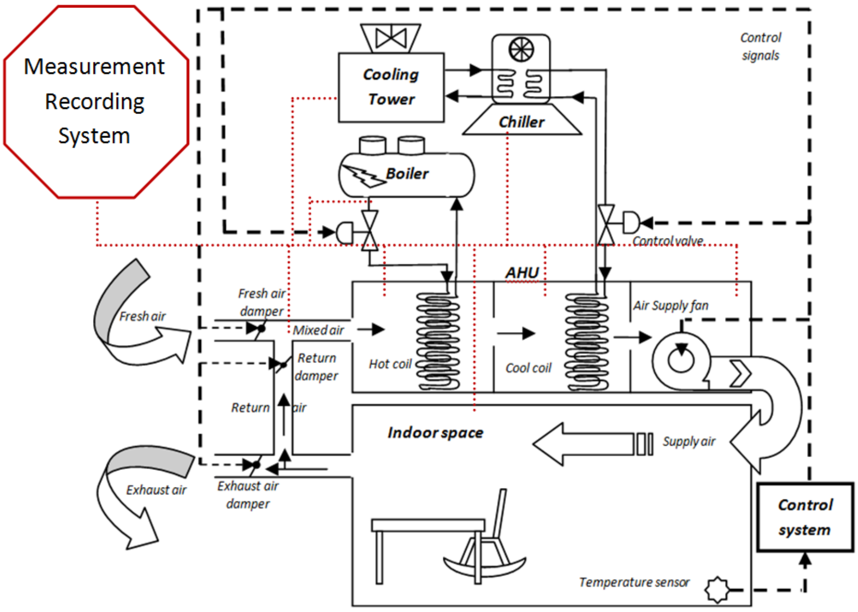


Figure 6.10 schematic diagram of UTS HVAC system

This system is located in air conditioning laboratory at University of technology Sydney (UTS) and designed for study and research purposes. A schematic diagram of this system is also presented in figure 6.10 based on subsystems and single controllable room. The HVAC system consist of five main sections including AHU, room space, heating system, cooling system and control system. Some parameters including temperature, pressure and humidity are measured from different sections and record in a computer centre to analysis data for our purpose. Different ranges of measurement and control actuators have been configured for this laboratory. Temperature and humidity measured by temperature (PT1000) and humidity sensor and convert to proper signal by their own transmitters to 4-20 mA and 0-5 V. These transmitters then connected to the laboratory reader card noise free wirings. Both This card is then configured by RS-232 protocol to the PC with WIDOWS operator system. A specific software record data in excel or text file format or connect to MATLAB software for direct use in workspace or Simulink environments.

Figure 6.11 Measurement systems for HVAC laboratory of UTS

Two monometers are installed to measured local pressure based on gauge configuration. This measurement will be done manually as the monometers are for local reading and there is not any transmitter for reading in computers. Both measurement system for transmitter type and monometers are shown in figure 6.11 for this laboratory.

For control of the room temperature there are two options in this system. A pneumatic actuator for a control valve is used to control of cold or hot water flow rate. Also there is an air supply fan speed controller which is controlled by an inverter based on frequency.

Figure 6.12 actuators system for UTS HVAC room temperature control

Both actuators are shown in figure 6.12 as main part for control the room temperature. Water flow rate can be change by a feedback signal based on difference between room temperature an adjusted set point. This is done by a PID controller which is installed on the AHU system. A simulated room space has been configured for this HVAC system as we can see in figure 6.13. There is a heating system inside the room which is generating heat and moisture based on number of inside people. This can simulate a building with certain number of people then it is possible to see influence of people on model and simulation. The gross floor area of the room is 32 square meters and building height is 2.7 meters. The ground area to be occupied by the building has a rectangular shape. Also walls, windows, floor and roof are modelled according to ASHRAE (2000) transfer function approach. This system was run for working hours of building during specific days between 8 a.m. to 10 p.m. Lighting power density is 2 W/m^2 a common value used in practice for a commercial building. The heat gain from the lights is assumed to be 40% convective.

Figure 6.13 room space of UTS laboratory HVAC system

Figure 6.14 shows the cooling system in this laboratory including a chiller and cooling tower system. The central cooling plant installed in the building consists of one water cooled chiller, one cooling tower, one air handling unit, two chilled water pumps and two condenser water pumps. The chiller has screw compressors, with a nominal capacity of 7 kW, and uses refrigerant R-407C. For the plate-type evaporator, the evaporator temperature is set to be 4°C. The temperature of the supply chilled water is taken at 7°C at design conditions. The chiller comprises two refrigeration circuits in parallel in which each circuit includes one thermostatic expansion valve and one screw compressor. The chiller can operate down to about 10% of its rated full load capacity via a modulating slide valve in the compressor.

Figure 6.14 Chiller and cold water system of HVAC lab.

The design air flow rate and the electric power input of variable speed cooling tower fan at maximum air flow rate are 1700 m^3/h and 0.35 kW respectively. The design air flow rate of the air handling unit with variable air volume fan is 2500 m^3/h and its rated power input is 1.2kW. The design water flow rate and electric power of the chilled water pump is 2.1 m^3/h and 0.32 kW respectively. The design water flow of each condenser water pump is 2.5 m^3/h and their electric power is 0.15 kW. All circulator pumps operate at a constant speed. A photograph of AHU section of UTS HVAC system which is used for the case study is shown in Figure 6.15.

Figure 6.15 Air Handling Unit (AHU) of UTS laboratory

Room temperature(T_r), supply temperature(T_s), cooling coil temperature (T_c), room and supply moistures(r_r,r_s), chiller inlet and outlet water temperatures(T_{chi},T_{cho}) and chiller power consumption(P_{ch}) take as HVAC system output and chiller cold water mass flow rate (m_{ch}) takes as HVAC system input for black box model. All mentioned parameters are regular parameters which are used in a HVAC control system. So customers are not force to take extra cost for new sensor installation. Test is run three times in similar outdoor condition for 14 hours from 8:00 am to 10:00 pm. first day of running data used for black box modelling of the HVAC system. The second day data was including artificial faults of supply fan which are spread during a full day at the times 10 am, 1 pm and 4 pm. third test is instruct based on return damper fault with same condition of second test. All tests are conducted during summer season with highest outdoor temperature between 34 to 35℃ at afternoon time.

In order to determine the optimal control variables of the cooling system, a number of tests under different running conditions were conducted. High precision sensors/transducers were used for measuring all operating variables. Manometers were used for obtaining supply air flow rate. The temperature sensor for supply air is of platinum resistance type with accuracy: ± 0.1°C. Water temperatures were measured by mercury thermometers of precision: ± 0.2°C. The ambient temperatures were measured by a digital thermometer of precision: ± 0.8°C. Powers of components were measured by a digital ac/dc power clamp multimeter precision: ± 3.5%. Therefore a total of fifty-six points of system power consumption and other variables were measured for each fifteen-minute period. The building and sensible and latent cooling loads were calculated from monitoring data. Indoor sensible loads were determined by assuming that they are exactly the same as the product of monitored supply fan air flow rate and the difference in the monitored temperature between the supply and air zone. The building latent loads were calculated using the product of the fan air flow and the difference in supply and return humidity ratios. Both humidity ratios and temperature are determined through the monitored air temperature and relative humidity. Then the controlled and uncontrolled variables were stored and arranged in database files so that process can be performed using the proposed approach.

6.2.3.2 Experimental Black Box Model Evaluation

6.2.3.2.1 Short History to HVAC Modelling

In parallel with modelling of other energy systems, HVAC modelling was developed by Arguello-Serran[84], Tashtoush[85] during two decades but these models are generated for a special work. Our new generated model is a general dynamic model without any extra assumption which controlled by a control system same as real HVAC system. Dynamic models of HVAC system tend to be computationally demanding and for practical utility, particularly in the case of control development and fault detection, must be amenable to numerical solution within an acceptable time frame. A transient model that takes several hours or perhaps days to solve on a computer is of limited practical value and the efficiency of the numerical solution procedure is clearly of paramount practical importance. Models of HVAC system also contain a large number of parameters and the ability to either estimate or measure the model parameters is another key issue for practical applications. A detailed mathematical model, whose parameters are not open to estimation, or easy measurement, is clearly of limited practical use. Our primary objective is to develop nonlinear transient mathematical model of a HVAC system that can be solved efficiently and make a possibility for reviewing the affection of different kind of faults on different variables. A close loop simulation of this model toward its control system for control of different parameters is another new application on simulation in which investigated in this research.

Fault detection and diagnosis (FDD) are important in process engineering and have attracted a lot of attention recently. The main benefits of FDD in HVAC applications derive from reduced operating costs and/or improved indoor environment. Detailed literature reviews in this field can be found in the papers of Comstock et al. [86], Reddy et al. [87] as well as Dexter and Pakanen [88]. Chiller is an important component of HVAC equipment that consumes a significant portion of energy. Chiller systems account for a large portion of the energy consumption of HVAC system of buildings. It is estimated that energy consumption of chiller plants typically contribute to 35-40% of the total building electricity consumption in commercial buildings in Hong Kong. Further- more, chiller performance degrades naturally and different kinds of faults (component faults and sensor faults) may occur in the course of operation, which might result in a great waste of energy. As for sensor faults in chiller systems, a series of comprehensive investigations can be found in the works of Wang and Wang [89, 90], and Wang and Xiao [91]. Component hard faults which cause the system to stop functioning (e.g., seized compressors, broken fan belts and malfunctioning electrical components) are usually easier to detect, since they occur abruptly and result in a sudden failure of some part of the plant [92]. Effects of cooling tower fault on the chiller are the prime concern in this section.

Several techniques have been proposed for nonlinear system identification. Most of them are based on parameterized nonlinear models such as Wiener–Hammerstein models, Volterra series, wavelet networks, neural networks, etc. The parameter estimation can be performed using non-adaptive techniques such as least squares methods and higher order statistics-based methods and adaptive techniques such as the back-propagation algorithm and adaptive gradient learning [93], [94], [96], [97], [100], [101], [103], [104], [106], [108]. Neural networks (NN's) [95], [97], [105] have been also applied to modelling and identification of nonlinear systems such as memory-less channels [98], [101] (e.g., traveling wave tube (TWT) amplifiers [96], [106]), nonlinear Wiener systems [93], [108], and satellite communication channels [98], [100]. NN models have shown good performance compared with classical techniques [99], [108]. If these NN approaches are to be used in real systems, it is important that the algorithm designer understands their learning behaviour and performance capabilities. Several authors have studied the convergence properties of neural network algorithms (see an extended bibliography in [99]). For example, Shynk and Roy [107] studied a simple, but very interesting structure consisting of two inputs and one neuron. The authors studied, in particular, the influence of the learning and momentum rates on the algorithm behaviour near convergence. In [102], the influence of the number of layers on the performance and convergence behaviour of the back-propagation algorithm has been studied. A statistical analysis of the two-layer back-propagation algorithm (with an arbitrary number of neurons) has been presented in [98]. This study has been generalized to use a feed-forward back-propagation algorithm for modelling of chiller system.

6.2.3.2.2 ANN Black Box Model (Chiller Approach)

In this section chiller model is constructed by an Artificial Neural Network (ANN) based on feed-forward-back-propagation (FFBP) training algorithm. The reason for generation of this model is make infrastructure to increase robustness for fault detector algorithm (online SVM classifier). It is not only for robustness issues but also it can prepare a fault free reference to make possibility for unsupervised fault detection.

There are different type of faults in any section of HVAC system which can be categorize as detectable or undetectable nature. But some faults are more important due to energy waste problems and safety problem. One of main critical fault is cooling tower fan fault which neglecting in many cases lead to significant problem through the air quality and wasting of energy. If detection of these fault types accrue with late time then it can lead to many energy wastes in primary (chiller) and secondary (AHU) systems.

Another problem is lack of measurement devices in this area. There are not many instruments in a cooling tower to check the condition of different devices in this part of HVAC system. But some unusual signage can be delivered to lower level such as chiller or AHU. In this study we show it is possible to check cooling tower faults by analysing downstream data from chillers. This could be possible due to powerful detection algorithm which applied for fault detector.

On the other hand many algorithm have been developed and applied but those were tested with different hard situation and minimize data becomes popular both in industry and laboratories. Hence, in this research our focus is to test our proposed algorithm with tough condition of experimental data. So term of tough condition is refer to minimum data and indirect data from secondary system. This situation is available for cooling tower fault detection by indirect data from secondary system like chiller. That makes interest point for cooling tower fault detection by the chiller feature sensitivity as tough test condition in this research.

Different black box algorithms can be develop for a series of raw data in any real case study. But ANN is more reliable and cost effective methodology due to its capability to deliver maximum dynamic of the system. In first stage of this stage a chiller black box model has been developed from experimental data to setup fault free reference. For this reason our laboratory HVAC system with all subsystems including AHU, cooling tower and chiller run for one business day which is conducted for 12 hours (HVAC was run from 8:00 am to 8:00 pm, also other experimental data have been presented in appendix B). Then all measured variable record in a table to use those related variables for chiller model. The model evaluate for a day working hours based on a summer sunny day. The result of this experimental data in other cases brought in the appendix B.

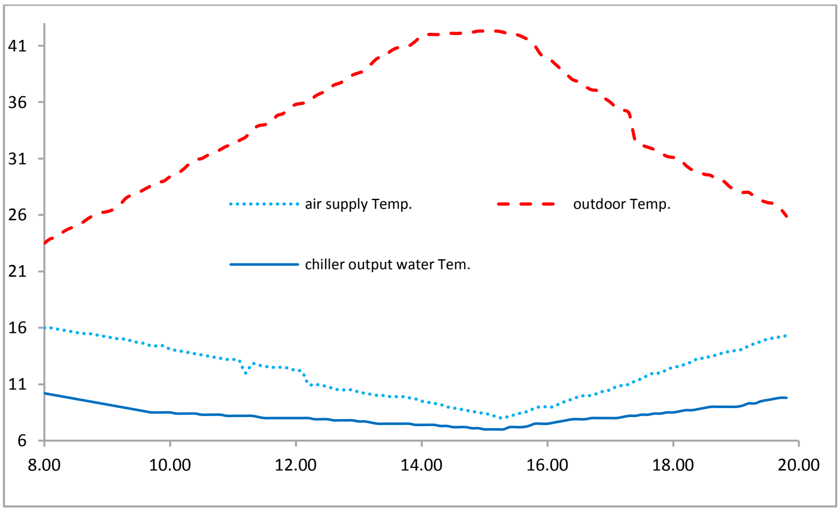


Figure 6.16 experimental data for Ta,Tch,Ts from 8:00am to 8:00pm

A temperature pattern shows in figure 6.16 to compare difference between outdoor temperature and two temperature in primary (chiller) and secondary (AHU) sections. The chiller output temperature (T_{ch_out}), air supply temperature (T_s) and outdoor dry bulb temperature (T_a) for normal condition (faulty free condition) are presented in this figure between 8:00 am to 8:00 pm. As we can see T_{ch_out} and T_s change with the outlet temperature by an inverse profile due to their effort for adjusting the indoor temperature with the set point.

There are various definitions of models on a certain system but each model defines just for special purpose. The accuracy of model is an important key factor that determines the model validation. According to our real case study, four parameters measured for modelling of the chiller. Water inlet temperature (T_{ch_in}) and chiller power consumption (P_{ch}) are used as input variables in contrast of water outlet temperature (T_{ch_out}) and air supply fan temperature (T_s) as two output variables used in our basic model. The model is generated according to neural network algorithm which gives us a high accuracy based on Feed-Forward Back-Propagation (FFBP) application. The network used two layers and some parameters that are tuned for minimum error of network model compare to real case.

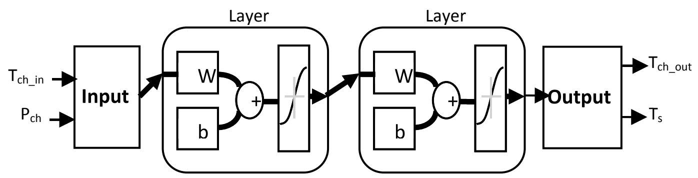


Figure 6.17 Two layer NN block diagram with FF structure

The run is conducted by one busy hours of a day from 8am to 8 pm. Figure 6.17 shows a block diagram of chiller model with Multi Input-Multi Output (MIMO) in which there are two inputs against two outputs. As we can see there are two middle layers and on input and one output layer for this model. The training model will be done once a day at any night time which HVAC system is on sleep mode. It should be noted that model accepted if there was no fault during following day based on this algorithm otherwise last fault free model has been revised with new one. This procedure makes sure the algorithm have access to fault free model at any time without any restriction.

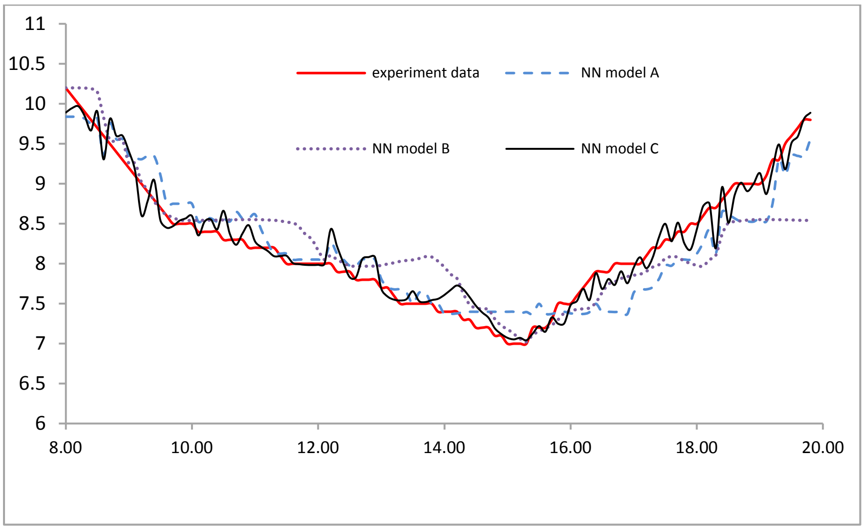


Figure 6.18 NN models valdation with experimental data for T_{ch_out}

There are different input-output configurations for train the proposed neural network based on experimental data. Both outputs (T_{ch_out}, T_s) can be train with a single input which makes two possible configuration of training. Also both inputs can be configuring with both outputs which makes another opportunity of training. Totally, there are three data set configurations for training which can lead to different results. In this research a Neural network model with label of (A) is represented for training model of both outputs including $T_{ch\_out,}$ T_s via chiller inlet water temperature (T_{ch_in}). (B) Label is based on both outputs and power consumption (P_{ch}) input training. Also (C) label is according with two outputs via both inputs (T_{ch_in} , P_{ch}).

Different results of neural network model configuration are compared both on figure 6.18 and 6.19 for any output respectively. Both figures are including real experimental data which presented with red solid graph. Type (A) configuration is based on one input and two outputs which showed with dash blue line in both figures. Dot violet line is representative of (B) type configuration based on another input and both outputs. The last graph with black solid line belongs to C configuration founded on both inputs and both outputs.

Figure 6.18 present a comparison for three NN model and real experimental data for the chiller outlet water temperature (T_{ch_out}). As we can see type (A) configuration has been followed experimental data with high accuracy in first half of day but it does not make good estimation after hours 14:00. (B) type configuration model has minimum accuracy between all three configurations in this figure. But it seems (C) type configuration with two inputs and two outputs have been best approaches for experimental data. It is clear that black graph tried to follow the red graph in many points with maximum fluctuation of 0.4℃.

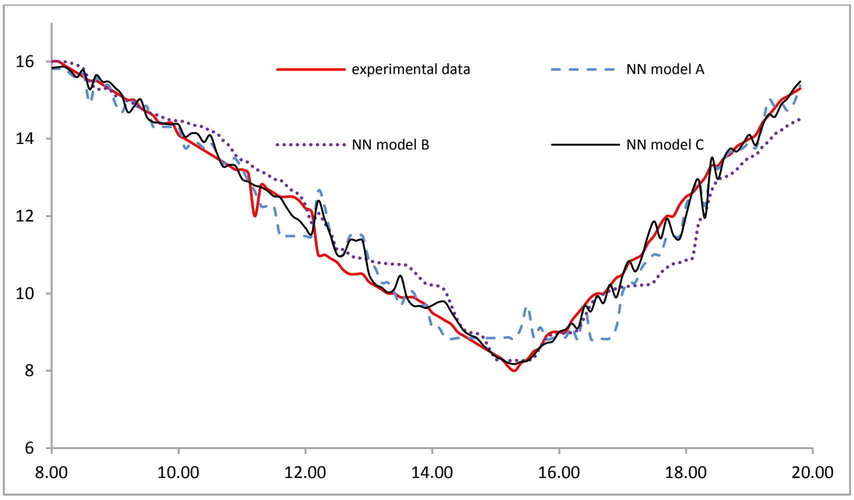


Figure 6.19 NN models valdation with experimental data for T_s

Figure 6.19 shows the difference between all three configurations and real experimental data for air supply temperature (T_s) as second output. Black graph neural network model based on both inputs and both outputs is the best follower as we expected. Both (A) and (B) configurations have been more deviation from experimental data compare to (C) configuration. Although (A) model (blue line) follows the red line better than (B) type model but it does not make an accurate model as well as black line configuration.

In (C) type model outputs follow the experiment data with highly accuracy even there are some fluctuations as the both figures represented. This fluctuation (between 0 to 0.4 °C) can be neglected compare to experimental measurement noises which are usually around 0.7 °C. This analysis was done to find the best configuration of chiller model based on minimum inputs-outputs. However, best choice for this case study model is as 2input-2output NN model (model C) with FFBP method of training based on results in figures 6.18 and 6.19.

6.2.3.3 Cooling Tower Fault Analysis

6.2.3.3.1 Generation, Application and Effect Review of Artificial Fault

Many energy systems are subject to faults or malfunctions and this is more important for public system applications like HVAC. Cooling tower is only part of cooling system which normally forgotten among different subsections of HVAC system. The failure of important components in the cooling tower, such as motor, fan, actuators or sensors can result in unsatisfactory performance, decreased availability, emergency shutdown or even significant damage to the system, humans and the environment. For these reasons, we are interest to check the model sensitivity to some special faults. HVAC system may suffer from many faults or malfunctions during operation but one commonly encountered fault is defined as cooling tower fault that shows its direct effect on different parts of an HVAC system. As we analysis, the direct effect of any fault for cooling tower is on the chiller and air supply temperature. Selecting of this fault was based on condition of application and avoiding damage to the system because it was comfort to turn off the cooling tower fan without any significant malfunction on the HVAC system. This results have been taken as the basic fault for test and review of detector algorithm through the system. So we can test performance and capability of proposed fault detector system just by application of this artificial faults and check whether it can detect any further fault in term of time and magnitude.

The cooling tower fan stopped five times during a day and stop duration is about 10-15 minutes in each round. This stop can be substitute by any fan fault due to motor stop or out of service fan for an artificial fault on the system. 40% of data including two stop have been used for training part and 60% used for test duty. Our analysis over these range of data shows some characters such as water outlet temperature (T_{ch_out}) and air supply temperature (T_s) have quick response to this fault compare to other variables on HVAC system. The effect of this fault is presented in figure 6.20 for water outlet temperature of chiller. As we can see this artificial fault could make a notable change between 1 to 2.5 °C on this character based on the outdoor condition. These five point are at starting time and repeat for any two hours for three times then system stay for normal work until night time and another fault has been configured for latest working time. This configuration of the faults make a possibility to review algorithm responsibility at different range of day time. It can also make a possibility to check the system stability during hours of work.

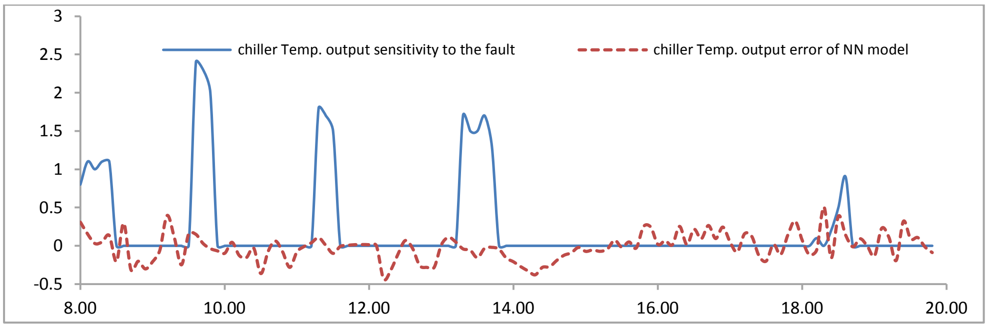


Figure 6.20 cooling tower fault effect via NN model error on T_{ch_out}

Another graph is also reviewed in this figure with signage of red broken line. This graph represent the error between real experimental data and ANN model during daily work. As we explained before the ANN model is developed to use as fault free model for comparison with faulty data. But same as any model this black box model has its own deviation from reality. It is presented here to compare the magnitude of this error with effects of fault on the system variables. But this deviation is as much as low compare to deviation from fault in which we can neglect for detection algorithm. The magnitude of variation is between -0.5 °C to +0.5 °C for model uncertainty and could be neglected compare to 1 to 2.5 °C of fault overshooting. There other also ways in case of significant error form ANN modelling. One of regular ways is to use a mean value of model rather than original values. In this case many fluctuations have been deleted by opposite values due to their negative and positive signage.

This type of fault has also strong effect on the air supply fan temperature compare to previous variable (chiller outlet water temperature). These effects have been configured in figure 6.21for this typical fault.

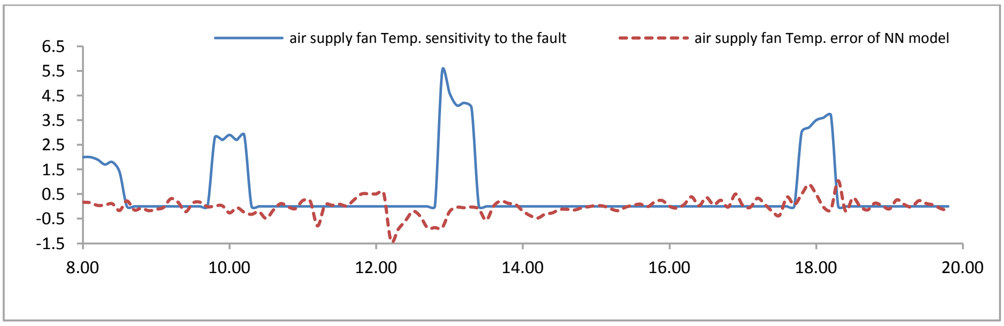


Figure 6.21 cooling tower fault effect via NN model error on T_s

There are four fault signs in this graph but it was more logical if we could see five sign for faults because we had five stop period in reality. In this case we neglect one of the faults to review the effect of detector algorithm in case of out of service sensors. This is regular in reality when operator lose the measurement devices during their normal works. A wide range of change between 2-6 °C is a significant signage on air supply temperature to detector algorithm for check power of fault in the system. Same as last figure there is a comparison for neural network model error and fault error on this output. It can gives us a sense for fault detector sensitivity via model error. As we guess the model error can be neglected compare to fault error. The final results will have an evidence for this assertion.

6.2.3.3.2 Application of Online SVM in Configure with ANN Black Box Model

There are huge amounts of data generated in any industrial case study due to different sections and subsections. Navigation through this databases and find those data which can be more helpful and avoid from unnecessary data can be possible based on data mining techniques and statistical based algorithms. For large datasets like building management systems (BMS) a standard SVM techniques (off line SVM) become infeasible. This motivates to combine such online techniques such as incremental-decremental method toward SVM which we call online SVM. For this specific detection on the HVAC system we developed online SVM and ANN model as represented in figure 6.22. This figure shows the proposed fault detection scheme by using incremental-decremental support vector machine classification. The main purpose of the system is to detect faults by monitoring key HVAC variables. Online incremental training for SVM algorithm has been developed at previous sections and the result of this development is used in this subsection. A coding of this development is presented in appendix A based on MATLAB program which used widely in this research.

In this algorithm cooling tower fault is detect by SVM classifier based on residual rule by comparing of system outputs (chiller water outlet temperature and air supply temperature) with the black box ANN model outputs. Two input and two output model based on previous section has been configured due to its best respond to modelling parameters. This configurations has been reviewed at first of this section

After model was established a label of y is set as +1 (non-faulty) when the error is smaller than a given threshold and it is set as -1 (faulty) when the error is bigger than that threshold. There is a trade-off for threshold selection as it should not become lass than model error or measurement error.

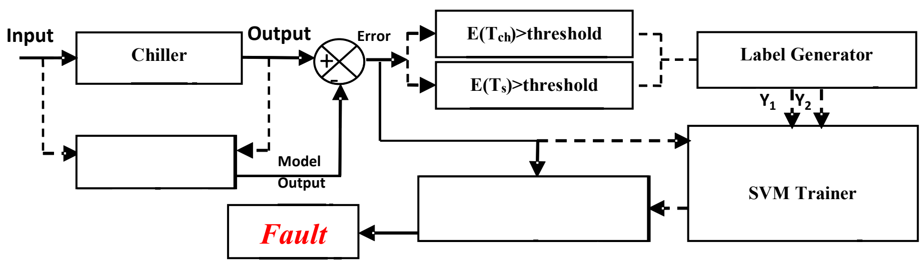


Figure 6.22 Schematic diagram of NN-model based fault detector

For highly safe factor the threshold selected as 1°C to cover all possible errors. This due to avoid any possible error from modelling which was discussed at last section.

Although it may effect on classifier performance but SVM can manage it as a strong classifier.

In this case labels including y1 and y2 are generated with two output variables in which y1 is conducted by outlet water temperature of chiller and y2 is for air supply temperature. The training for classification can be done by chiller's outputs or errors of outputs. But error outputs give us better accuracy on classification as we analysed. Also we got different result for different training sets and labels. The data sets can be contain (errors, y1) or (errors, y2) based on error training. All of these analyses are presented in section C. these labels with their original data will be send to trainer in next step for training of online SVM. Any data with its label is configured as a pair to training section. It is also possible to use any error and its own label as pair for training which difference has been explained in next step.

6.2.3.3.3 Result and Review of Fault on Experimental Case Study

In order to use online SVM classification method to achieve a better isolation performance, two faulty data sets have been used in the classification section. A three-layer SVM classifier is designed, in which the normal and two chiller fault output conditions are all taken into consideration. The kernel function must be properly selected for SVM classifier in order to achieve high classification accuracy. In general, linear function, polynomial function, radial basis function (RBF), sigmoid function, and Gaussian function can be adopted as the kernel function. We tested all possible kernel function to achieve the best response for HVAC fault detection system. There was different result from poor to high performance for specific kernel. But the best response could get from Gaussian function and RBF depend on type of fault in different section. In this research, RBF has been selected as based kernel of online SVM because of its excellent performance in the simulation result.

In this research, two tests are conducted systematically. The diagnosis results and corresponding characteristics of the SVM classifiers are shown in Figures 6.23, 6.24, 6.25 and 6.26. Test 1 is designed to investigate the SVM classifier performance based on pair of output and generated labels. The steady-state output data is used to build the SVM classifier. As mentioned in previous section, the data within the threshold under the normal condition (based on NN model) indicate fault free, and the data beyond the threshold indicate faults.

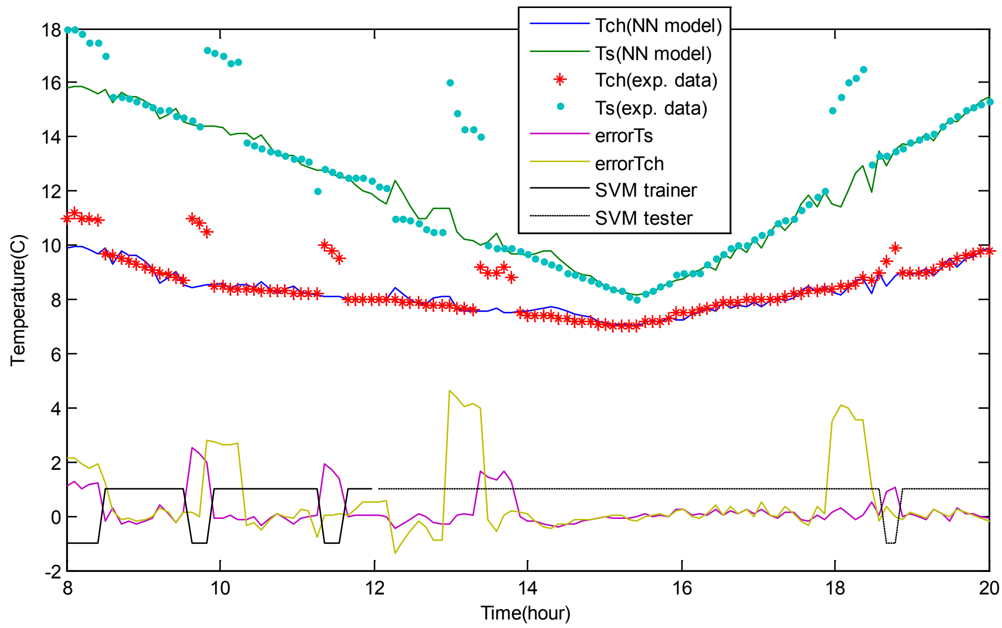


Figure 6.23. SVM fault detection result based on output training and y1

For each normal/faulty condition, one business day data (12 hours data per day between 8am and 20pm) are used. Therefore, a total of 120 samples are collected based on 10 samples per hour. %40 of the data for each condition is used as the training data, whilst the rest are used as the testing data for fault diagnosis. As we can see in all figures, there are samples data for T_{ch_out} and T_s, NN model prediction, the errors based on real data and NN model and SVM classification line.

There are three faults in first %40 of data which used for train and two faults in %60 of rest data for test. The classification line for training part is black-solid line versus black-dash line which represented for test part. As we can see, all different training could response to the fault detection in test part but difference in classification accuracy. The label changes from (+1; non-faulty situation) to (-1; faulty situation), when the fault can be detect.

Figure 6.23 represent the fault detection based on training set of T_{ch} (output temp. of chiller water) and T_s (air supply temp.) with label of y1. First testing fault between hours of 12 to 14 can't be detected by this trainer but other fault between hours of 18-20 was detected based on T_{ch_output}.

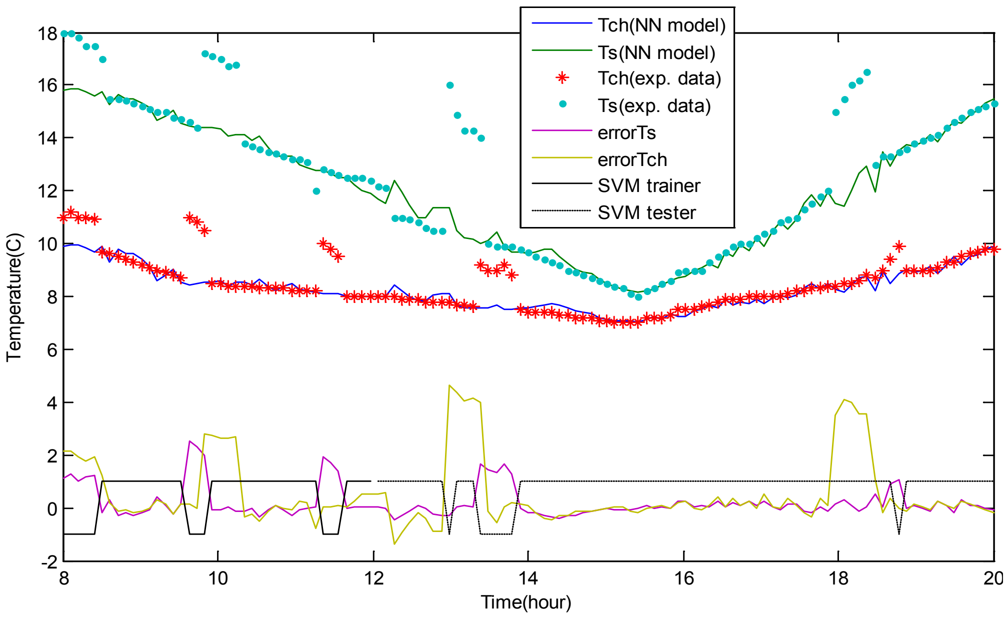


Figure 6.24 SVM fault detection result based on output training and y2

Just by using second label (y2) we give this possibility to SVM for better performance. It is clearly illustrated in figure 6.24 that both faults could be detected with high accuracy based on chiller output temperature but it is in low accuracy based on air supply fan. The second test is application of errors instead of using outputs directly.

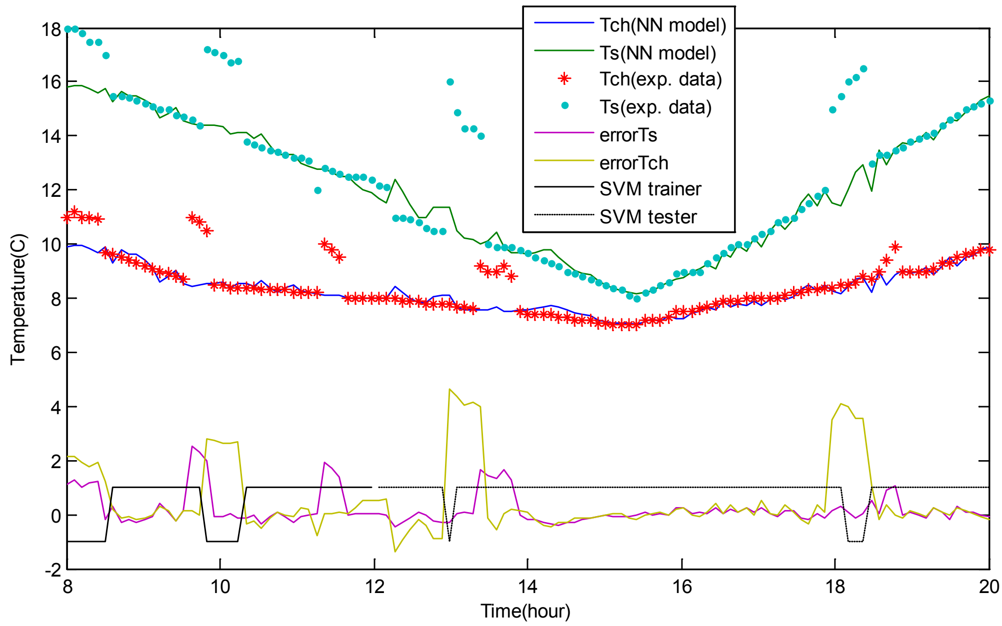


Figure 6.25 SVM fault detection result based on error and y1

This application could not help us for fault detection based on chiller temperature as figure 6.25 shows. But detection rooted in air supply fan highly improved by using error application. In this case we reach %100 of accuracy for fault detection by error based training. This fact can see in figure 6.26 as the error-based train.

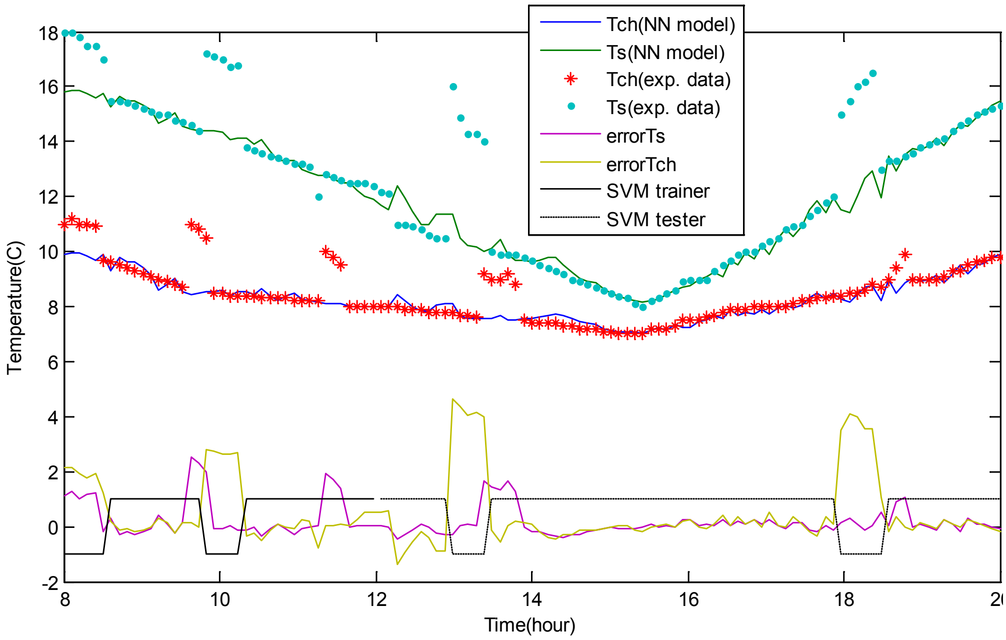


Figure 6.26 SVM fault detection result based on error and y2

6.2.3.4 AHU Fault Analysis

6.2.3.4.1 AHU Artificial Faults Application and Test

HVAC system may suffer from many faults or malfunctions during operation. It is possible to apply this algorithm to whole system on a Building Management System (BMS) or as apart on a device, sensor or actuator. In our test we consider two commonly encountered faults due to some limitation for fault generation. These two faults are: Supply fan fault and Return damper fault.

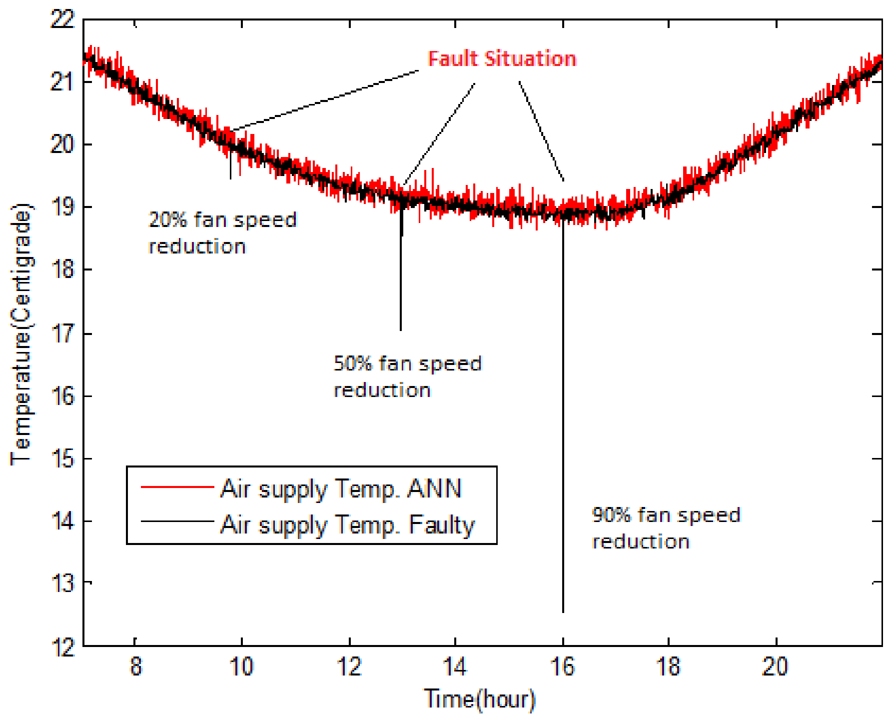


Figure 6.27 sensitivity of supply air temperature to supply fan fault

Air supply fan speed was forced to reduce for three times based on 20%, 50% and 90% of its nominal value. It is tried to make a compatibility between this fault generation and our simulation fault from last sections. This reduction was three minutes for each time, to build the first artificial fault. Return air damper was closed for three times based on 30%, 50% and 70% of its normal work and this reduction was for 15 minutes each time to build the second artificial fault. Definition of these faults was based on our study among most common faults in the AHU part of HVAC system and these faults were used to test the performance of fault detector system. Our records shows measured variable are sensitive to different faults depend on type of the fault. For example supply air temperature (Ts) is strongly sensitive to supply fan fault as shown in figure 6.27.

Another good example is cooling coil temperature (Tc) which has a significant change to return damper fault that is presented in figure 6.28. A sensitivity analysis is established based on sensors respond to the different faults as we can see in figures 6.27 and 8. Then this analysis is sent to fault detector algorithm to automated detection base on its training.

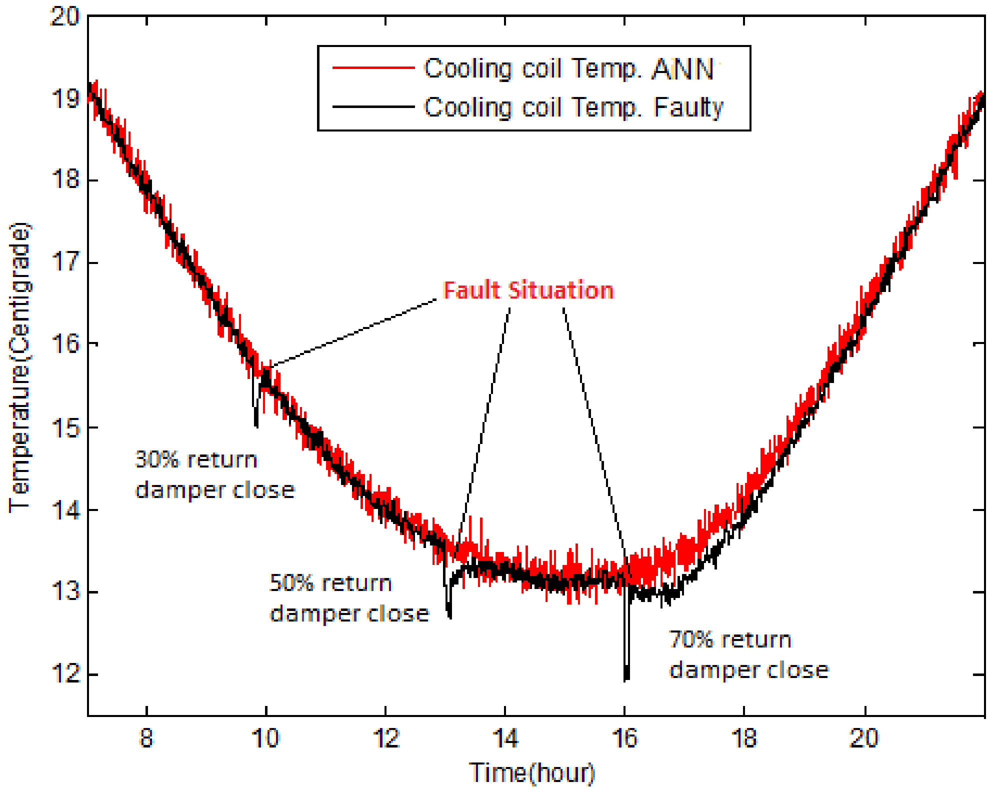


Figure 6.28 sensitivity of cooling coil temperature to supply fan fault

6.2.3.4.2 Fault Diagnosis Approaches

Sensors data are reading by an I/O card that is connected directly to a laptop. Data are stored by MATLAB software in version of 2010b which are installed on the laboratory laptop. The ANN Black box model is generated after finishing of first test in the first day. This model was used as the healthy reference (fault free model) for classifier during the second and third tests. Outdoor conditions for second and third tests were similar to the first test. An example of similar days is when day is sunny and the maximum outdoor temperature for afternoon times (between 2 to 3 pm) is 34 to 35℃.

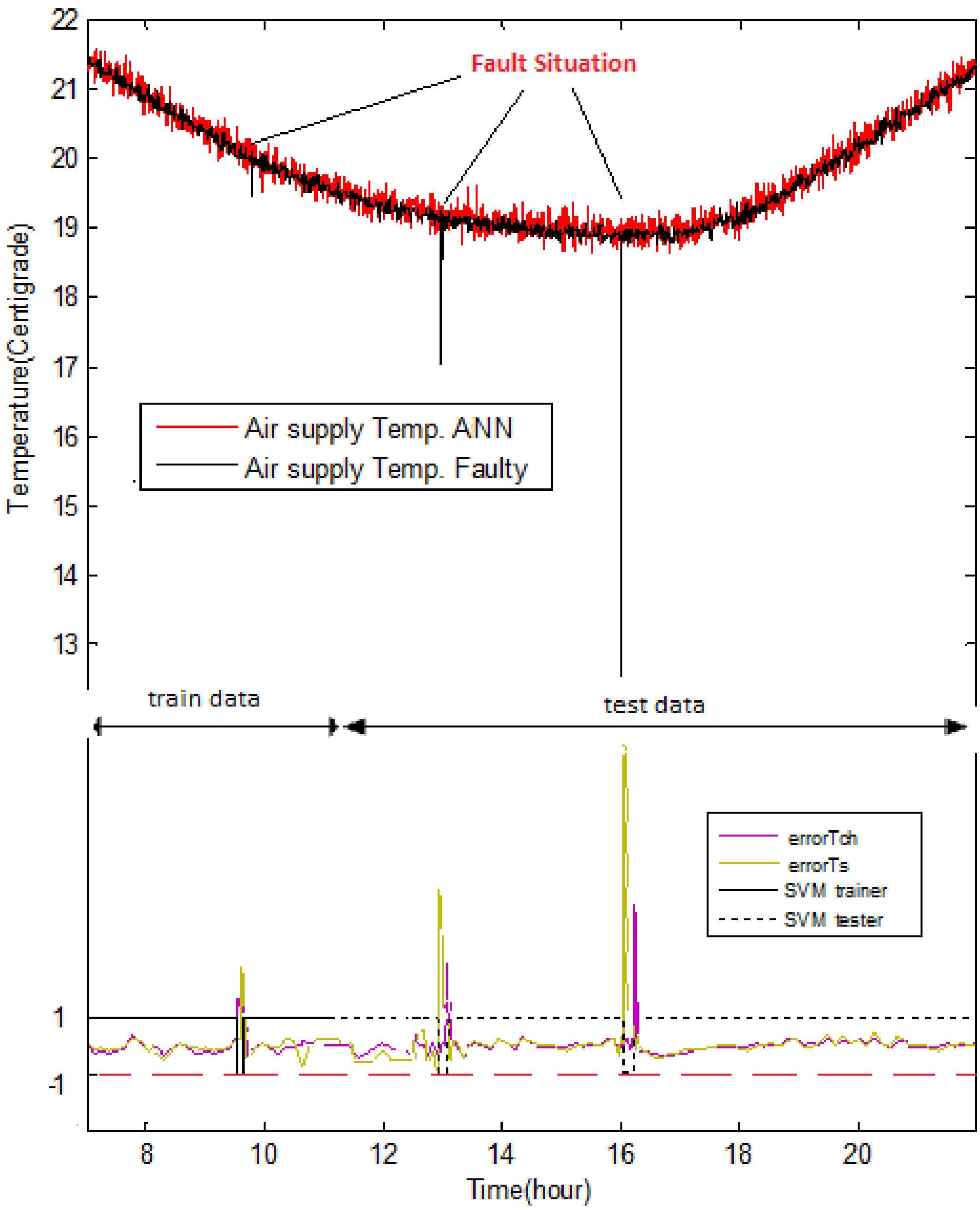


Figure 6.29 result of SVM fault detector for fan fault based on air supply temperature analysis (first fault used for training and two other following faults used for prediction)

Three range of fan speed reduction applied as an artificial fault for second test. So fan speed reduces three times when it was worked at its nominal value. Speed forced to reduce to 80% of nominal value at 10 am for around three minute and then it backs to nominal value. This faulty condition was applied at 1 pm just a difference for 50% reduction in nominal value for second time. A significant fault applied at 4 pm with reducing of speed to 90% of its nominal value for third time. The first fault has been used for training of SVM and two others applied to check the performance of detection area. Two faults in the hours of 13 and 16 have been detected with minimum delay (3 min later) and any error in detection as shown figure 6.29.

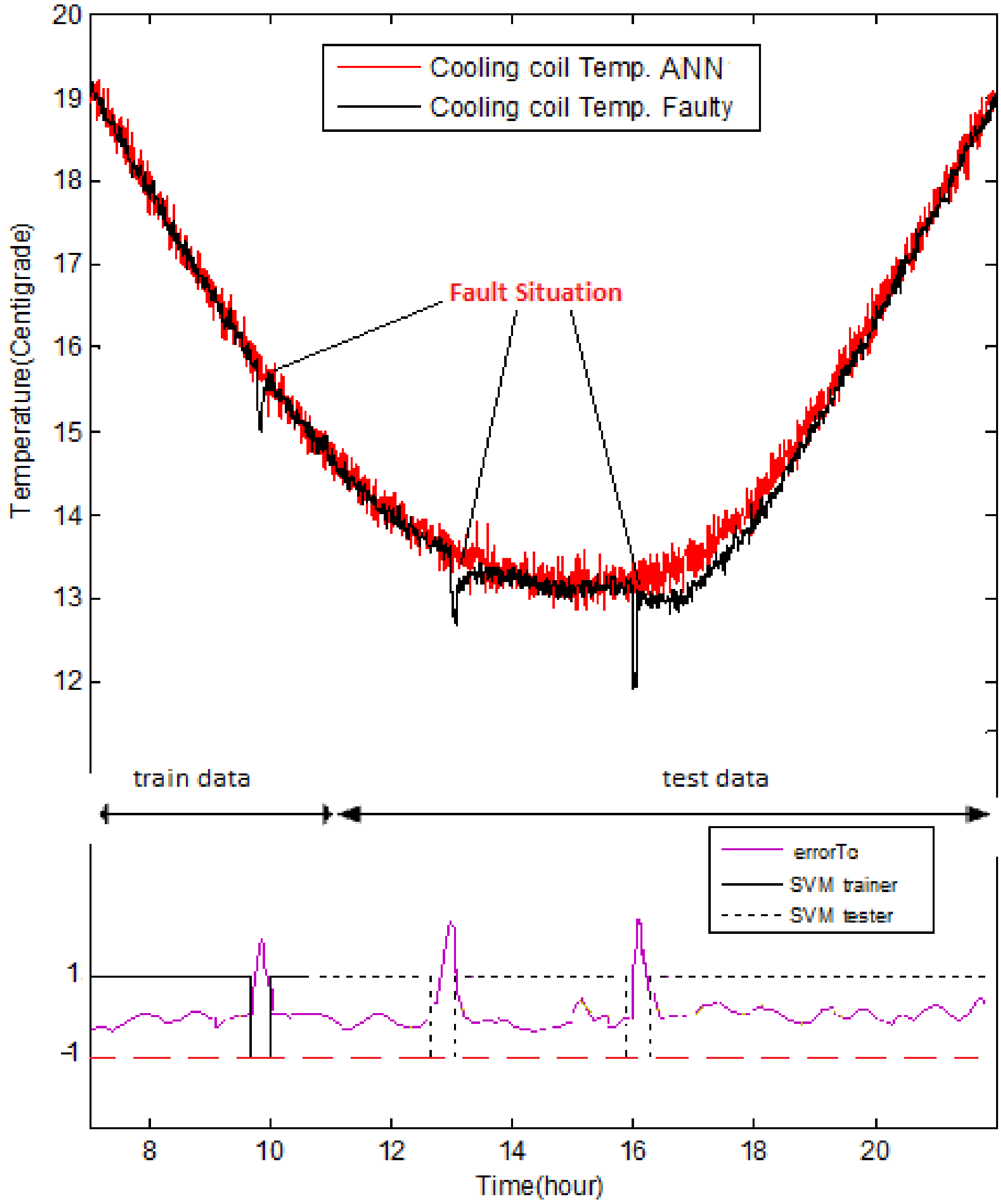


Figure 6.30 result of SVM fault detector for fan fault based on air cooling coil temperature analysis (first fault used for training and two other following faults used for prediction)

It is clear HVAC system is a delay system due to inherently delays of temperature but this proposal based on combine ANN model and online SVM algorithm could detect the faults with minimum delay. It should be noted this delay consider as zero due to long terms delay on HVAC systems. This could be reached because of preparing a healthy reference (fault free reference) for SVM based on black box model. By this method we are now able to increase robustness in SVM for such a delayed system.

Another common fault for a HVAC system has been applied in the next step to increase the confidential of this algorithm. A series of artificial return damper's fault applied in the three fault stage with another similar day. Closing situation of the return damper was change between 30%, 50% and 70% during 10 am, 1 pm and 4 pm for 15 minutes for creating the second artificial fault. This faults return to normal working position after this 15 minutes. The result is shown in figure 6.30 for this type of fault during full day. First fault used to training and two others used for testing same as last procedure. The result increases our confidential for application of this algorithm as we expected. This fault has been detected in just first seconds of occurring and shows the robustness in higher level of achievement.

6.2.4 Conclusion

This section has been tried to focus on testing and analysing of developed algorithms which have been finished on previous sections. These tests were conducted with laboratories operational data to indicate final results of this research through two different approaches for primary and secondary subsystems of HVAC based on laboratories operational data.

First approach for primary subsystem focuses on the fault detection of HVAC cooling tower from chiller observation under real time working conditions. A high accurate model based on NN algorithm with FFBP has been developed for healthy reference of fault detector. Moreover, an online SVM classifier has been applied which can be trained during the operating of the HVAC system. High quality of classification regarding to minimum data application is the first advantage of proposed algorithm. Minimal time delay for fault detecting as well as high speed trainer make this capability to this configuration for using in any subsection of HVAC system. Different with off line method, the proposed approach can even detect new unknown faults for the training of the classifier in real time. Furthermore, this online approach can more efficiently train the FDI modular by throwing out unnecessary data (leave out vectors) and just used a series of data with high priority regarding to classification. Due to these properties, the proposed algorithm can be implemented in every system which needs to high response. Simulation study indicates that the proposed approach can efficiently detect typical HVAC fault.

Second approach based on secondary subsystem of HVAC has been focuses on the robust fault detection of HVAC system under real time working conditions for AHU system. Last online SVM classifier combine with ANN black box model which has been developed was trained during the operating of the HVAC system with AHU data. This configuration could detect easily two main fault of secondary system including fan supply and damper faults.

Different with other algorithms, it is possible to detect any type of faults in HVAC system with minimum delay (few seconds or minutes) and error. Using minimal data for training and increasing SVM accuracy and reliability are now applicable by this combination algorithm. Furthermore, this approach can used as equipment monitoring to shows the device situation of the working. Our result shows detecting a fault by a healthy reference is more accurate compare to training with parameters. In addition, some working conditions such as speed or position can be detected by this algorithm to check the related sensors measurement. Moreover, the proposed algorithm can be implemented in a semi-unsupervised learning frame work due to using a healthy reference model in SVM. Simulation study indicates that the proposed approach can efficiently detect and isolate typical HVAC faults.

Chapter 7

7 Conclusion and Future Work

7.1 Conclusion

There are different sections in any HVAC system whose operations strongly affect building energy efficiency and indoor air quality. There is a need for more robust FDI strategies for all section of HVAC. Different methodology with type of algorithms are available for a FDI procedure and diverse combining can be develop for FDI depend on type of systems and faults. Work and test on real case studies especially for dangerous faults is not logical and not possible in many cases unless a reliable FDI can guarantee to prevent. So it is necessary to develop a trusted simulation model that can produce both fault free and faulty operation data. This also helpful for test and try of different FDI strategy to develop best combination. Therefore, in this study, a dynamic HVAC simulation model that can simulate both fault free and faulty operation data has been developed and tested against different outdoor conditions. Then different faults have been analysed and evaluated based on the model development. In next step two main FDI methods has been reviewed and tested based on simulation data and one was selected to develop. The selected method was SVM which has been developed with online incremental learning specifically for such time delay systems like HVAC. Finally this robust FDI methodology that contains black box model and online FDI methods is developed and tested using experimental data and simulated data.

The dynamic simulation model developed in this study is based on heat, mass and momentum balance between different sections of HVAC. There are some major differences between this model and other modelling systems, including differences in components details and control systems. Values for nearly all parameters used in the simulation models re-determined based on some experimental data from UTS HVAC laboratory system. Some of the parameters are identified directly from the manufacturer catalogue data provided by the ERS and others needed to be calculated based on the catalogue information. Pressure resistance values and damper coefficients calculated from catalogue data cause large air flow rate simulation errors. Therefore, pressure resistance experiment is designed and performed using the laboratory test facility. Using system operation data collected from this study, the developed HVAC model is validated against some pervious experimental data and shows satisfactory performance.

A fault modelling methodology is developed in this study, in which faults are categorized as sensor faults, controller device faults, equipment faults, and controller faults. All common AHU faults are modelled and analysed in this study. A sensitivity to different faults has been developed to identify best sensitive characters for a specific fault. Experimental data generated from this study are used to validate the performance of developed FDI system. It is found that the developed FDI algorithm in this study are able to replicate typical cooling tower and AHU fault symptoms.

Healthy reference is necessary to differentiate measurement fluctuations caused by thermal load condition changes from by faults. The feasibility of using ANN black box model for healthy references has been demonstrated in this study. A FFBB learning technique in an ANN model is developed. Experimental data from two main section of HVAC system including primary and secondary systems are used to test the model prediction. It is found that online SVM method does not require much training data and can effectively detect abrupt faults. The performance goes up if healthy reference in combination with online SVM detector used.

ANN black box model is developed as a promising tool for detecting degradation faults. It can also help for case of untrained faults which normally can categorized as unsupervised fault detector. Our record shows a future fault can be detected by combination of ANN black box model and online SVM detector even the fault has not been trained before.

In general, a dynamic HVAC simulation model that is capable of producing fault free and faulty operation data for analysing of different fault sensitivity is developed in this study. Secondly, a practical FDI technique for HVAC system is develop and validated with both fault-free and faulty experimental data. The FDI model can be developed only based on fault-free measurement data and is feasible for load fluctuations caused by weather and internal load changes. Moreover, the FDI model is capable to handle different types of faults, including sensor or process faults, abrupt or degradation faults. It can use for such untrained faults or as unsupervised technique.

7.2 Future Work

The findings of this study supply a good data driven method for automatic HVAC fault detection, more research is needed in the future to enhance the capabilities of fault diagnostics. The recommendations for the future study include:

1. The developed dynamic building zone for multi zone should also be developed and validate.
2. More systematic validation is needed for the developed HVAC FDI methodology, especially investigating a) how much training data is sufficient for different faults; b) how is the performance in start-up and commissioning procedure; c) how is the performance in industrial case study with minimum sensors and data.
3. Unsupervised method should be developed and test with more case study. The validation of ANN black box model should be reviewed in many different case study to prof the capability of the algorithm in heavy tasks.
4. A fault symptom library can be developed based on the simulation data and validate with experimental data.
5. The developed FDI methodology for HVAC system is very genetic. The use of this method on other systems should be examined.

The completion of the above recommendations, as well as others, would improve the accuracy of the dynamic simulation model and expand the capabilities of the FDI methodology developed in this study.

Bibliography

Agami T., Jia, H., Yongzhong, P.and Reddy R. Characteristic physical parameter approach to modelling chillers suitable for fault detection, diagnosis, and evaluation [Journal] // Journal of Solar Energy Engineering, Transactions of the ASME. - 2012. - 3 : Vol. 125. - pp. 258-265.

Agami T.and Reddy R. Application of a generic evaluation methodology to assess fourdifferent chiller FDD methods [Journal] // HVAC and R Research. - 2007. - 5 : Vol. 13. - p. 711¬729.

Annex 25 Building optimization and fault diagnosis source book [Book]. - [s.l.] : International Energy Agency for Real Time Simulation of HVAC Syetems for Building Optimization, Fault Detection and Diagnosis, 2009.

Aravena J. Detecting change using pseudo power signatures [Conference] // Proc. 2002 IFAC Congress. - Barcelona, Spain : [s.n.], 2002.

Arbor A., Harutunian, M. Functional model and second law analysis method for energy efficient process design: applications in HVAC systems design [Book]. - [s.l.] : UMI Dissertation Services, 2003.

Ardehali M.M., Smith, T.F., House, J.M. and Klaassen, C.J. Building Energy Use and Control Problems: An Assessment of Case Studies [Journal]. - [s.l.] : ASHRAE , 2003. - Vol. 109.

Arguello-Serrano B. and Velez-Reyes, M. Nonlinear control of a heating, ventilating, and air conditioning system with thermal load estimation [Journal]. - [s.l.] : IEEE Trans. Control Systems Tech., 1999. - 1 : Vol. 7. - pp. 56–63.

Arthur A. HVAC design portfolio: 865 airside systems flow diagrams and details [Book]. - New York : McGraw-Hill, 2003.

ASHRAE DOE Plan Projects Savings for New Building Energy Costs [Journal]. - [s.l.] : ASHRAE, 2000.

Bailey M.B. The design and viability of a probabilistic fault detection and diagnosis method for vapor compression cycle equipment. - Colorado, USA : Ph.D. thesis, School of Civil Engineering of University of Colorado, 2008.

Bakshi B. R. and Stephanopoulos, G. Wave-net: a multiresolution, hierarchical neural network with localized learning [Journal]. - [s.l.] : American Institute of Chemical Engineers Journal, 2003. - 1 : Vol. 39. - pp. 57-81.

Bartlett P. L. and Taylor, J. Generalization performance of support vector machines and other pattern classifiers. In B. Sch¨olkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in Kernel Methods — Support Vector Learning [Book]. - Cambridge : Cambridge University, 2000. - pp. 43–54.

Basseville M. and Nikiforov, I. Fault isolation for diagnosis: nuisance rejection and multiple hypotheses testing [Conference] // 15th IFAC World Congress, IFAC. - Barcelona : [s.n.], 2002.

Becraft W. and Lee, P. An integrated neural network/expert system approach for fault diagnosis [Journal]. - [s.l.] : . Computers and Chemical Engineering, 2003. - 10 : Vol. 17. - pp. 1000-1014.

Bendapudi S. and Braun, J. E. Development and Validation of a Mechanistic, Dynamic Model for a Vapor Compression Centrifugal Liquid Chiller, [Journal]. - Atlanta, GA : ASHRAE, 2002. - Vols. Report #4036-4.

Bennett K. P. and Duality, J. geometry in SVM classifiers [Conference] // Proceedings of the 17th International Conference on Machine Learning. - San Francisco, California : [s.n.], 2000. - pp. 57–64.

Bourdouxhe J.P., Grodent, M. , and Lebrun J. HVAC1 Toolkit: Algorithms and Subroutines for Primary HVAC Systems Energy Calculations. - [s.l.] : Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 2007.

Braun J., Chaturvedi ,N. An inverse gray-box model for transient building load prediction [Journal] // HVAC and R Research. - 2010. - 1 : Vol. 8. - pp. 73-99.

Braun J.E. Automated fault detection and diagnostics for vapor compression cooling equipment [Conference]. - [s.l.] : ASME Transactions, 2002. - Vol. 31. - pp. 278-296.

Braun J.E., Lawrence, T., Mercer, K., and Li, H. Fault Detection and Diagnostics for Rooftop Air Conditioners.. - Sacramento, California : California Energy Commission, 2003. - Vols. P-500-03-096-A1.

Breuker M.S. and Braun J.E. Common faults and their impacts for rooftop air conditioners [Journal] // International Journal of Heating, Ventilating, Air Conditioning and Refrigerating Research. - 2008a. - 3 : Vol. 4. - pp. 303-318.

Breuker M.S. and Braun J.E. Evaluating the performance of a fault detection and diagnostic system for vapor compression equipment [Journal] // International Journal of Heating, Ventilating, Air Conditioning and Refrigerating Research. - 2008b. - 3 : Vol. 4. - pp. 401-425.

Breuker M.S. Evaluation of a statistical, rule-based fault detection and diagnostics method for vapor compression air conditioners. - Purdue, Indiana : Master's thesis, School of Mechanical Engineering, Purdue University, 2007.

Burges C.J.C. A tutorial on support vector machines for pattern recognition [Journal]. - [s.l.] : Data Mining and knowledge discovery, 2009. - 2 : Vol. 2.

Carling P. Comparison of three fault detection methods based on field data of an air-handling unit [Journal]. - [s.l.] : ASHRAE Transactions, 2010. - 1 : Vol. 108.

Castro N. Performance evaluation of a reciprocating chiller using experimental data and model predictions for fault detection and diagnosis [Journal]. - [s.l.] : ASHRAE Transactions 108(1)., 2002. - 1 : Vol. 108.

Cauwenberghs G. and Poggio, T. Incremental and decremental support vector machine learning [Journal]. - [s.l.] : Advances in Neural Information Processing Systems, 2001. - Vol. 13. - pp. 409-415.

Cerqueira A.S., , et al. Power quality events recognition using a SVM-based method [Journal]. - [s.l.] : Electric Power Systems Research, 2008. - 9 : Vol. 79. - pp. 1546-1552.

Chen B. and Braun, J.E. Evaluation the potential on-line fault detection and diagnostics for rooftop air-conditioners [Conference] // 2000 International Refrigeration Conference. - W. Lafayette, IN. : Purdue University, 2000.

Chen J. and Patton, R.J. Robust Model-based Fault Diagnosis for Dynamic Systems [Book]. - Boston : Kluwer Academic Publishers, 1999.

Chen R.J. and Zhang, H. Design of unknown input observers and robust fault detection filters [Journal]. - [s.l.] : International Journal of Control, 2007. - 1 : Vol. 63. - pp. 85-105.

Cheung J. T. Representation and extraction of trends from process data. - Cambridge, MA : Doctor of Science Thesis, Massachusetts Institute of Technology, 2002.

Chew H. G., Bogner, R.E. and Lim, C.C Dual support vector machine with error rate and training size biasing [Conference] // In Proceedings of ICASSP. - 2001. - pp. 1269–72.

Chew H. G., Lim, C. C. and Bogner, R. E. An implementation of training dual-nu support vector machines [Book]. - Kluwer : Optimization and Control with Applications, 2003.

Cho S.H., Yang, H.C. Zaheer-uddin, M. and Ahn, B.C. Transient pattern analysis for fault detection and diagnosis of HVAC system [Journal]. - [s.l.] : Energy Conversion and Management, 2005. - Vol. 46. - pp. 3103-3116.

Chung D. and Modarras, M. A method of fault diagnosis: presentation of a deepknowledge system [Report]. - New York, NY : AIChE National Meeting, 2010.

Cibse B. and Guide. H. Buiding control systems [Journal]. - [s.l.] : Oxford, 2000.

Colin J. and Houdas, Y. Experimental determination of coefficient of heat exchanges by convection of human body [Journal]. - [s.l.] : Journal of applied Physiology, 1999. - 1 : Vol. 22. - pp. 31–38.

Comstock M.C., Braun, J.E., and Groll, E.A. A survey of common faults in chillers [Journal]. - [s.l.] : ASHRAE Transactions, 2002. - 1 : Vol. 108.

Comstock M.C., Chen, B. and Braun, J.E. Literature Review for Application of Fault Detection and Diagnostic Methods to Vapor Compression Cooling Equipment [Report]. - [s.l.] : HL 1999-19, Report #4036-2, Ray Herrick Laboratories, Purdue University, 1999.

Cristianini N. and Shawe, J. Support Vector Machines and other kernel-based learning methods [Book]. - [s.l.] : Cambridge University Press, 2000.

Dash S., Rengaswamy, R. and Venkatasubramanian, V. A novel interval-halving algorithm for process trend identification [Conference]. - Jejudo Island, Korea : The 4th IFAC Workshop on On-line Fault Detection and Supervision in the Chemical Process Industries, 2001. - pp. 173-178.

Dexter A., Pakanen, J. Demonstrating Automated Fault Detection and Diagnosis Methods in Real Buildings [Book]. - Espoo, Finland : VIT, 2001. - Vol. IEA Annex 34.

Dexter A.L. and Ngo, D. Fault diagnosis in air-conditioning systems: a multi-step fuzzy model approach [Journal]. - [s.l.] : International Journal of HVAC&R Research, 2001. - Vol. 7. - pp. 83-102.

DOE Buildings Energy Data Book [Book]. - U.S. Department of Energy : [s.n.], 2003.

Donald B. Andrew T. Wavelet methods for time series analysis [Book]. - Cambridge; New York : Cambridge University Press, 2009.

Dote Y., Ovaska, S.J. and Gao, X.Z. Fault detection using RBFN- and AR-based general parameter methods. [Conference] // Ieee International Conference on Systems, Man, and Cybernetics. - 2002. - Vols. 1-5. - pp. 77-80.

Dounis A.I. and Manolakis, D.E. Design of a fuzzy system for living space thermal comfort regulation [Journal]. - [s.l.] : Applied Energy, 2001. - 2 : Vol. 62. - pp. 119–144.

Du R., Chen, Y.D. and Chen, Y.B. Four dimensional holospectrume a new method for analyzing force distributions", [Journal]. - [s.l.] : ASME Transactions: Journal of Manufacturing Engineering and Science, 2006. - Vol. 119. - pp. 95-104.

Du R., Gao, R. and Wang, L.H. Monitoring and Diagnosis of Sheet Metal Stamping Processes, Condition-based Monitoring and Control for Intelligent Manufacturing, in: [Book]. - Verlag, New York : Springer, 2005.

Du Z.M., Jin, X.Q. and Fan, B. Fault Diagnosis for Sensors in HVAC Systems Using Wavelet Neural Network [Conference] // Proceedings of the 4th Asian Conference on Refrigeration and Air-Conditioning. - 2009. - pp. 409-415.

Eskin E. Anomaly detection over noisy data using learned probability distributions [Conference]. - [s.l.] : Proceedings of the Seventeenth International Conference on Machine Learning, 2000.

Federspiel C.C. User-adaptable and minimum-power thermal comfort control. - Cambridge, MA : PhD dissertation, Dept. Mech. Eng., Massachusetts Institute of Technology, 2002.

Geramanis A. Kanarachos and K. Multivariable control of single zone hydronic heating systems with neural networks [Journal]. - [s.l.] : Energy Conversion and Management, 2008. - 13 : Vol. 39. - pp. 1317–1336.

Gertler A. Intelligent supervisory control. Artificial Intelligence Handbook [Book]. - NC, USA : Research Triangle Park, 2009. - Vol. II.

Ghiaus C. Fault diagnosis of air-conditioning systems based on qualitative bond graph [Journal]. - [s.l.] : Energy and Buildings, 1999. - Vol. 30. - pp. 221-232.

Glass A.S., Gruber, P., Roos, M. and Todtli, J. Qualitative model-based fault detection in air-handling units [Journal]. - [s.l.] : EEE Control Systems Magazine, 2005. - 4 : Vol. 15. - pp. 11-22.

Gordon J.M. and Ng., K.C. Predictive and diagnostic aspects of a universal thermodynamic model for chillers [Journal]. - [s.l.] : International Journal of Heat and Mass Transfer, 2005. - 5 : Vol. 38. - pp. 807-18.

Gordon J.M. and Ng., K.C. Thermodynamic modeling of reciprocating chillers [Journal]. - [s.l.] : Journal of Applied Physics, 2004. - 6 : Vol. 75. - pp. 2769-74.

Grimmelius H.T., Woud, J.K. and Been G. On-line failure diagnosis for compression refrigerant plants [Journal] // International Journal of Refrigeration. - 2009. - 1 : Vol. 18. - pp. 31-41.

Grimmelius H.T., Woud, J.K. and Been, G. On-line failure diagnosis for compression refrigerant plants [Journal]. - [s.l.] : International Journal of Refrigeration 18(1), 2005. - 1 : Vol. 18. - pp. 31-41.

Henao H., et al. An improved signal processing-based fault detection technique for induction machine drives [Conference] // The 29th Annual Conference of the Ieee Industrial Electronics Society. - 2003. - Vols. 1-3. - pp. 1386-1389.

Henk C. and Luc, L. ARX models and real-time model-based diagnosis [Journal]. - [s.l.] : ASHRAE Transactions, 2007. - 1 : Vol. 103. - pp. 657-671.

Herbert W., Stanford, B. HVAC water chillers and cooling towers fundamentals, application, and operation [Book]. - New York : Marcel Dekker, 2003.

Hittle D. The building loads analysis and system Thermodynamics program, BLAST", [Report]. - Illinois : US Army Construction engineering research Laboratory, 2007.

Hou Z.J., Lian, Z.W., Yao, H. and Yuan, X.J. Data mining based sensor fault diagnosis and validation for building air conditioning system [Journal]. - [s.l.] : Energy Conversion and Management, 2006. - Vol. 47. - pp. 2479-2490.

House J.M. and Kelly, G.E. An overview of building diagnostics, in [Conference]. - Gaithersburg, USA : National Conference on Building Commissioning and Diagnostics for Commercial Buildings, 2000. - pp. 1-9.

House J.M., Lee, W.Y. and Shin, D. R. Classification techniques for fault detection and diagnosis of an air-handling unit [Journal]. - [s.l.] : ASHRAE Transactions, 1999. - 1 : Vol. 105. - pp. 1087-1097.

Huang Y. J. LBL, DOE-2 user guide, version 2.1, LBL report No. LBL-8689 Rev. 2 [Report]. - Berkeley, California : Lawrence Berkeley Laboratory, 2000.

Hydeman M., Webb, N., Sreedharan, P. and Blanc, S. Development and testing of a reformulated regression-based electric chiller model [Journal]. - [s.l.] : ASHRAE Transactions, 2012. - 2 : Vol. 108. - pp. 1118-1127.

Hyvarinen J., Karki, S. Building Optimization and Fault Diagnosis [Book]. - Finland : McGraw-Hill, 1996. - Vol. IEA Annex 25.

Inatsu H., Matsuo, H., Fujiwara, K., Yamada, K. and Nishizawa, K. Development of refrigerant monitoring systems for automotive air-conditioning systems [Journal]. - [s.l.] : Society of Automotive Engineers, 2002. - Vol. SAE Paper No. 920212..

Johannesmeyer M., Singhal, A., Seborg, D. Pattern matching in historical data [Journal]. - [s.l.] : AIChE J, 2002. - Vol. 48. - pp. 2022–2038.

Katipamula S. and Brambley, M.R. Methods for fault detection, diagnostics and prognostics for building systems- A review, Part I [Journal] // IJHVAC&R Research. - 2005a. - 1 : Vol. 11. - pp. 3-25.

Katipamula S. and Brambley, M.R. Methods for fault detection, diagnostics and prognostics for building systems- A review, Part II [Journal] // IJHVAC&R Research. - 2005b. - 2 : Vol. 12. - pp. 169-187.

Katipamula S. and Claridge, DE. Use of simplified systems model to measure retrofit energy savings [Journal]. - [s.l.] : Transactions of the ASME Journal of Solar Energy Engineering, 2003. - pp. 57-68.

Kim M. and Kim, M.S. Performance investigation of a variable speed vapor compression system for fault detection and diagnosis [Journal]. - [s.l.] : International Journal of Refrigeration, 2005. - Vol. 28. - pp. 481-488.

Kim M., Kim, M.S. Performance investigation of a variable speed vapor compression system for fault detection and diagnosis [Journal]. - [s.l.] : International Journal of Refrigeration, 2005. - Vol. 28. - pp. 481-488.

Kim P.S. and Lee, E.H. A new parity space approach to fault detection for general systems. [Conference]. - [s.l.] : High Performance Computing and Communications, 2005. - Vol. 3726. - pp. 535-540.

Knebel DE. Simplified energy analysis using the modified bin method [Journal]. - Atlanta, GA : ASHRAE, 2003.

Konstantinov K. B. and Yoshida, T. Real-time qualitative analysis of the temporal shapes of the (bio) process variables [Journal]. - [s.l.] : American Institute of Chemical Engineers Journal, 2009. - 11 : Vol. 38. - pp. 1703-1715.

Kourti T Application of latent variable methods to process control and multivariate statistical process control in industry [Journal]. - [s.l.] : Journal of Adaptive Control and Signal Processing, 2004. - 12 : Vol. 29. - pp. 211-221.

Krzanowski W. J. Between-groups comparison of principal components [Journal]. - [s.l.] : J. Amer.Stat. Assoc., 2008. - 367 : Vol. 74. - pp. 703-707.

Kumamaru. T., Utsunomiya, T., Iwasaki, Y., Shoda, I. and Obayashi, M. A fault diagnosis systems for district heating and cooling facilities [Conference] // International Conference on Industrial Electronics, Control, and Instrumentation. - Kobe, Ja : [s.n.], 2001.

Kusuda T. Comparison of energy calculation procedures [Journal]. - [s.l.] : ASHRAE Journal, 2001. - 8 : Vol. 23. - pp. 21-24.

Lee W., Stolfo, S. and Mok, K. Adaptive intrusion detection: a data mining approach, Artificial Intelligence Review, Kluwer Academic Publishers [Journal]. - 2000. - 6 : Vol. 14. - pp. 533-567.

Lee W., Stolfo, S. J. and Mok, K. Data mining in work flow environments: Experiences in intrusion detection [Conference] // Proceedings of Conference on Knowledge Discovery and Data Mining (KDD-99). - 2009.

Lee W.Y. and John, M. Subsystem level fault diagnosis of a building's air-handling unit using general regression neural networks [Journal]. - [s.l.] : AppliedEnergy, 2004. - 1 : Vol. 77. - pp. 153–170.

Lee W.Y., House, J.M. and Kyong, N.H. Subsystem level fault diagnosis of a building's air-handling unit using general regression neural networks [Journal]. - [s.l.] : Applied Energy, 2004. - Vol. 77. - pp. 153-170.

Lee W.Y., House,J.M., Park, C. and Kelly, G.E. Fault diagnosis of an air-handling unit using artificial neural networks [Journal]. - [s.l.] : ASHRAE Transactions, 1996b. - 1 : Vol. 102. - pp. 540-549.

Lee W.Y., Park, C. and Kelly, G.E. Fault detection of an air-handling unit using residual and recursive parameter identification methods [Journal]. - [s.l.] : ASHRAE Transactions, 1996a. - 1 : Vol. 102. - pp. 528-539.

Liang J. and Du, R. Model-based Fault Detection and Diagnosis of HVAC systems using Support Vector Machine method [Journal]. - [s.l.] : International Journal of Refrigeration, 2007. - 6 : Vol. 30. - pp. 1104-1114.

Lippmann R. P. An introduction to computing with neural nets [Journal]. - [s.l.] : IEEE ASSP Magazine, 2004. - Vol. 4. - pp. 4-22.

Liu M. and Claridge, DE. Use of calibrated HVAC system models to optimize system operation [Journal]. - [s.l.] : ASME Journal of Solar Energy Engineering, 2008. - Vol. 120. - pp. 12-31.

Lo C.H., et al. Fuzzy-genetic algorithm for automatic fault detection in HVAC systems [Journal]. - [s.l.] : Applied Soft Computing, 2007. - 2 : Vol. 7. - pp. 554-560.

Lu N., Fuli, W., Furong, G. Combination method of principal component and wavelet analysis for multivariate process monitoring and fault diagnosis [Journal]. - [s.l.] : Industrial and Engineering Chemistry Research, 2003. - 18 : Vol. 42. - pp. 4198-4207.

MacGregor J.F. and Kourti, T. Statistical process control of multivariate processes [Journal]. - [s.l.] : Control Engineering Practice, 2009. - Vol. 3. - pp. 403–414.

Mallat S. A theory for multiresolution signal decomposition: the wavelet representation [Journal]. - [s.l.] : IEEE Pattern Anal. and Machine Intell, 2010. - 7 : Vol. 11. - pp. 674-693.

Mardia K.V., Kent, J.T. and Bibby J.M. Multivariate Analysis [Book]. - [s.l.] : London: Academic, 2006.

McKellar M.G. Failure diagnosis for a household refrigerator. - Purdue, Indiana : Master's thesis, School of Mechanical Engineering, Purdue University, 2007.

Moya M. and Hush, D. Network constraints and multiobjective optimization for oneclass classification [Journal]. - [s.l.] : . Neural Networks, 2006. - 3 : Vol. 9. - pp. 463–474.

Moya M., Koch, M. and Hostetler, L. One-class classifier networks for target recognition applications [Conference] // Proceedings World Congress on Neural Networks. - 2003. - pp. 797–801.

Norford L.K., Wright, J.A., Buswell, R.A., Luo, D., Klaassen, C.J. and Suby, A. Demonstration of fault detection and diagnosis methods for air-handling units [Journal]. - [s.l.] : International Journal of Heating, Ventilating, Air Conditioningand Refrigerating, ASHRAE 1020-RP, 2002. - 1 : Vol. 8. - pp. 41-47.

Ortega F., Ordieres, J. B., Menendez, C. and Gonzalez, N.C. Development of neural-based diagnostic system to control the ropes of mining shifts [Conference] // Proceedings of the International Conference in Ales,. - France : [s.n.], 2005. - pp. 108-111.

Pakanen J.E. and Sundquist, T. Automation-assisted fault detection of an air-handling unit; implementing the method in a real building [Journal]. - [s.l.] : Energy and Buildings, 2003. - Vol. 35. - pp. 193-202.

Pandit S.M. and Wu, S.M. Time Series and System Analysis with Applications [Book]. - New York : John Wiley and Sons, 2003.

Peitsman H.C. and Bakker, V. Application of black-box models to HVAC systems for fault detection [Journal]. - [s.l.] : ASHRAE Transactions, 2006. - 1 : Vol. 102. - pp. 628-640.

Peitsman H.C. and Bakker, V. Application of black-box models to HVAC systemsfor fault detection [Journal] // ASHRAE Transactions. - 2009. - 1 : Vol. 102. - pp. 628-640.

Qin S. J. Statistical Process Monitoring: Basics and Beyond [Journal]. - [s.l.] : Journal of Chemometrics, 2003. - 480 : Vol. 17.

Ralston P., DePuy, G., Graham, J. H. Computer-based monitoring and fault diagnosis: a chemical process case study [Journal]. - [s.l.] : ISA TRANSACTIONS, 2011. - 1 : Vol. 40. - pp. 85-98.

Reddy T. and Agami, K. Application of a generic evaluation methodology to assess four different chiller FDD methods (RP-1275), " , v 13, n 5, p [Journal]. - [s.l.] : HVAC and R Research, 2007. - 5 : Vol. 13. - pp. 711-729.

Reddy T.A., Niebur, D., Gordon, J., Seem, J., Andersen, K.K., Cabrera, G., Jia, Y. and Pericolo, P. Development and comparison of on-line model training techniques for model-based FDD methods applied to vapor compression chillers [Report]. - [s.l.] : ASHRAE R, 2001.

Reiter R. A theory of diagnosis from first principles [Journal]. - [s.l.] : Artificial Intelligence, 2008. - 1 : Vol. 332. - pp. 57-95.

Rich S. H. and Venkatasubramanian, V. Model-based reasoning in diagnostic expert systems for chemical process plants [Journal]. - [s.l.] : Computers and Chemical Engineering, 2005. - Vol. 11. - pp. 111-122.

Ritter G. and Gallegos, M. Outliers in statistical pattern recognition and an application to automatic chromosome classification [Conference]. - [s.l.] : Pattern Recognition Letters, 2007. - Vol. 18. - pp. 525–539.

Rossi T. and Braun, J. Statistical, rule-based fault detection and diagnosticmethod for vapor compression air conditioners [Journal] // HVAC&R Research. - 2007. - 1 : Vol. 3. - pp. 19-37.

Rossi T.M. and Braun, J.E. A statistical, rule-based fault detection and diagnostic method for vapor compression air conditioners [Journal]. - [s.l.] : International Journal of Heating, Ventilating, Air Conditioning and Refrigerating Research, 2007. - 1 : Vol. 3. - pp. 19-37.

Rossi T.M. and Braun, J.E. Minimizing operating costs of vapor compression equipment with optimal service scheduling [Journal]. - [s.l.] : International Journal of Heating, Ventilating, Air Conditioning and Refrigerating Research, 2006. - 1 : Vol. 2. - pp. 3-26.

Rossi T.M. Detection, diagnosis, and evaluation of faults in vapor compression cycle equipment. - Purdue, Indiana : Ph.D. thesis, School of Mechanical Engineering, Purdue University, 2005.

Salsbury T.I. and Diamond, R.C. Fault detection in HVAC systems using model-based feedforward control [Journal]. - [s.l.] : Energy and Buildings, 2001. - Vol. 33. - pp. 403-415.

Salsbury T.I. and Diamond, R.C. Fault detection in HVAC systems using model-based feedforward control. [Journal]. - [s.l.] : Energy and Buildings, 2001. - 4 : Vol. 33. - pp. 403-415.

Sch¨olkopf B. , Smola, A. J., Williamson, R. C. and Bartlett, P. L. New support vector algorithms [Journal]. - [s.l.] : Neural Computation, 2000. - Vol. 12. - pp. 1207–1245.

Sch¨olkopf B. and Smola, A. J. Learning with Kernels [Book]. - Cambridge, MA : MIT Press, 2002.

Sch¨olkopf B., Burges, C. and Smola, A. J. Advances in Kernel Methods - Support Vector Learning [Book]. - Cambridge, MA : MIT Press, 2008.

Sch¨olkopf B., unchen, M. Support Vector Learning [Book]. - Berline : Technische University at Berline, 2007.

Schein S., Jeffrey, H., John, M. Application of control charts for detecting faults in variable-air-volume boxes [Journal]. - [s.l.] : ASHRAE Transactions, 2003. - 2 : Vol. 109. - pp. 671-682.

Scholkopf B., Williamson, R.C., Smola, A.J., Shawe Taylor and Platt, J. Support vector method for novelty detection [Journal]. - [s.l.] : Neural Information Processing Systems, 2008. - pp. 582-588.

Simani S., Fantuzzi, C. and Patton, R.J. Model-based Fault Diagnosis in Dynamic Systems using Identification Techniques [Book]. - New York : Springer, 2003.

Singhal A, Seborg, D. Pattern matching in multivariate time series databases using a moving window approach [Journal]. - [s.l.] : Industrial and Engineering Chemistry Research, 2002a. - Vol. 41. - pp. 3822–3838.

Singhal A., Seborg, D. Pattern matching in historical batch data using PCA [Journal]. - [s.l.] : IEEE Control Syst. Mag., 2002b. - 5 : Vol. 22. - pp. 53–63.

Smola A. J. Bartlett, P. L., Sch¨olkopf, B. and Schuurmans, D. Advances in Large Margin Classifiers [Book]. - Cambridge, MA : MIT Press, 2000.

Sreedharan P. and Haves, P. Comparison of chiller models for use in model-based fault detection. [Conference] // International Conference for Enhanced Building Operations (ICEBO). - Austin, TX : organized by Texas A&M University, 2001.

Stallard L.A. Model based expert system for failure detection and identification of household refrigerators. - Purdue, Indiana : Master's thesis, School of Mechanical Engineering, Purdue University, 2005.

Stylianou M. and Nikanpour, D. Performance monitoring, fault detection, and diagnosis of reciprocating chillers [Journal]. - [s.l.] : ASHRAE Transactions, 2006. - 1 : Vol. 102. - pp. 615-627.

Stylianou M. Classification functions to chiller fault detection and diagnosis [Journal]. - [s.l.] : ASHRAE Transactions, 2007. - 1 : Vol. 103. - pp. 645-648.

Syed N.A., Liu, H. and Sung, K.K. Incremental Learning with Support Vector Machines [Conference]. - [s.l.] : in Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI-99), 2009.

Tarassenko L., Hayton, P. and Brady, M. Novelty detection for the identification of masses in mammograms [Conference] // Proc. of the Fourth International IEE Conference on Artificial Neural Networks. - 2005. - Vol. 409. - pp. 442–447.

Tarifa E. and Scenna, N. Fault diagnosis, directed graphs, and fuzzy logic [Journal]. - [s.l.] : Computers and Chemical Engineering, 2007. - Vol. 21. - pp. 649-654.

Tashtoush B., Molhim, M. and Al-Rousan, M. Dynamic model of an HVAC system for control analysis [Journal]. - [s.l.] : Energy, 2005. - 10 : Vol. 30. - pp. 1729-1745.

Tassou S.A. and Grace, I.N. Fault diagnosis and refrigerant leak detection in vapour compression refrigeration systems [Journal]. - [s.l.] : International Journal of Refrigeration, 2005. - Vol. 28. - pp. 680-688.

Tax D.M.J. One-Class Classification; Concept Learning in the Absence of Counter- Examples. - Delft : Ph.D. thesis, Delft University of Technology, 2001.

Trane B. A Business of American Standard Companies, Introduction to HVAC Systems [Book]. - [s.l.] : Literature Order Number TRG-TRC018-EN, 2004.

Tsutsui H. and Kamimura, K. Chiller condition monitoring using topological case-based modeling. [Journal]. - [s.l.] : ASHRAE Transactions, 2006. - 1 : Vol. 102. - pp. 641-648.

Vapnik V. statistical learninig theory [Book]. - [s.l.] : John Wesley and Songd, 1998.

Venkatasubramanian V. and Rengaswamy, R. A review of process fault detection and diagnosis Part III: process history based methods [Journal]. - [s.l.] : Computers and Chemical Engineering, 2003. - Vol. 27. - pp. 327-346.

Venkatasubramanian V., Rengaswamy, R., Yin, K. and Kavuri, S. N. A review of process fault detection and diagnosis, Part I: Quantitative model-based methods [Journal] // Computers in Chemical Engineering. - 2003a. - Vol. 27. - pp. 293-311.

Venkatasubramanian V., Rengaswamy, R., Yin, K. and Kavuri, S. N. A review of process fault detection and diagnosis, Part III: Process history based methods [Journal] // Computers in Chemical Engineering. - 2006c. - Vol. 27. - pp. 327-346.

Venkatasubramanian V., Rengaswamy, R., Yin, K. and Kavuri, S. N. A review ofprocess fault detection and diagnosis, Part II: Qualitative models and search strategies [Journal] // Computers in Chemical Engineering. - 2006b. - Vol. 27. - pp. 313-326.

Viswanadham N. and Srichander, R. Fault detection using unknown-input observers [Journal]. - [s.l.] : Control Theory and Advanced Technology, 2008. - Vol. 3. - pp. 91-101.

Wagner J. and Shoureshi, R. Failure detection diagnostics for thermofluid systems [Journal]. - [s.l.] : Journal of Dynamic Systems, Measurement, and Control, 2002. - 4 : Vol. 114. - pp. 699-706.

Wang S.W. and Xiao, F. Detection and diagnosis of AHU sensor faults using principle component analysis method [Journal]. - [s.l.] : Energy Conversion and Management, 2004. - Vol. 45. - pp. 2667-2686.

Watanabe K., Hirota, S. and Himmelblau, D. M. Diagnosis of multiple simultaneous fault via hierarchical artificial neural networks [Journal]. - [s.l.] : . American Institute of Chemical Engineers Journal, 2004. - 5 : Vol. 40. - pp. 839-848.

Wise B.M and Gallagher, N. B. PLS_Toolbox Version 4.0 for use with MATLABTM [Book]. - Manson, WA : Eigenvector Research, 2006.

Witten I.H. and Frank, E. Data Mining: Practical Machine Learning Tools and Technique with Java Implementations [Book]. - [s.l.] : Morgan Kaufman Publishers, 2000.

Yairi T., Kato, Y. and Hori, K. Fault Detection by Mining Association Rules from House-keeping Data [Conference]. - [s.l.] : Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space (ISAIRAS), 2001.

Yoon S. and MacGregor, F. Fault diagnosis with multivariate statistical models part I: using steady state fault signatures [Journal]. - [s.l.] : Journal of Process Control, 2011. - 4 : Vol. 11. - pp. 387-400.

Yoshida H. and Kumar, S. ARX and AFMM model-based on-line real-time data base diagnosis of sudden fault in AHU of VAV system [Journal]. - [s.l.] : Energy Conversion and Management, 1999. - Vol. 40. - pp. 1191-1206.

Yoshida H., Kumar, S. and Morita, Y. Online fault detection and diagnosis in VAV air handling unit by RARX modeling [Journal]. - [s.l.] : Energy and Buildings, 2001. - Vol. 33. - pp. 391-401.

Yoshimura M. and Ito, N. Effective diagnosis methods for air-conditioning equipment in telecommunications buildings [Conference] // INTELEC 89: The Eleventh International Telecommunications Energy Conference. - Centro dei, Firenze : [s.n.], 1999. - Vol. 21. - pp. 1-7.

Yoshimura M., Ito, N. Effective diagnosis methods for air conditioning equipment in telecommunication buildings [Conference] // Eleventh International Telecommunications Energy Conference. - Florence, Italy : [s.n.], 1989. - Vol. 21/1. - pp. 1-7.

Zhang K., Jiang, B. and Shi, P. A New Approach to Observer-Based Fault-Tolerant Controller Design for Takagi-Sugeno Fuzzy Systems with State Delay [Journal]. - [s.l.] : Circuits Systems and Signal Processing, 2009. - 5 : Vol. 28. - pp. 679-697.

Zhang Y. M. and Li, X. R. Detection and diagnosis of sensor and actuator failures using IMM estimator [Journal]. - [s.l.] : AES, 2008. - 4 : Vol. 34. - pp. 1293-1312.

Zhou Q., Wang, S.W. and Ma, Z.J. A model-based fault detection and diagnosis strategy for HVAC systems [Journal]. - [s.l.] : International Journal of Energy Research, 2009. - 10 : Vol. 33. - pp. 903-918.

Appendices

Appendix A: MATLAB Code

HVAC Simulation code based on mathematical modelling

```
%HVAC Healthy Model (Mathematical)

function dx = HVACmodel(x)

%Parameters value
f_mix=960/3600;
V_ahu=2;
r=4;
%T_o=30;
H_fg=30000;    %H_fg=u+pv=29000 for 7C and 50000 for 12C
K_wv=0.62198*10^-5;
C_p=1000;
P_o=101325;
h_cc=30;    %Forced liquid (flowing) water - Free Convection Gas : 10
- 40 W/m2K
A_cc_air=5*1.5;
rho=1.15;
l=H_fg*K_wv/C_p;
P_Tcc=1200+101325;
V_a=5*5*3;
%Q_load=1200;
h_w=1.2;
A_w=4*(5*3);
C_cc=800;
h_water=40;
A_cc_w=4.5*1.5;
C_water=4200;
T_water_in=7;
C_w=850;
h_o=1;

f_cc=x(7);  %control signal

Q_load=x(8);
T_o=x(9);

%State Space equations
%x1 == Ts; x2 == Ta; x3 == Tcc; x4 == Tw; x5 == Ps; x6 == Pa;
%dx(7)=T_water_out
dx = zeros(7,1);
dx(1) = ((f_mix/V_ahu)*(((T_o/r)+((r-1)/r)*x(2))-
x(1)))+((f_mix*H_fg*K_wv/(C_p*V_ahu))*(((P_o/r)+((r-1)/r)*x(6))-
x(5)))+((h_cc*A_cc_air/(rho*C_p*V_ahu))*(x(3)-
x(1)))+((l*h_cc*A_cc_air/(rho*C_p*V_ahu))*(min(P_Tcc-x(5),0)));
dx(2) = ((f_mix/V_a)*(x(1)-x(2)))+((f_mix*H_fg*K_wv/(C_p*V_a))*(x(5)-
x(6)))+(Q_load/(rho*C_p*V_a))+((h_w*A_w*(x(4)-x(2)))/(rho*C_p*V_a));
dx(3) = -((h_cc*A_cc_air*(x(3)-x(1)))/C_cc)-
((l*h_cc*A_cc_air*(min((P_Tcc-
```

```
x(5)),0)))/C_cc)+((2*f_cc*h_water*A_cc_w*C_water*(T_water_in-
x(3)))/(C_cc*(2*f_cc^(1/3)*C_water+h_water*A_cc_w)));
dx(4) = -((h_w*A_w*(x(4)-x(2)))/C_w)-((h_o*A_w*(x(4)-T_o))/C_w);
dx(5) = ((l*h_cc*A_cc_air*(min((P_Tcc-
x(5)),0)))/(rho*H_fg*V_ahu*K_wv))+((f_mix/V_ahu)*(((P_o/r)+(r-
1)*x(6)/r)-x(5)));
dx(6) = ((f_mix/V_a)*(x(5)-x(6)));
dx(7) = T_water_in -((((2*f_cc*h_water*A_cc_w*C_water*(T_water_in-
x(3)))/(1*(2*f_cc^(1/3)*C_water+h_water*A_cc_w)))/(f_cc*C_water+0.0000
1));
```

```
%HVAC Faulty Model (Mathematical)

function dx = HVACFAULTmodel(x)

%Parameters value
f_mix=960/3600;
V_ahu=2;
                   %r=4;
                 %T_o=30;
H_fg=30000;    %H_fg=u+pv=29000 for 7C and 50000 for 12C
K_wv=0.62198*10^-5;
C_p=1000;
P_o=101325;
h_cc=30;     %Forced liquid (flowing) water - Free Convection Gas : 10
- 40 W/m2K
A_cc_air=5*1.5;
rho=1.15;
l=H_fg*K_wv/C_p;
P_Tcc=1200+101325;
V_a=5*5*3;
                    %Q_load=1200;
h_w=1.2;
A_w=4*(5*3);
C_cc=800;
h_water=40;
A_cc_w=4.5*1.5;
C_water=4200;
T_water_in=7;
C_w=850;
h_o=1;

Q_load=x(8);
T_o=x(9);
P_s0=x(10);    %supply fan speed fault  in fault free situation is 0
range of fault(0-10)
r=x(11);     %return damper fault   in fault free situation is 4
range of fault(4-0.1)
alpha=x(12);  %colling water pipe fault   in fault free situation is 1
range of fault(1-0)
f_cc=x(7)*alpha;  %control signal

  %State Space equations
%x1 == Ts; x2 == Ta; x3 == Tcc; x4 == Tw; x5 == Ps; x6 == Pa;
%dx(7)=T_water_out
dx = zeros(7,1);
```

```
dx(1) = ((f_mix/V_ahu)*(((T_o/r)+((r-1)/r)*x(2))-
x(1)))+((f_mix*H_fg*K_wv/(C_p*V_ahu))*(((P_o/r)+((r-1)/r)*x(6))-
x(5)))+((h_cc*A_cc_air/(rho*C_p*V_ahu))*(x(3)-
x(1)))+((l*h_cc*A_cc_air/(rho*C_p*V_ahu))*(min(P_Tcc-x(5),0)));
dx(2) = ((f_mix/V_a)*(x(1)-x(2)))+((f_mix*H_fg*K_wv/(C_p*V_a))*(x(5)-
x(6)))+(Q_load/(rho*C_p*V_a))+((h_w*A_w*(x(4)-x(2)))/(rho*C_p*V_a));
dx(3) = -((h_cc*A_cc_air*(x(3)-x(1)))/C_cc)-
((l*h_cc*A_cc_air*(min((P_Tcc-
x(5)),0)))/C_cc)+((2*f_cc*h_water*A_cc_w*C_water*(T_water_in-
x(3)))/(C_cc*(2*f_cc^(1/3)*C_water+h_water*A_cc_w)));
dx(4) = -((h_w*A_w*(x(4)-x(2)))/C_w)-((h_o*A_w*(x(4)-T_o))/C_w);
dx(5) = ((l*h_cc*A_cc_air*(min((P_Tcc-
x(5)),0)))/(rho*H_fg*V_ahu*K_wv))+((f_mix/V_ahu)*(((P_o/r)+(r-
1)*x(6)/r)-x(5)));
dx(6) = ((f_mix/V_a)*(x(5)-x(6)))-P_s0;
dx(7) = T_water_in-(((2*f_cc*h_water*A_cc_w*C_water*(T_water_in-
x(3)))/(1*(2*f_cc^(1/3)*C_water+h_water*A_cc_w)))/(f_cc*C_water+0.0000
1));
```

Online incremental SVM training code for two class

```
function [a, b, g, inds, inde, indw] = svcm_train(x, y, C);
% function [a, b, g, inds, inde, indw] = svcm_train(x, y, C);
%        support vector classification machine
%        incremental learning, and leave-one-out cross-validation
%        soft margin
%        uses "kernel.m"
%
%        x: independent variable, (L,N) with L: number of points; N:
dimension
%        y: dependent variable, (L,1) containing class labels (-1 or
+1)
%        C: soft-margin regularization constant
%
%        a: alpha coefficients (to be multiplied by y)
%        b: offset coefficient
%        g: derivatives (adding one yields margins for each point)
%        inds: indices of support vectors
%        inde: indices of error vectors
%        indw: indices of wrongly classified leave-one-out vectors

%%%%%%%%%% version 1.11; last revised 06/07/2002; send comments to
gert@jhu.edu %%%%%%%%%%

%%% GLOBAL VARIABLES:
global doloo online query terse verbose debug memoryhog visualize eps
if isempty(doloo)
    doloo = 1;                          % perform loo at end of training
end
if isempty(online)
    online = 0;                         % take data in the order it is
presented
end
if isempty(query)
    query = 0;                          % choose next point based on
margin distribution
end
if isempty(terse)
    terse = 0;                          % print out only final results
end
if isempty(verbose)
    verbose = 0;                        % print out details of
intermediate results
end
if isempty(debug)
    debug = 0;                          % use only for debugging; slows
it down significantly
end
if isempty(memoryhog)
    memoryhog = 0;                      % use more memory; good only if
kernel dominates computation
end
if isempty(visualize)
    visualize = 0;                      % record and plot trajectory of
coeffs. a, g, and leave-one-out g
end
if isempty(eps)
    eps = 1e-6;                         % margin "margin"; makes Q
strictly positive definite
end
```

```
%%% END GLOBAL VARIABLES

[L,N] = size(x);
[Ly,Ny] = size(y);
if Ly~=L
    fprintf('svcm_train error: x and y different number of data points
(%g/%g)\n\n', L, Ly);
    return
elseif Ny~=1
    fprintf('svcm_train error: y not a single variable (%g)\n\n', Ny);
    return
elseif any(y~=-1&y~=1)
    fprintf('svcm_train error: y takes values different from {-
1,+1}\n\n');
    return
end

eps2 = 2*eps/C;
tol = 1e-6;         % tolerance on derivatives at convergence, and
their recursive computation

fprintf('Support vector soft-margin classifier with incremental
learning\n')
fprintf('  %g training points\n', L)
fprintf('  %g dimensions\n\n', N)

keepe = debug|memoryhog;        % store Qe for error vectors as
"kernel cache"
keepr = debug|memoryhog;        % store Qr for recycled support
vectors as "kernel cache"
if verbose
    terse = 0;
    if debug
        fprintf('debugging mode (slower, more memory intensive)\n\n')
    elseif memoryhog
        fprintf('memoryhog active (fewer kernel evaluations, more
memory intensive)\n\n')
    end
    fprintf('kernel used:\n')
    help kernel
    fprintf('\n')
end

a = zeros(L,1);                      % coefficients, sparse
b = 0;                               % offset
W = 0;                               % energy function
g = -(1+eps)*ones(L,1);              % derivative of energy function

inds = [];                         % indices of support vectors; none
initially
inde = [];                         % indices of error vectors; none
initially
indo = (L:-1:1)';                  % indices of other vectors; all
initially
indr = [];                         % indices of "recycled" other
vectors; for memory "caching"
indl = [];                         % indices of leave-one-out vectors
(still to be) considered
```

```
indw = [];                          % indices of wrongly classified
leave-one-out vectors
ls = length(inds);                  % number of support vectors;
le = length(inde);                  % number of error vectors;
la = ls+le;                         % both
lo = length(indo);                  % number of other vectors;
lr = length(indr);                  % number of recycled vectors
lw = length(indw);                  % number or wrongly classified
leave-one-out vectors
processed = zeros(L,1);             % keeps track of which points have
been processed
R = Inf;                            % inverse hessian (a(inds) and b
only)
Qs = y';                            % extended hessian; (a(inds) plus b,
and all vectors)
Qe = [];                            % same, for inde ("cache" for Qs)
Qr = [];                            % same, for indr ("cache" for Qs)
Qc = [];                            % same, for indc (used for gamma and
Qs)
if visualize                        % for visualization
    figure(1)
    hold off
    clf
    axis([-0.1*C, 1.1*C, -1.2, 0.2])
    gctraj = [];
    figure(2)
    hold off
    clf
    figure(3)
    hold off
    clf
end

iter = 0;                           % iteration count
memcount = 0;                       % memory usage
kernelcount = 0;                    % kernel computations, counted one
"row" (epoch) at a time
training = 1;                       % first do training recursion ...
leaveoneout = 0;                    % ... then do leave-one-out sequence
(with retraining)
indc = 0;                           % candidate vector
indco = 0;                          % leave-one-out vector
indso = 0;                          % a recycled support vector; used as
buffer
free = a(indo)>0|g(indo)<0;         % free, candidate support or error
vector
left = indo(free);                  % candidates left
continued = any(left);
while continued                      % check for remaining free points
or leave-one-outs to process

    % select candidate indc
    indc_prev = indc;
    if online & indc_prev>0
        if query
            processed(indc_prev) = 1;      % record last point in the
history log
        else
            processed(1:indc_prev) = 1;    % record last and all
preceding points
        end
```

```
    end
    if query
%        [gindc, indc] = max(g(left));      % closest to the margin
         [gindc, indc] = min(g(left));      % greedy; worst margin
         indc = left(indc);
    else
         indc = left(length(left));         % take top of the stack,
"last-in, first-out"
    end

    % get Qc, row of hessian corresponding to indc (needed for gamma)
    if keepr & lr>0 & ...               % check for match among recycled
vectors
         any(find(indr==indc))
             ir = find(indr==indc);                 % found, reuse
             Qc = Qr(ir,:);
             indr = indr([1:ir-1,ir+1:lr]);         % ... remove from
indr
             Qr = Qr([1:ir-1,ir+1:lr],:);           % ... and Qr
             lr = lr-1;
    elseif indc==indso                  % support vector from previous
iteration, leftover in memory
         Qc = Qso;
    elseif ls>0 & ...                   % check for match among support
vectors
         any(find(inds==indc))
             is = find(inds==indc);                 % found, reuse
             Qc = Qs(is+1,:);
    elseif keepe & le>0 & ...           % check for match among stored error
vectors
         any(find(inde==indc))
             ie = find(inde==indc);                 % found, reuse
             Qc = Qe(ie,:);
    elseif indc~=indc_prev              % not (or no longer) available,
compute
         xc = x(indc,:);
         yc = y(indc);
         Qc = (yc*y').*kernel(xc,x);
         Qc(indc) = Qc(indc)+eps2;
         kernelcount = kernelcount+1;
    end

    % prepare to increment/decrement z = a(indc)' or y(indc)*b,
subject to constraints.
    % move z up when adding indc ((re-)training), down when removing
indc (leave-one-out or g>0)
    upc = ~leaveoneout & (g(indc)<=0);
    polc = 2*upc-1;                     % polarity of increment in z
    beta = -R*Qs(:,indc);               % change in [b;a(inds)] per change
in a(indc)
    if ls>0
         % move z = a(indc)'
         gamma = Qc'+Qs'*beta;          % change in g(:) per change in z =
a(indc)'
         z0 = a(indc);                  % initial z value
         zlim = max(0,C*polc);          % constraint on a(indc)
    else % ls==0
         % move z = y(indc)*b and keep a(indc) constant; there is no
a(:) free to move in inds!
         gamma = y(indc)*Qs';           % change in g(:) per change in z =
y(indc)*b
```

```
        z0 = y(indc)*b;             % initial z value
        zlim = polc*Inf;            % no constraint on b
    end
    gammac = gamma(indc);
    if gammac<=-tol
        fprintf('\nsvcm_train error: gamma(indc) = %g <= 0 (Q not
positive definite)\n\n', gammac)
    elseif gammac==Inf
        fprintf('\nsvcm_train error: gamma(indc) = %g (Q rank
deficient)\n\n', gammac)
        return
    end

    % intrinsic limit: g(indc) = 0, where indc becomes support vector
    if ~leaveoneout                 % only consider when training indc,
not when removing indc
        zlimc = z0-g(indc)'./gammac;
    else                            % leave-indc-out!
        zlimc = polc*Inf;
    end

    % support vector constraints: 0<=a(inds)<=C
    zlims = Inf*polc;                              % by default,
immaterial
    if ls>0
        is = find(inds==indc);
        if any(is)                                 % leave-indc-out,
remove from inds
            zlims = z0;                            % clamp z; no change
to variables
        else
            betaa = beta(2:ls+1);                  % beta terms
corresponding to a(inds)  (not b)
            void = (betaa==0);                     % void zero betaa
values ...
            if any(any(~void))
                warning off % suppress div. by 0
                zlims = z0+(C*(betaa*polc>0)-a(inds))./betaa;
                warning on
                zlims(void) = polc*Inf;            % ... which don't
enter the constraints
                [zmins, is] = min(zlims*polc,[],1);
                imin = find(zlims==zmins);
                if length(imin)>1
                    [gmax, imax] = max(abs(betaa(imin)),[],1);
                    is = imin(imax);
                end
                zlims = zmins*polc;                % pick tightest
constraint
            end
        end
    end

    % error vector constraints: g(inde)<=0
    zlime = Inf*polc;                              % by default,
immaterial
    if le>0
        ie = find(inde==indc);
        if any(ie)                                 % leave-indc-out,
remove from inde
```

```
            zlime = z0;                          % clamp z; no change
to variables
        else
            gammae = gamma(inde);
            void = (gammae*polc<0)|(gammae==0);  % void g moving down,
or zero gamma...
            if any(any(~void))
                warning off % suppress div. by 0
                zlime = z0-g(inde)./gammae;
                warning on
                zlime(void) = polc*Inf;          % ... which don't
enter the constraints
                [zmine, ie] = min(zlime*polc,[],1);
                imin = find(zlime==zmine);
                if length(imin)>1
                    [gmax, imax] = max(abs(gammae(imin)),[],1);
                    ie = imin(imax);
                end
                zlime = zmine*polc;              % pick tightest
constraint
            end
        end
    end

    % ordinary vector constraints: g(indo)>=0 (only for those that
already are)
    zlimo = Inf*polc;                            % by default,
immaterial
    if lo>0
        gammao = gamma(indo);
        void = (indo==indc)|(g(indo)<0)|(gammao*polc>0)|(gammao==0);
                                                 % void c, g negative,
g moving up, or zero gamma...
        if online
            void = void|~processed(indo);        % ... or, if online,
points not seen previously,...
        end
        if any(any(~void))
            warning off % suppress div. by 0
            zlimo = z0-g(indo)./gammao;
            warning on
            zlimo(void) = polc*Inf;              % ... which don't
enter the constraints
            [zmino, io] = min(zlimo*polc,[],1);
            imin = find(zlimo==zmino);
            if length(imin)>1
                [gmax, imax] = max(abs(gammao(imin)),[],1);
                io = imin(imax);
            end
            zlimo = zmino*polc;                  % pick tightest
constraint
        end
    end

    % find constraint-satisfying z
    [z,flag] = min([zlim;zlimc;zlims;zlime;zlimo]*polc);
    z = z*polc;
    if (z-z0)*polc<0
        fprintf('\nsvcm_train error: z-z0 of wrong polarity (Q not
positive definite)\n\n')
        return
```

```
    end

    if verbose & ~leaveoneout & abs(z-z0)<eps
        fprintf('%g*', flag)            % procrastinating iteration!
no progress made
    end

    % update a, b, g and W from z-z0
    if ls>0                             % z = a(indc)
        a(indc) = z;
        b = b+(z-z0)*beta(1);
        a(inds) = a(inds)+(z-z0)*beta(2:ls+1);
        W = W+(z-z0)*(g(indc)'+0.5*(z-z0)*gammac);                  %
energy
    else                                % z = y(indc)*b
        b = y(indc)*z;
    end
    g = g+(z-z0)*gamma;                 % update g
    iter = iter+1;
    if visualize & ~leaveoneout
        atraj(1:L,iter)=a;              % record trajectory of a(:)
over time
        gtraj(1:L,iter)=g;
        ctraj(iter)=indc;
    end

    % bookkeeping: move elements across indc, inds, inde and indo, and
update R and Qs
    converged = (flag<3);               % done with indc; no other
changes in inds/inde
    incl_inds = 0;
    if flag==1                          % a(indc) reaches the limits 0
or C, stop moving
        if upc                                  % a(indc)=C, add to
inde
            inde = [inde; indc];
            le = le+1;
            if keepe
                Qe = [Qe; Qc];
            end
            a(indc) = C;                        % should be OK, just
to avoid round-off
        else % ~upc                                % a(indc)=0, indc
stays in (or moves to) indo
            a(indc) = 0;                        % should be OK, just
to avoid round-off
        end
    elseif flag==2                      % add indc to support vectors
...
        incl_inds = 1;
        indb = indc;                            % ... store in buffer
indb for now
        Qb = Qc;
    elseif flag==3                      % one of support vectors
becomes error or other vector
        indso = inds(is);                       % outgoing inds
        Qso = Qs(is+1,:);                       % could be reused
later
        free_indc = (indc==indso);              % leave-indc-out: indc
is part of inds
```

```
        if beta(is+1)*polc<0 | free_indc        % a(indso)=0 or
indso=indc, move to indo
            if ~free_indc
                if keepr
                    indr = [indr; indso];       % also recycle into
indr for later use
                    Qr = [Qr; Qs(is+1,:)];
                    lr = lr+1;
                end
                if a(indso)>C/2
                    fprintf('svcm_train error: a(indso)=%g; 0
anticipated\n', a(indso));
                end
                a(indso) = 0;                   % should be OK, just
to avoid round-off
            end
            g(indso) = 0;                       % same
            indo = [indo; indso];
            lo = lo+1;
        else % beta(is+1)*polc>0 & ~free_indc   % a(indso)=C, move to
inde
            if a(indso)<C/2
                fprintf('svcm_train error: a(indso)=%g; C
anticipated\n', a(indso));
            end
            if keepe
                Qe = [Qe; Qs(is+1,:)];          % save to memory cache
            end
            a(indso) = C;                       % should be OK, just
to avoid round-off
            g(indso) = 0;                       % same
            inde = [inde; indso];
            le = le+1;
        end
        inds = inds([1:is-1,is+1:ls]);          % remove from inds
        stripped = [1:is,is+2:ls+1];            % also from Qs and R
...
        Qs = Qs(stripped,:);
        ls = ls-1;
        if ls > 0
            if R(is+1,is+1)==0
                fprintf('\nsvcm_train error: divide by zero in R
contraction\n')
                R(is+1,is+1)=1e-8;
            end
            R = R(stripped,stripped)-
R(stripped,is+1)*R(is+1,stripped)/R(is+1,is+1);
        else % no support vectors left
            R = Inf;
        end
    elseif flag==4                              % one of error vectors
becomes support/other vector
        indeo = inde(ie);                        % outgoing inde
        if indc==indeo                           % leave-indc-out
            indo = [indo; indeo];                % add inde(ie) to
other vectors
            lo = lo+1;
        else
            incl_inds = 1;                       % add inde(ie) to
support vectors ...
```

```
            indb = indeo;                        % ... store in buffer
indb for now
            if keepe
                Qb = Qe(ie,:);                   % recover from Qe
cache
            elseif indb==indso
                Qb = Qso;                        % recover from
previous outgoing support vector
            else                                 % not in memory
either way--- recompute
                Qb = (y(indb)*y').*kernel(x(indb,:),x);
                Qb(indb) = Qb(indb)+eps2;
                kernelcount = kernelcount+1;
            end
        end
        inde = inde([1:ie-1,ie+1:le]);           % remove from inde
        if keepe
            Qe = Qe([1:ie-1,ie+1:le],:);         % remove from Qe
        end
        le = le-1;
    elseif flag==5                       % one of other vectors
becomes support vector
        indoo = indo(io);                        % outgoing indo
        incl_inds = 1;                           % add indo(io) to
support vectors ...
        indb = indoo;                            % ... store in buffer
indb for now
        if keepr & lr>0 & any(find(indr==indb))  % check for match
among recycled vectors
            ir = find(indr==indb);                       % found,
reuse
            Qb = Qr(ir,:);
            indr = indr([1:ir-1,ir+1:lr]);               % ... remove
from indr
            Qr = Qr([1:ir-1,ir+1:lr],:);                 % ... and Qr
            lr = lr-1;
        elseif indb==indso
            Qb = Qso;                            % recover from
previous outgoing support vector
        else                                     % not in memory
either way--- recompute
            Qb = (y(indb)*y').*kernel(x(indb,:),x);
            Qb(indb) = Qb(indb)+eps2;
            kernelcount = kernelcount+1;
        end
        indo = indo([1:io-1,io+1:lo]);           % remove from indo
        lo = lo-1;
    end

    if incl_inds                        % move buffer indb into
support vectors inds
        inds = [inds; indb];                     % move to inds ...
        ls = ls+1;
        Qs = [Qs; Qb];                           % and also Qs and R
...
        if ls==1                                 % compute R directly
            R = [-Qb(indb), y(indb); y(indb), 0];
        else                                     % compute R
recursively
            if flag==2                           % from indc; use beta
and gamma
```

```
                    pivot = gamma(indb);
            else % flag==4                      % from inde; compute
beta and pivot
                beta=-R*Qs(1:ls,indb);
                pivot = [beta',1]*Qs(:,indb);
            end
            if pivot<eps2        % should be eps2 when kernel is
singular (e.g., linear)
                fprintf('\nsvcm_train error: pivot = %g < %g in R
expansion\n\n', pivot, eps2)
                pivot = eps2;
            end
            R =
[R,zeros(ls,1);zeros(1,ls+1)]+[beta;1]*[beta',1]/pivot;
        end
    end

    % minor correction in R to avoid numerical instability in
recursion when data is near-singular
    Qss = [[0;y(inds)],Qs(:,inds)];
    R = R+R'-R*Qss*R';

    % indc index adjustments (including leave-one-out)
    if converged & (upc|flag==2)                % indc is now part of
inds or inde
        i = find(indo==indc);
        indo = indo([1:i-1,i+1:lo]);  % remove indc from indo
        lo = lo-1;
    elseif keepr
        indr = [indr; indc];                    % recycle again into
indr for later use
        Qr = [Qr; Qc];
        lr = lr+1;
    end
    if leaveoneout
        indoc = indo(find(indo~=indc));         % indo other than indc
        free = a(indoc)>0|g(indoc)<0;           % candidate
support/error vectors in indoc
        if visualize & ~any(free)
            gctraj = [gctraj,[a(indc);g(indc)]];
        end
        converged = converged & ~any(free);
        if converged                            % leave-indc-out
reached a(indc)=0, all others settled
            indl = indl(find(indl~=indc));      % remove indc from
indl
            if visualize
                figure(1)
                hold on
                if any(gctraj)

plot(gctraj(1,:),gctraj(2,:),'ok',gctraj(1,:),gctraj(2,:),'-k')
                end
                gctraj = [];                     % cleanup for next
curve
            end
            if g(indc)<-1                       % if leave-indc-out
generates an error ...
                indw = [indw; indc];            % ... store its index
                lw = lw+1;                      % ... and increment
leave-one-out error count
```

```
            end
        end
    end

    % debugging mode: check for consistency; a, R, g and W
    if debug
        f = find(isnan(a));
        if any(f)
           fprintf('svcm_train error: a(%g) = %g\n', [f, a(f)]')
        end
        f = find(isnan(g));
        if any(f)
           fprintf('svcm_train error: g(%g) = %g\n', [f, g(f)]')
        end
        f = find(a>C);
        if any(f)
           fprintf('svcm_train error: a(%g) = %g > C\n', [f, a(f)]')
        end
        f = find(a<0);
        if any(f)
           fprintf('svcm_train error: a(%g) = %g < 0\n', [f, a(f)]')
        end
        f = inde(find(a(inde)>C|a(inde)<C));
        if any(f)
                fprintf('svcm_train error: a(%g) = %g ~= C\n', [f,
a(f)]')
        end
        if abs(y'*a)>tol
                fprintf('svcm_train error: y''*a = %g ~= 0
(tol=%g)\n', y'*a, tol);
        end
        Rdiv = max(max(abs(Qss*R-diag(ones(ls+1,1)))));
        if Rdiv>tol
            if flag==2|flag==4     % support vector added
                fprintf('svcm_train error: divergence %g in R
expansion (tol=%g; pivot=%g)\n',...
                         Rdiv, tol, pivot);
            elseif flag==3         % support vector removed
                fprintf('svcm_train error: divergence %g in R
contraction (tol=%g)\n', Rdiv, tol);
            else                   % no support vector added or
removed; strange...
                fprintf('svcm_train error: divergence %g in R ?
(tol=%g)\n', Rdiv, tol);
            end
        end
        if keepe&keepr             % only check g when Qe and Qr are
readily available
            greal = Qs'*[b;a(inds)]-(1+eps);
            if le>0
                greal = greal+Qe'*a(inde);
            end
            if lr>0
                greal = greal+Qr'*a(indr);
            end
            if max(abs(greal-g))>tol
                fprintf('svcm_train error: tolerance (tol=%g) exceeded
in computation of g\n', tol);
            end
        end
        f = inds(find(abs(g(inds))>tol));
```

```
        if any(f)
            fprintf('svcm_train error: g(%g) = %g ~= 0 (tol=%g)\n',
[f, g(f), f*0+tol]');
        end
        f = inde(find(g(inde)>tol));
        if any(f)
            fprintf('svcm_train error: g(%g) = %g > 0 (tol=%g)\n', [f,
g(f), f*0+tol]');
        end
        Wreal = 0.5*sum((g-b*y-1-eps).*a);        % energy
        if abs(Wreal-W)>tol*abs(W)
            fprintf('svcm_train error: energy W = %g ~= %g
(tol=%g)\n', W, Wreal, tol);
        end
        inda = sort([indo;inds;inde]);
        if any(inda~=(1:L)')
            fprintf('svcm_train error: union [indo;inds;inde] does not
equate entire set\n');
        end
        if ls~=length(inds)
            fprintf('svcm_train error: miscount in number of support
vectors\n');
        end
        if le~=length(inde)
            fprintf('svcm_train error: miscount in number of error
vectors\n');
        end
        if lo~=length(indo)
            fprintf('svcm_train error: miscount in number of other
vectors\n');
        end
    end

    memcount = max(memcount,ls+keepe*le+keepr*lr+~(keepe&keepr));
% kernel storage

    if verbose
        fprintf('    c: %g (#%g)', y(indc)>0, indc)
        if leaveoneout
            fprintf(', margin: %6.3g, gamma: %6.3g', g(indc)+1,
gammac)
        end
        fprintf(' | s: ')
        fprintf('%g', y(inds)>0)
        if ls<6
            fprintf(' (')
            fprintf('#%g', inds)
            fprintf(')')
        end
        fprintf(' | e: 0:%g, 1:%g\n', sum(y(inde)<0), sum(y(inde)>0))
    end

    if converged                          % indc finished; report
        if ~training & ~leaveoneout       % retraining or tracing back;
uninteresting
            % nothing
        elseif ~terse & ~leaveoneout
            fprintf(' [%3g%%] iter. %3g: %3g support vectors; %3g
error vectors;  energy %g\n', ...
                100-round(100*length(left)/L), iter, ls, le, W)
        elseif ~terse % & leaveoneout
```

```
            fprintf(' [%3g%%] iter. %3g: %3g leave-one-out errors',
100-round(100*length(indl)/L), iter, lw)
            fprintf(' (# %g: margin %g)\n', indc, g(indc)+1)
        elseif ~leaveoneout % & terse
            if ls+le>la
                fprintf('+')
            elseif ls+le<la
                fprintf('-')
            else
                fprintf('=')
            end
        else % leaveoneout & terse
            if g(indc)<-1
                fprintf('x')
            else
                fprintf('o')
            end
        end
    la = ls+le;
    end

    % prepare for next iteration, if any
    indoc = indo(find(indo~=indco));         % indo other than indco
(leave-one-out index, if active)
    free = a(indoc)>0|g(indoc)<0;            % candidate support/error
vectors in indoc
    if any(free)
        left = indoc(free);                  % candidates left, keep
(re-)training
        leaveoneout = 0;                     % interrupt leave-one-out,
if active
    else % ~any(free)                        % done; finish up and (re-
)initiate leave-one-out
        if training                          % first time around (not
re-training)
            % print out results of svm training
            if terse
                fprintf('\n\n  %g support vectors; %g error vectors;
energy %g\n\n', ...
                                 ls, le, W)
            else
                fprintf('\n%4g epoch kernel evaluations (%3g%% of run-
time)\n',...
                        kernelcount, round(kernelcount/(ls+le)*100))
                fprintf(  '%4g epoch vectors in memory  (%3g%% of
data)\n\n',...
                        memcount, round(memcount/(N+1)*100))
            end
            if visualize

                % plot a trajectory
                figure(2)
                h=image(atraj/C*length(gray));
                h2=get(h,'Parent');
                set(h2,'YDir','normal')      % 'image' normally reverts
the y axis
                colormap(1-gray)
                xlabel('Iteration')
                ylabel('Coefficients \alpha_{\it{i}}')
                print -deps atraj.eps
```

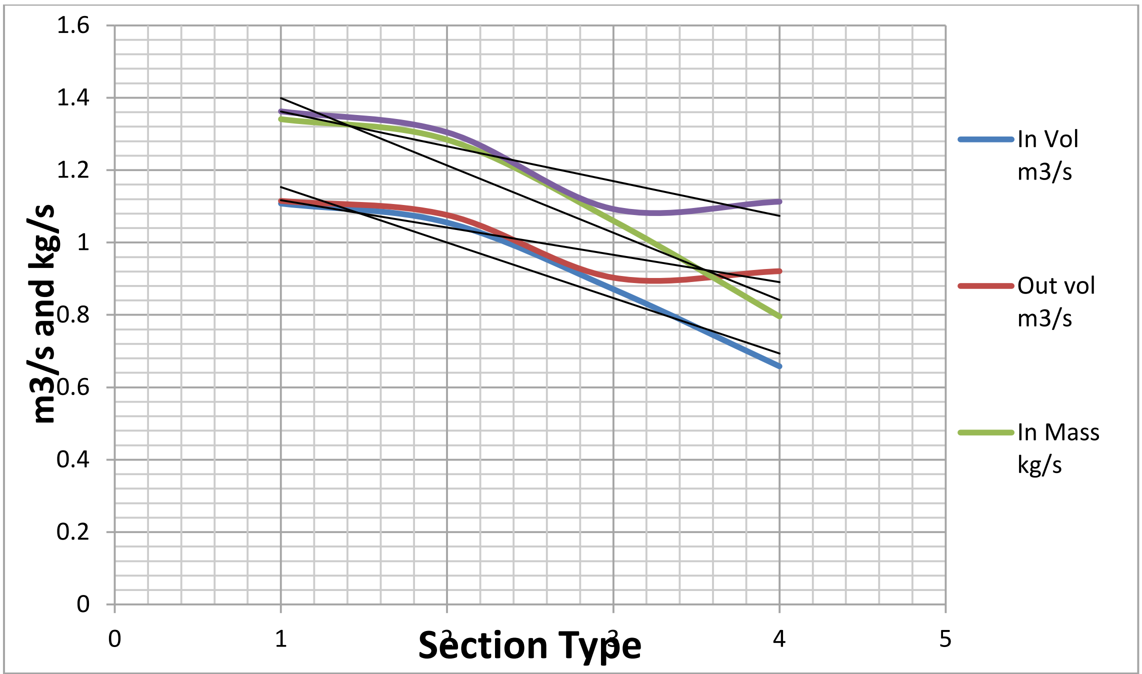
```
                % plot g trajectory
                figure(3)
                gmax = max(max(abs(gtraj)));
                h=image((gtraj/gmax+1)/2*length(gray));
                h2=get(h,'Parent');
                set(h2,'YDir','normal')      % 'image' normally reverts
the y axis
                grey = gray;
                redblue = min(grey,1-grey);
                redblue(:,1) = redblue(:,1)+1-grey(:,1);
                redblue(:,2) = 2*redblue(:,2);
                redblue(:,3) = redblue(:,3)+grey(:,3);
                colormap(redblue)
                xlabel('Iteration')
                ylabel('Coefficients {\it{g}_{\it{i}}}')
                print -deps gtraj.eps

                save traj atraj gtraj ctraj
            end
            if debug                         % store final result to
compare with retraining later
                afinal = a;
                gfinal = g;
                Wfinal = W;
            end

            % initiate leave-one-out, if desired
            if doloo
              leaveoneout = 1;
              indl = [inds;inde];            % support and error
vectors (others already correct)
              indw = indl(g(indl)<-1);       % leave-one-out errors so
far (g<-1) ...
              lw = length(indw);             % ... their number
              indl = indl(g(indl)>=-1);      % remove errors so far
from leave-one-out stack
              indco = indl(length(indl));    % pick first leave-one-out
index; top of stack ...
              left = indco;                  % ... and let indc=indco
(untrain; upc=0)
            else
              continued = 0;
            end

            training = 0;                    % don't ever visit again!
        elseif ~leaveoneout                  % retrained or traced back
            if debug&~any(find(indo==indco))
                % traced back; compare with previously trained results
                if max(abs(a-afinal))>tol
                        fprintf('svcm_train error: final coeffs. a
exceed tolerance (tol=%g)\n', tol);
                end
                if max(abs(g-gfinal))>tol
                        fprintf('svcm_train error: final derivatives g
exceed tolerance (tol=%g)\n', tol);
                end
                if abs(W-Wfinal)>tol*abs(W)
                        fprintf('svcm_train error: final energy W = %g
~= %g (tol=%g)\n', Wfinal, W, tol);
                end
            end
```

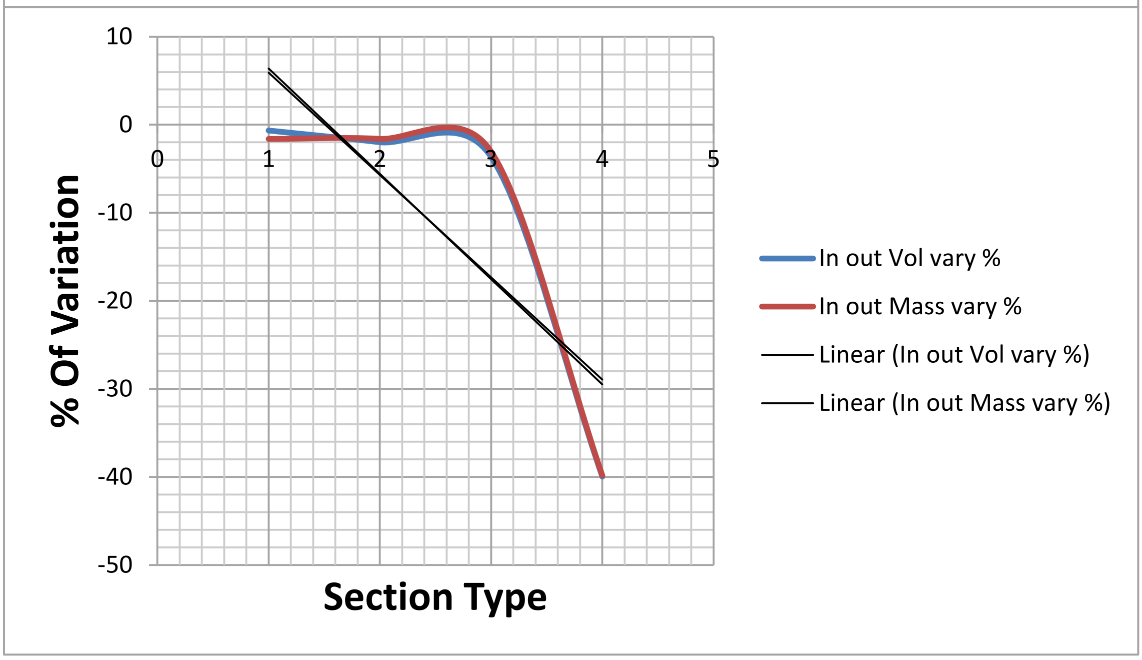
```
            if any(indl)
                indco = indl(length(indl)); % leave-one-out index; top
of stack
                left = indco;
                leaveoneout = 1;
            else % ~any(indl) % finished all leave-one-outs;
summarize, and done!
                continued = 0;
                ltw = sum(g<-1);              % number of training
errors
                if terse
                    fprintf('\n\n  %g leave-one-out errors;', lw)
                    fprintf(' %g training errors\n\n', ltw)
                else % ~terse
                    fprintf('\n%4g training points\n', L)
                    fprintf(  '%4g support/error vectors (%g/%g)\n',
la, ls, le)
                    fprintf(  '%4g leave-one-out errors   (%3.1f%%)\n',
lw, lw/L*100)
                    fprintf(  '%4g training errors        (%3.1f%%)\n',
ltw, ltw/L*100)
                    fprintf('\n%4g epoch kernel evaluations (%3g%% of
run-time)\n',...
                        kernelcount, round(kernelcount/(ls+le)*100))
                    fprintf(  '%4g epoch vectors in memory  (%3g%% of
data)\n\n',...
                        memcount, round(memcount/(N+1)*100))
                end
                if visualize                  % plot g trajectory
                    figure(1)
                    hold off
                    xlabel('\alpha_{\it{c}}')
                    ylabel('\it{g_c}')
                    axis([-0.1*C, 1.1*C, -1.2, 0.2])
                    h=line([0,0,C,C],[-1,0,0,-1]);
                    set(h,'Color',[0 0 0])
                    set(h,'LineStyle',':')
                    set(h,'LineWidth',[0.2])
                    h=line([0,C],[-1,-1]);
                    set(h,'Color',[0 0 0])
                    set(h,'LineStyle','--')
                    set(h,'LineWidth',[1.0])
                    print -deps gctraj.eps
                end
            end
            else % leaveoneout                % leave-one-out procedure
            if flag==1                        % finished leave-one-out,
now trace back
                % note: already decremented indl and updated indw/lw
above
                leaveoneout = 0;
                indco = 0;
            else                              % not done yet with leave-
one-out ...
                left = indco;                 % ... continue with
indc=indco
            end
        end
    end
end
```

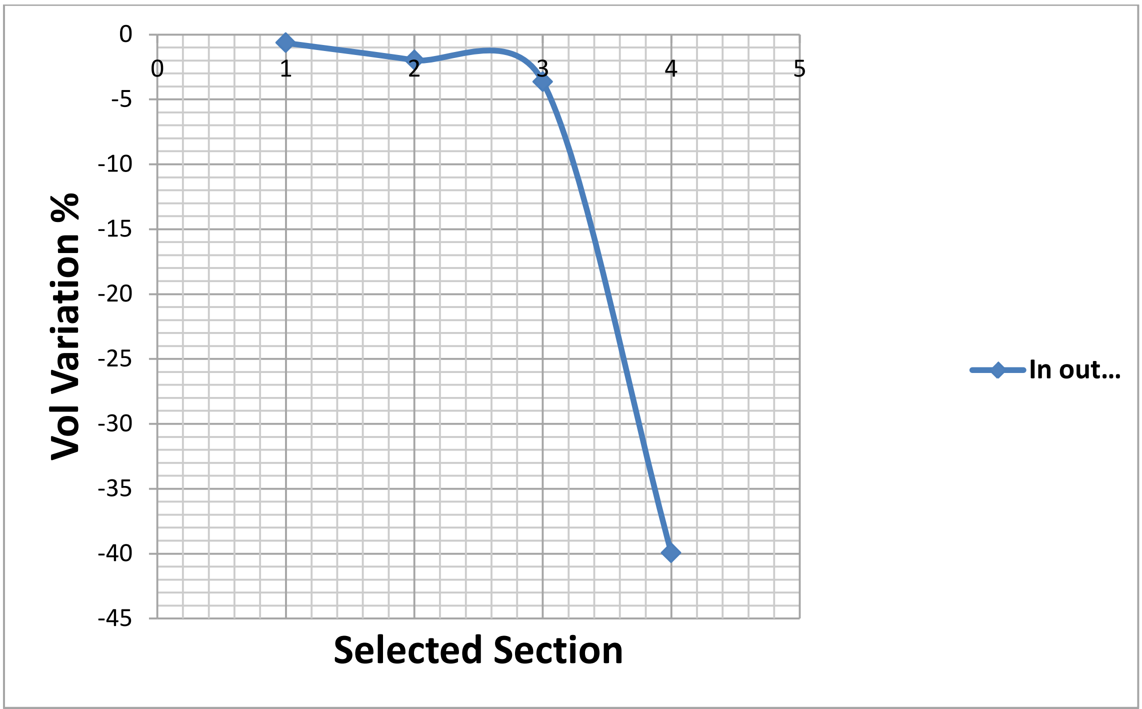
Appendix B: Experimental data

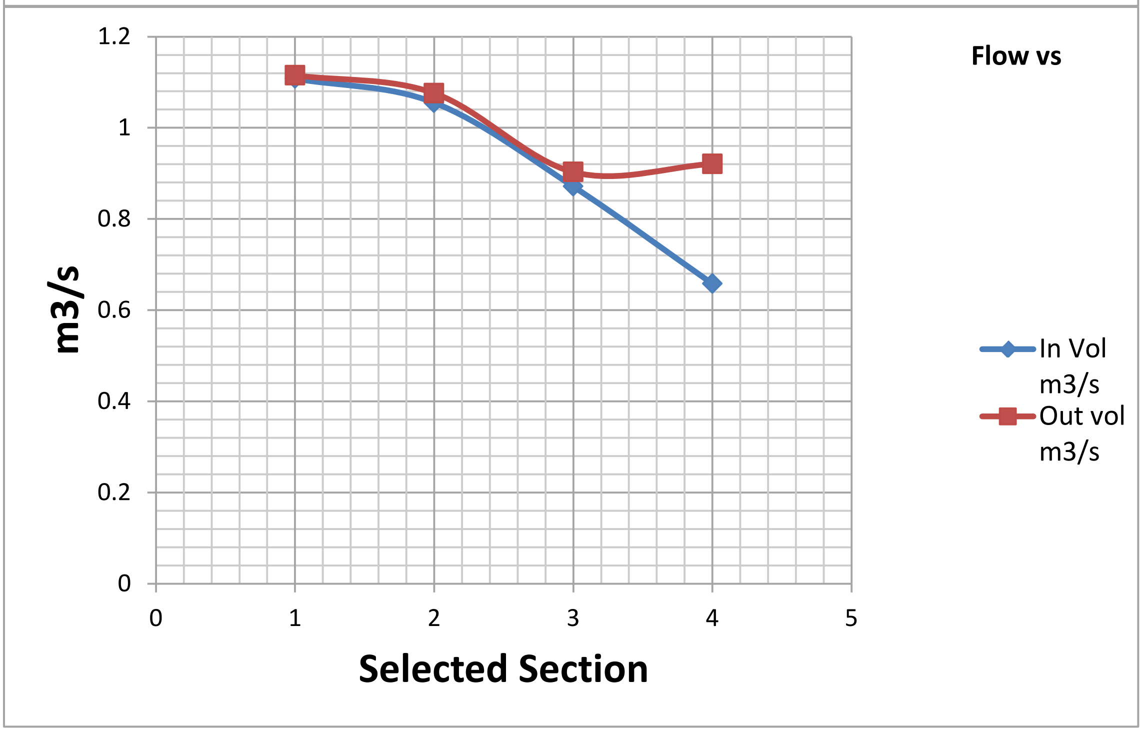
1. Supply Duct air performance

Faulty Situation

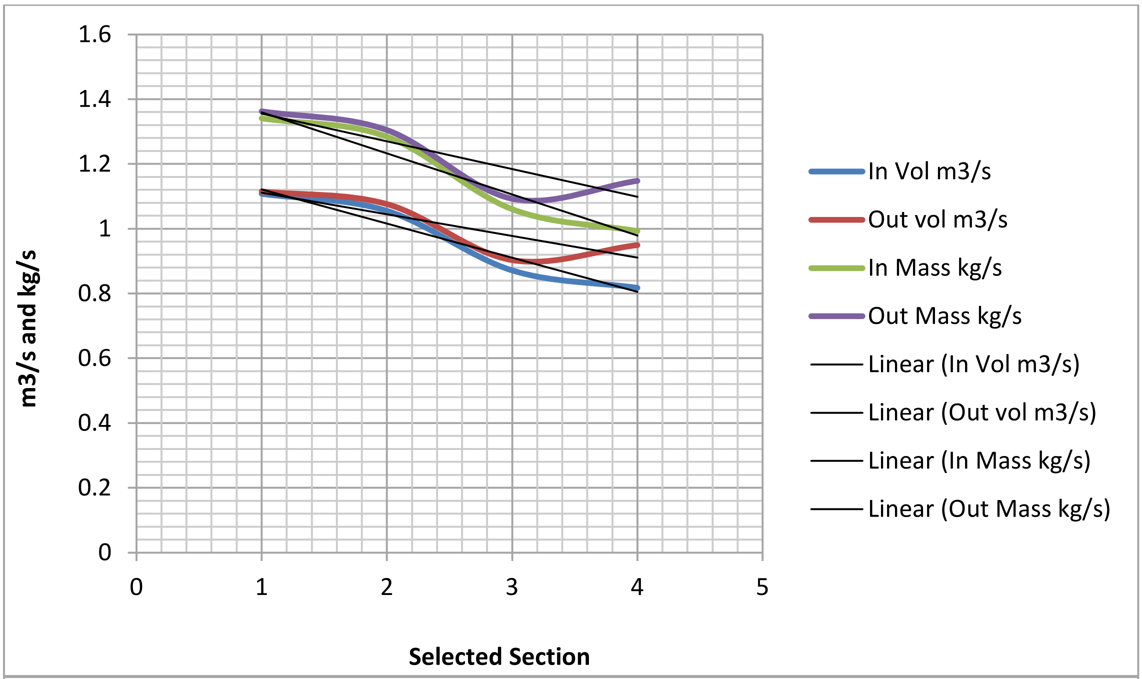


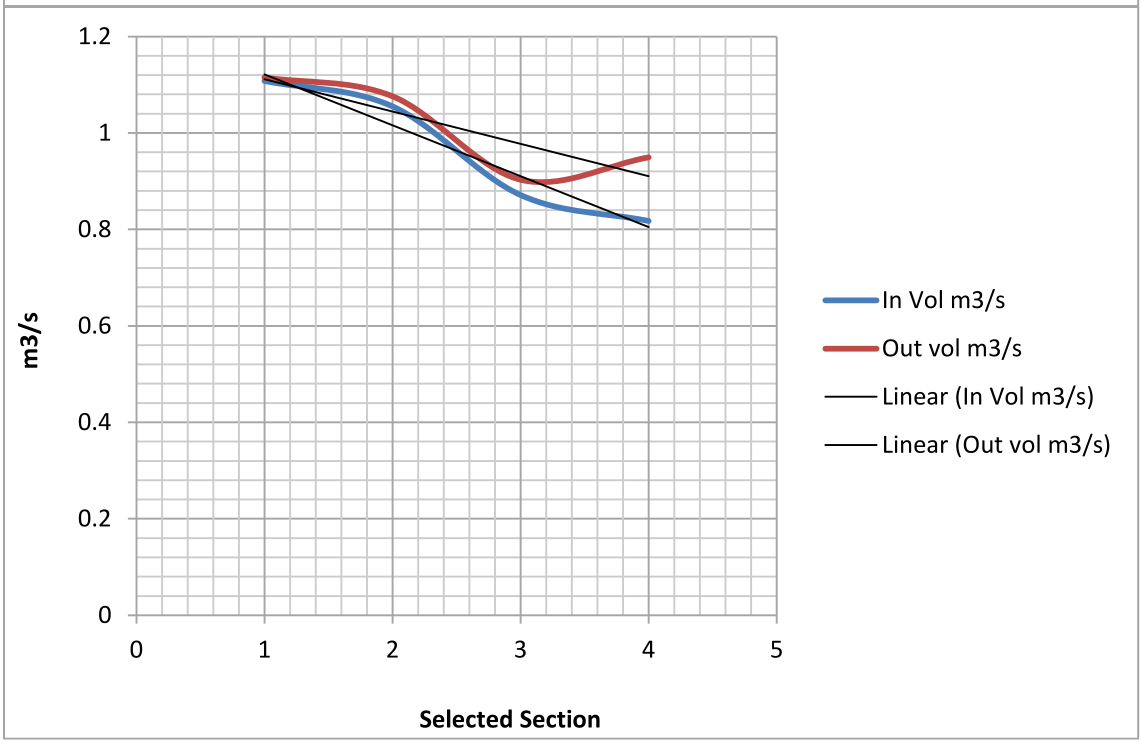


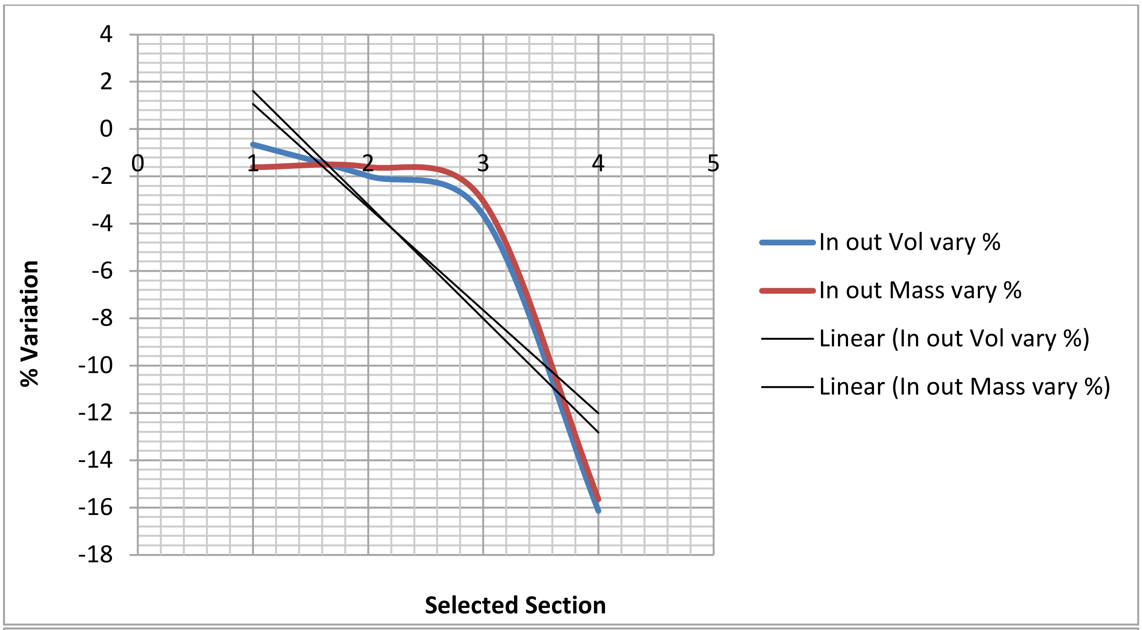

Vol Variation %
0
-5
-10
-15
-20
-25
-30
-35
-40
-45
0
1
2
3
4
5
Selected Section
In out...


Flow vs
m3/s
1.2
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
Selected Section
In Vol m3/s
Out vol m3/s

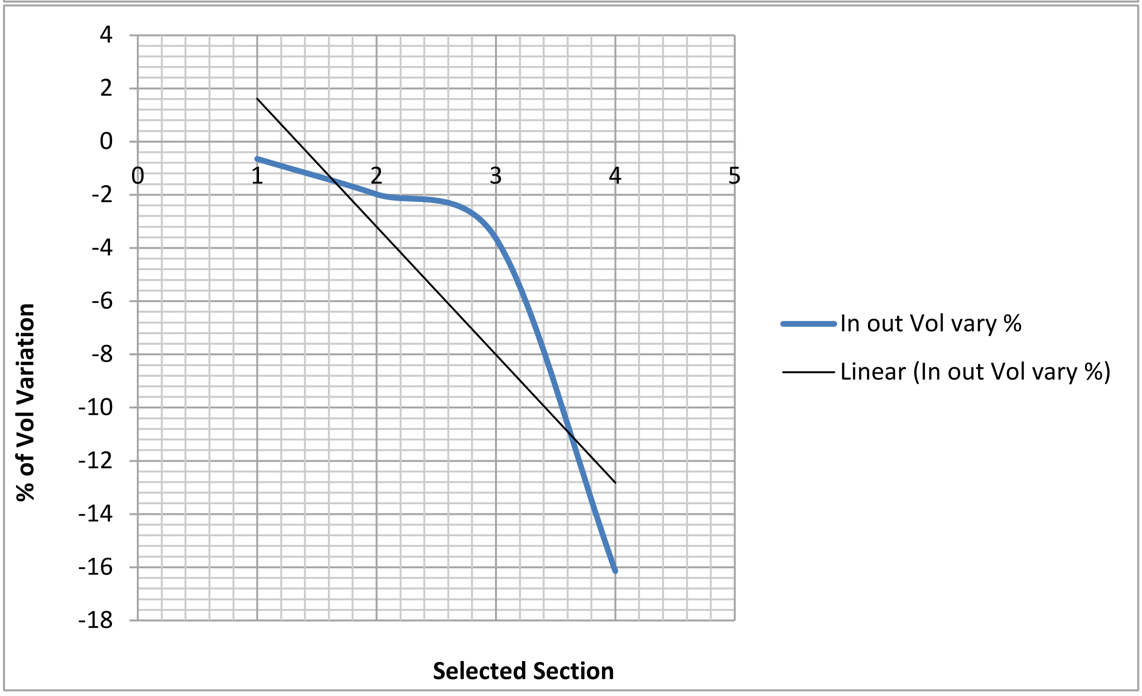
After Fault Removed





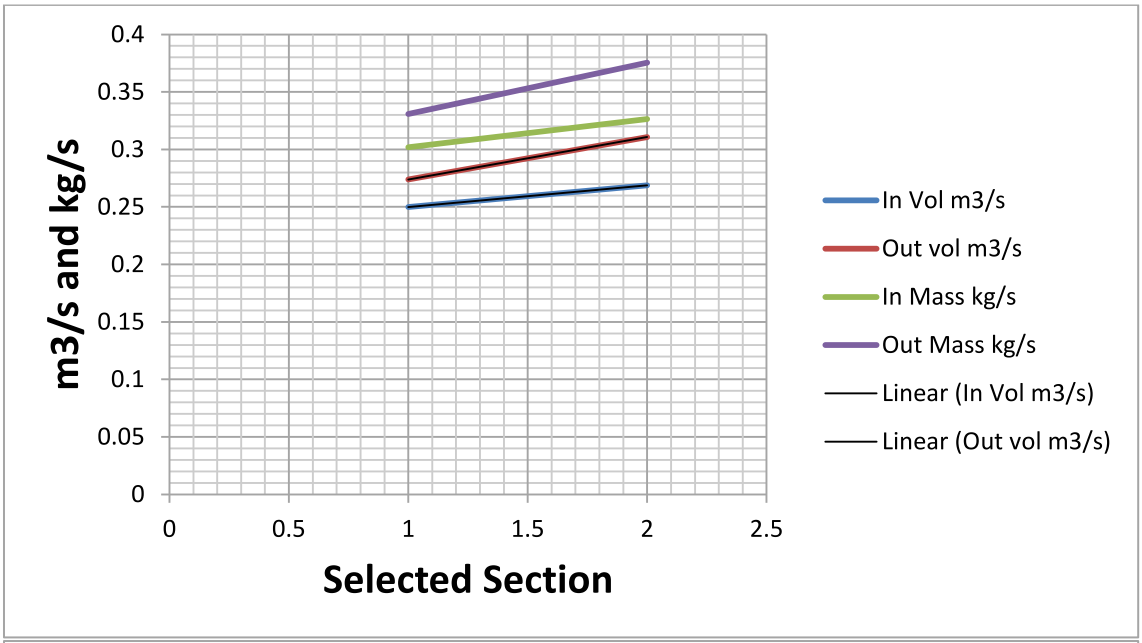


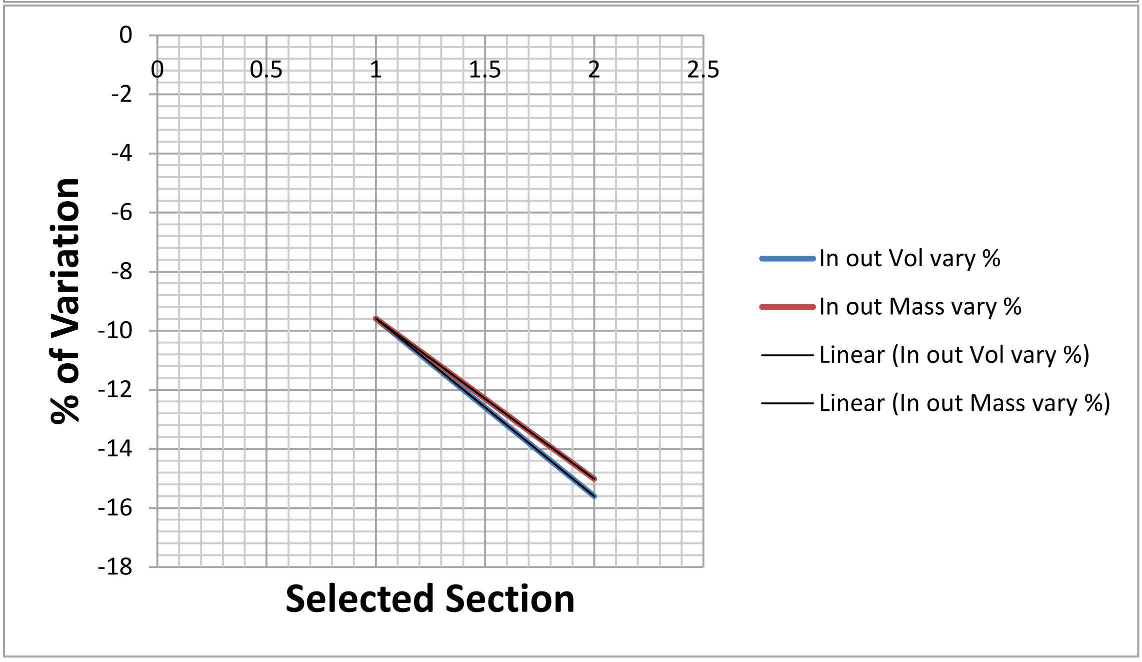
4
2
0
-2
-4
-6
-8
-10
-12
-14
-16
-18
0
1
2
3
4
5
% Variation
Selected Section
In out Vol vary %
In out Mass vary %
Linear (In out Vol vary %)
Linear (In out Mass vary %)

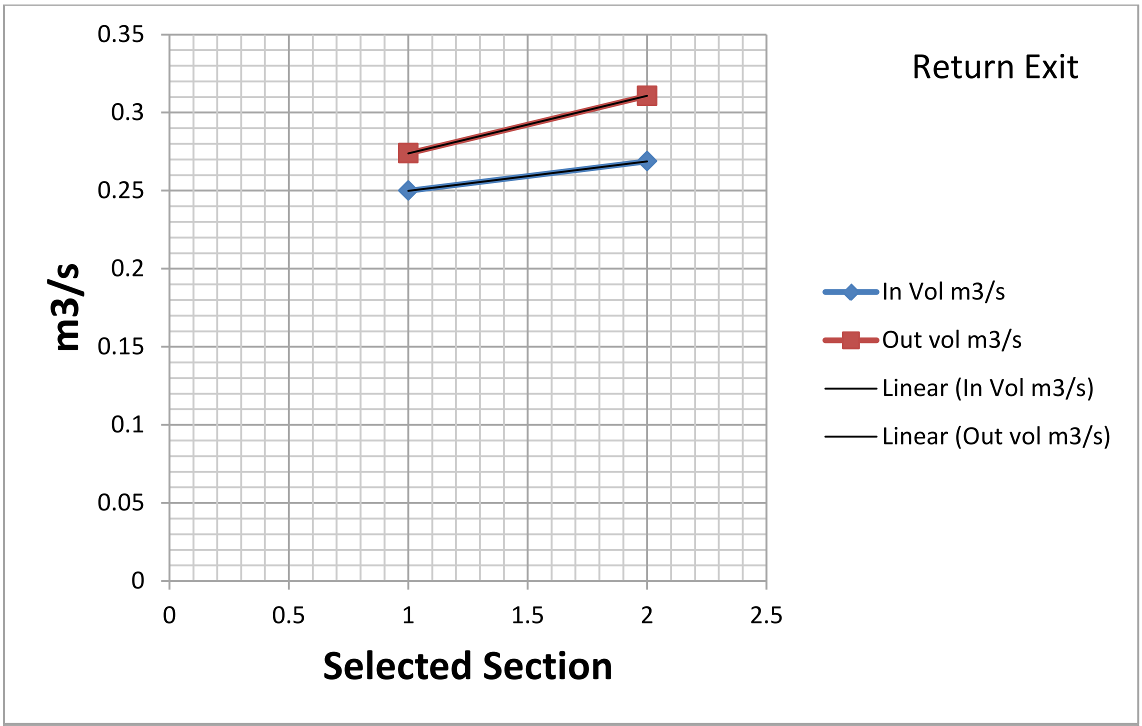


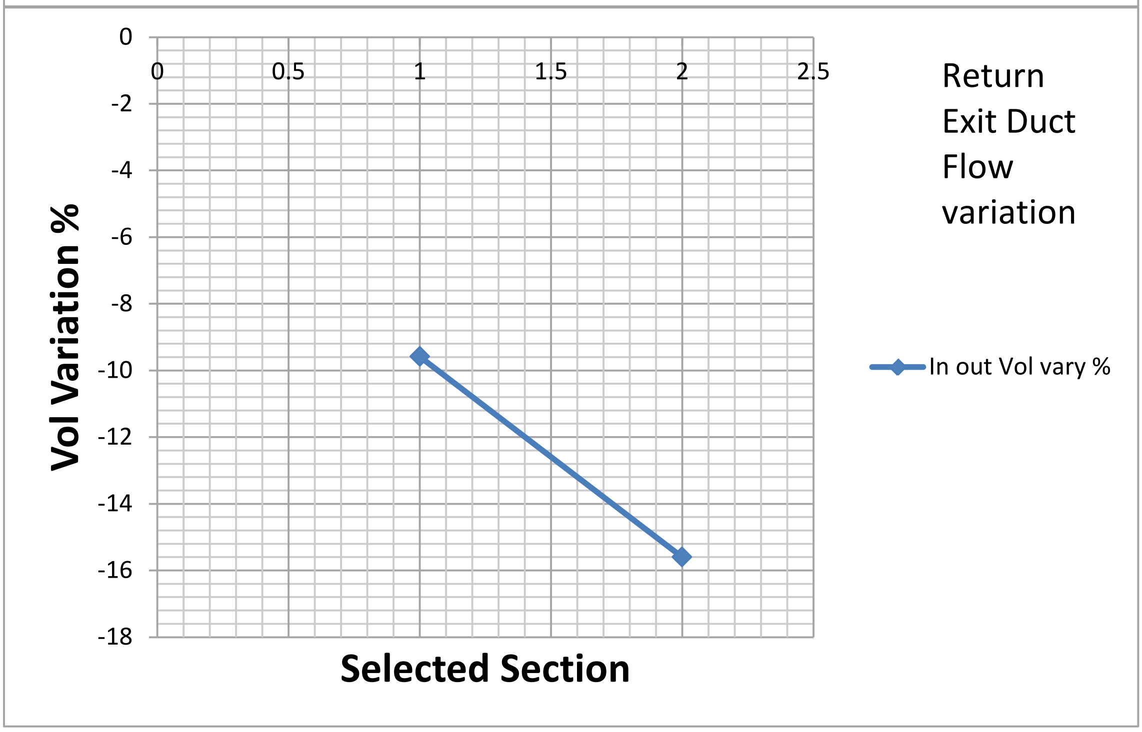
4
2
0
-2
-4
-6
-8
-10
-12
-14
-16
-18
0
1
2
3
4
5
% of Vol Variation
Selected Section
In out Vol vary %
Linear (In out Vol vary %)

2. Exit Duct air performance



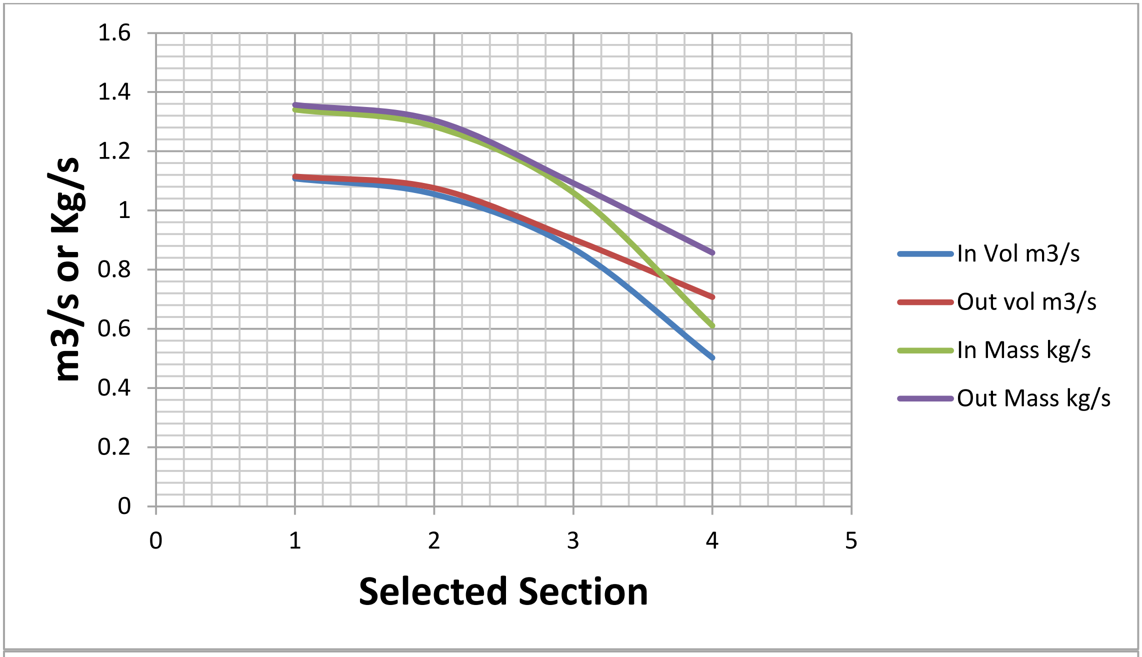


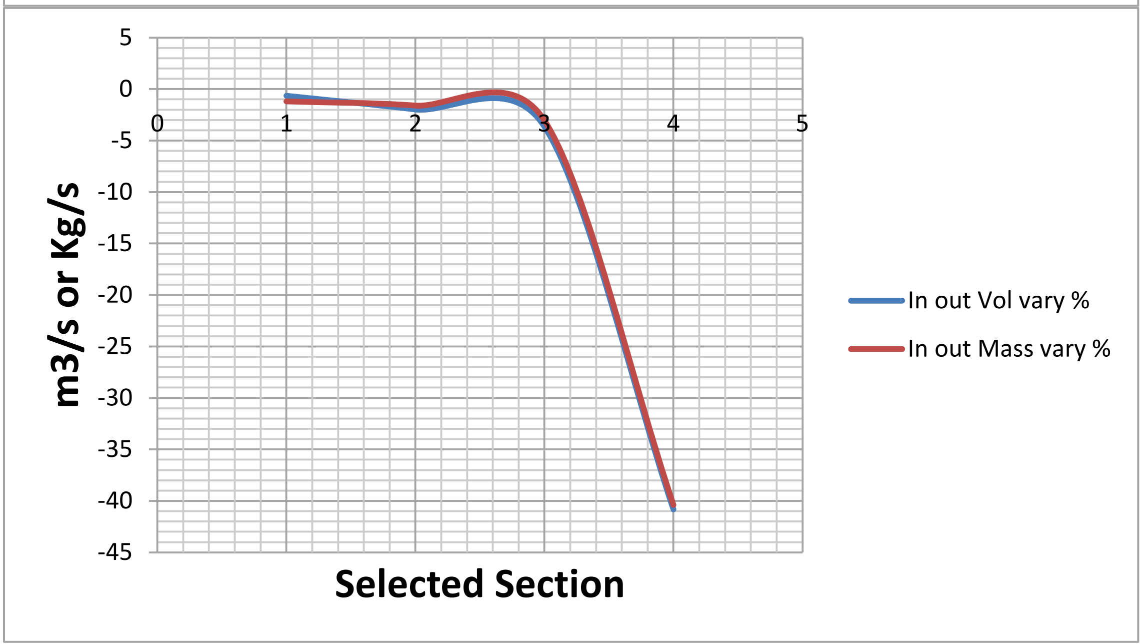

Return Exit
m3/s
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
0.5
1
1.5
2
2.5
Selected Section
In Vol m3/s
Out vol m3/s
Linear (In Vol m3/s)
Linear (Out vol m3/s)

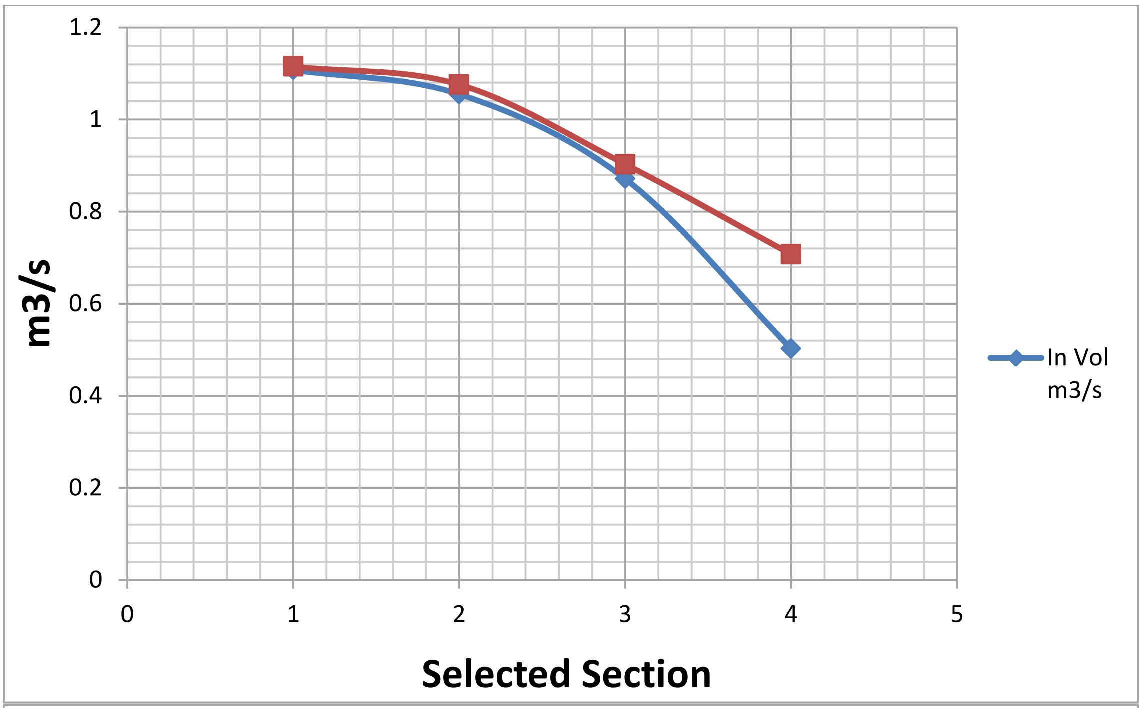

Return Exit Duct Flow variation
Vol Variation %
0
-2
-4
-6
-8
-10
-12
-14
-16
-18
0
0.5
1
1.5
2
2.5
Selected Section
In out Vol vary %

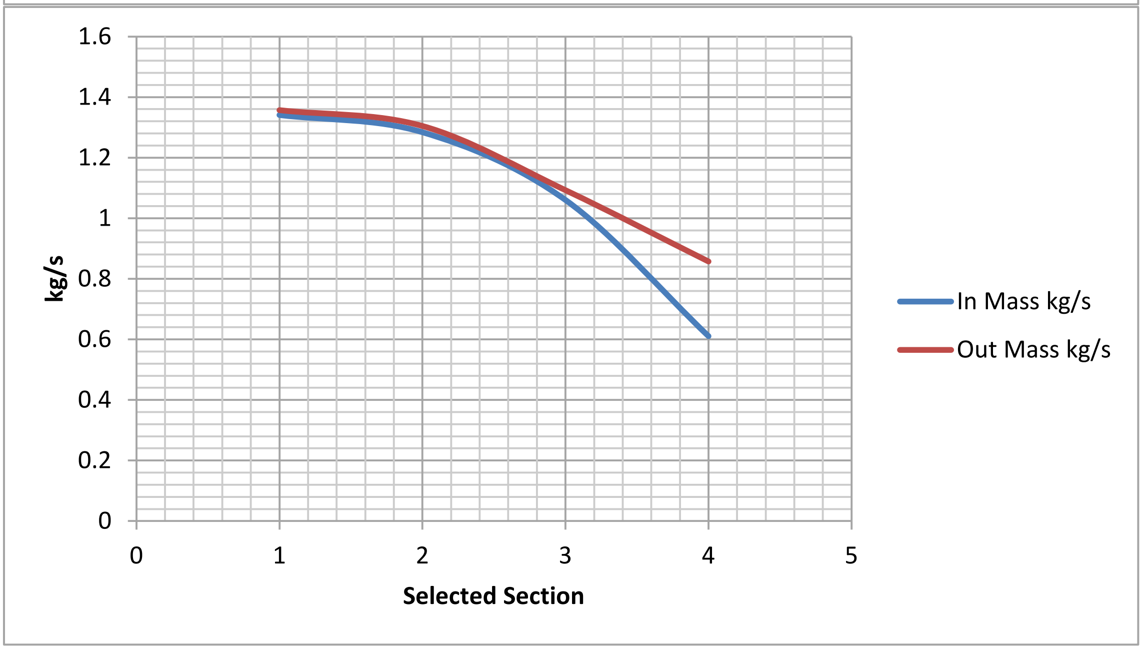
3. Supply Duct + Return Duct Performance

Faulty condition

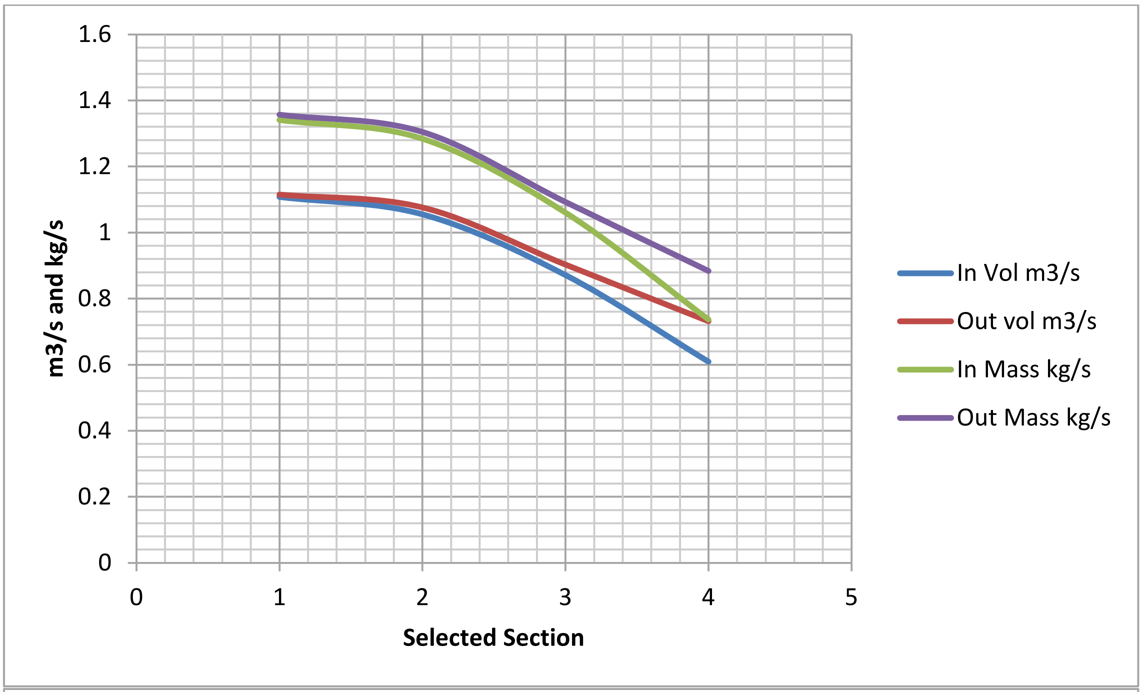


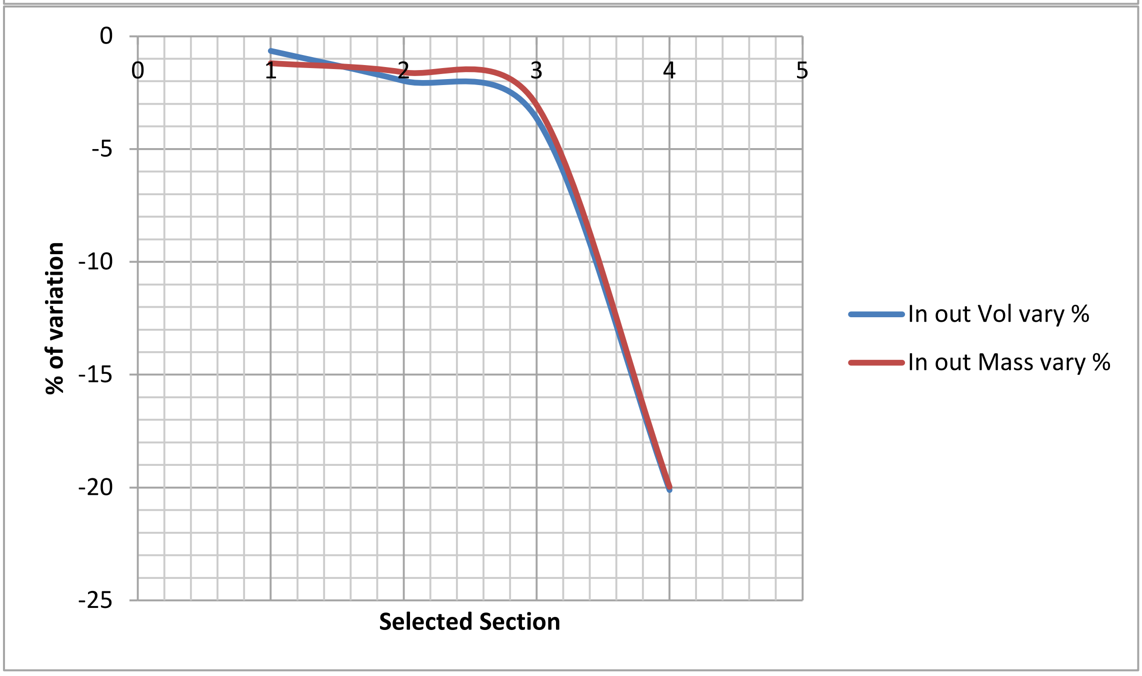


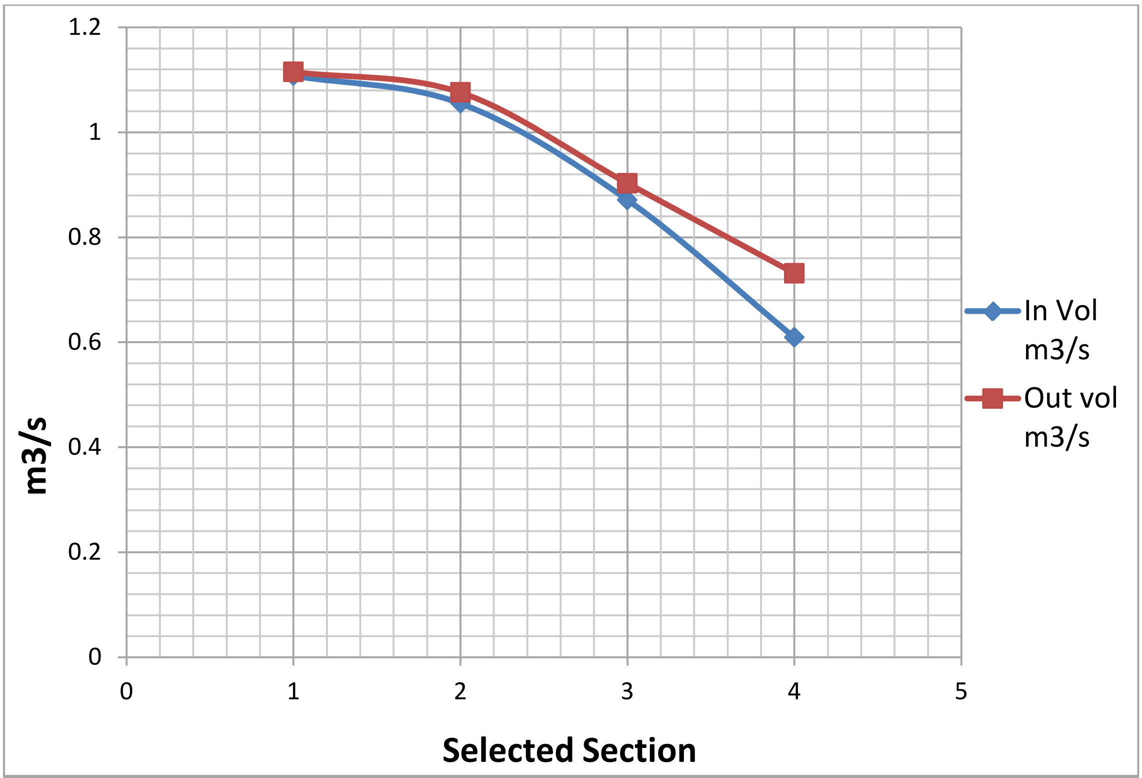

1.2
1
0.8
0.6
0.4
0.2
0
m3/s
0
1
2
3
4
5
Selected Section
In Vol m3/s

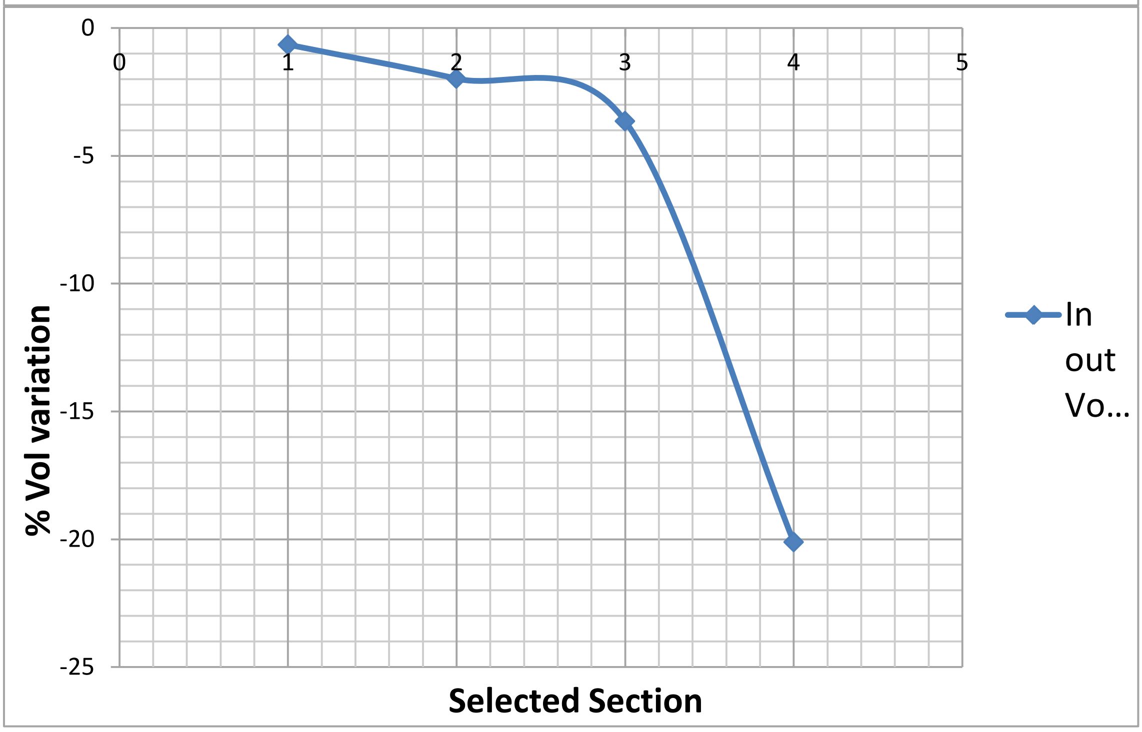

1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
kg/s
0
1
2
3
4
5
Selected Section
In Mass kg/s
Out Mass kg/s

After Fault Removed

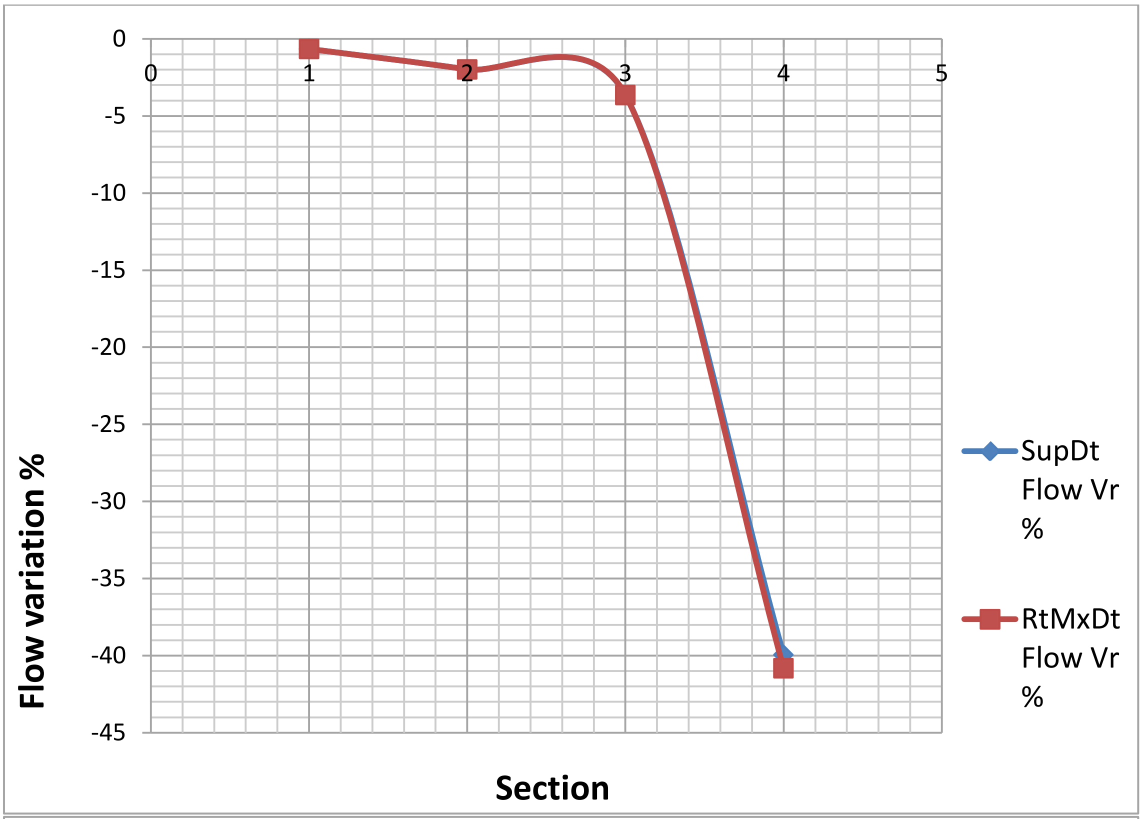


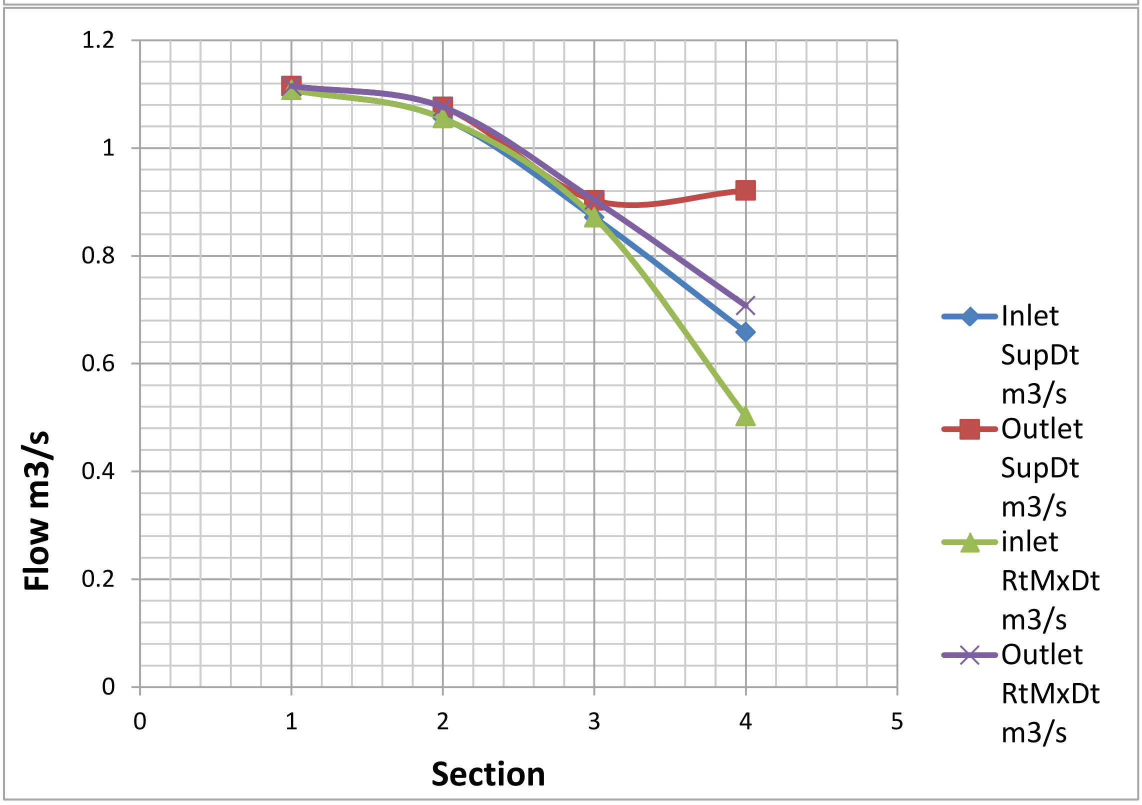



1.2
1
0.8
0.6
0.4
0.2
0
m3/s
0
1
2
3
4
5
Selected Section
In Vol m3/s
Out vol m3/s

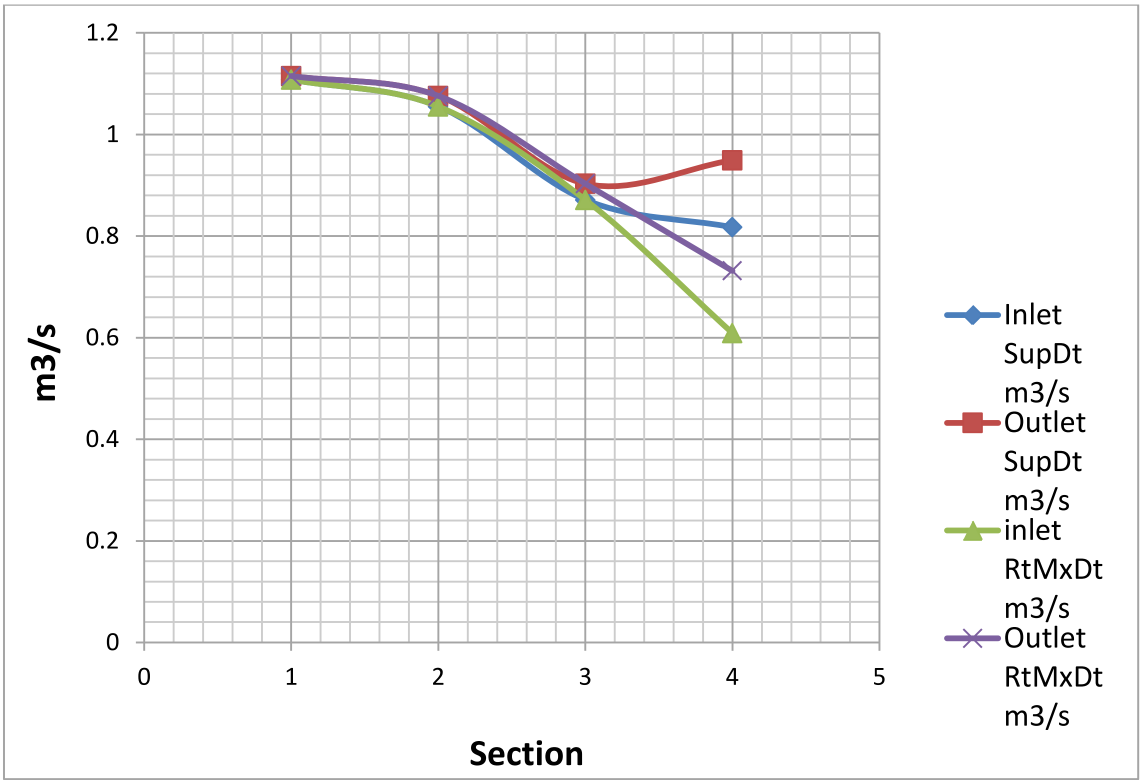

0
-5
-10
-15
-20
-25
% Vol variation
0
1
2
3
4
5
Selected Section
In out Vo...

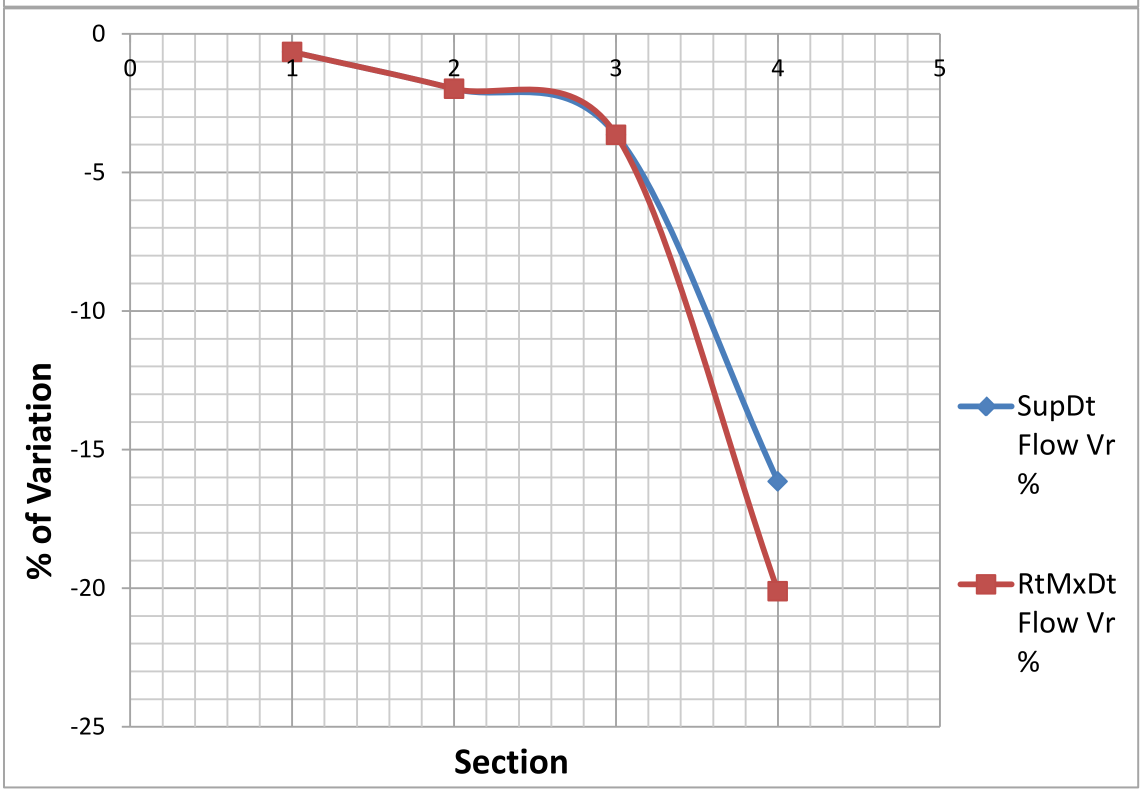
4. Supply and Return Mixing Duct with fault



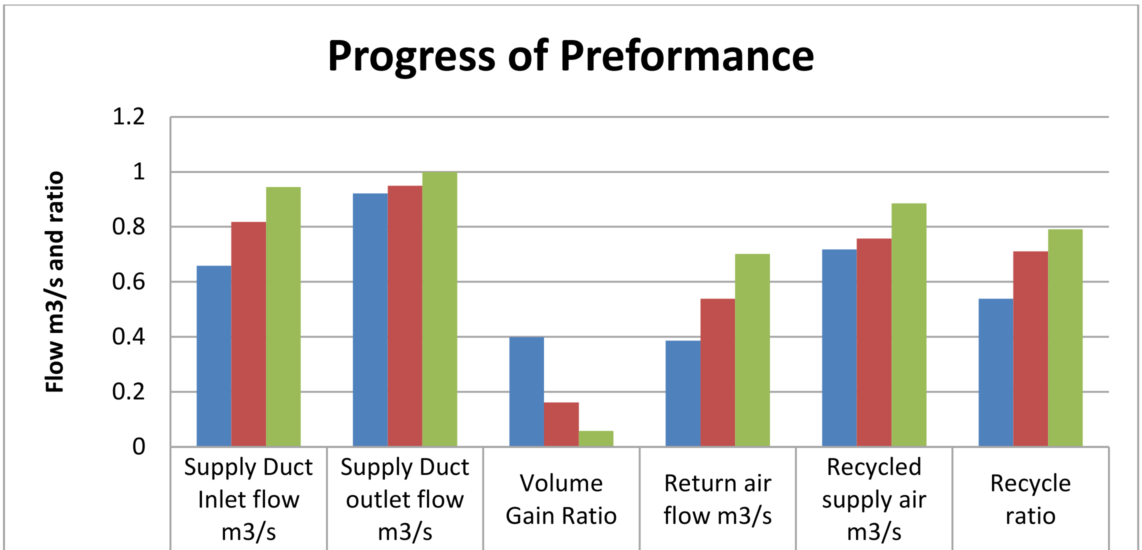


Supply and Return after fault removed

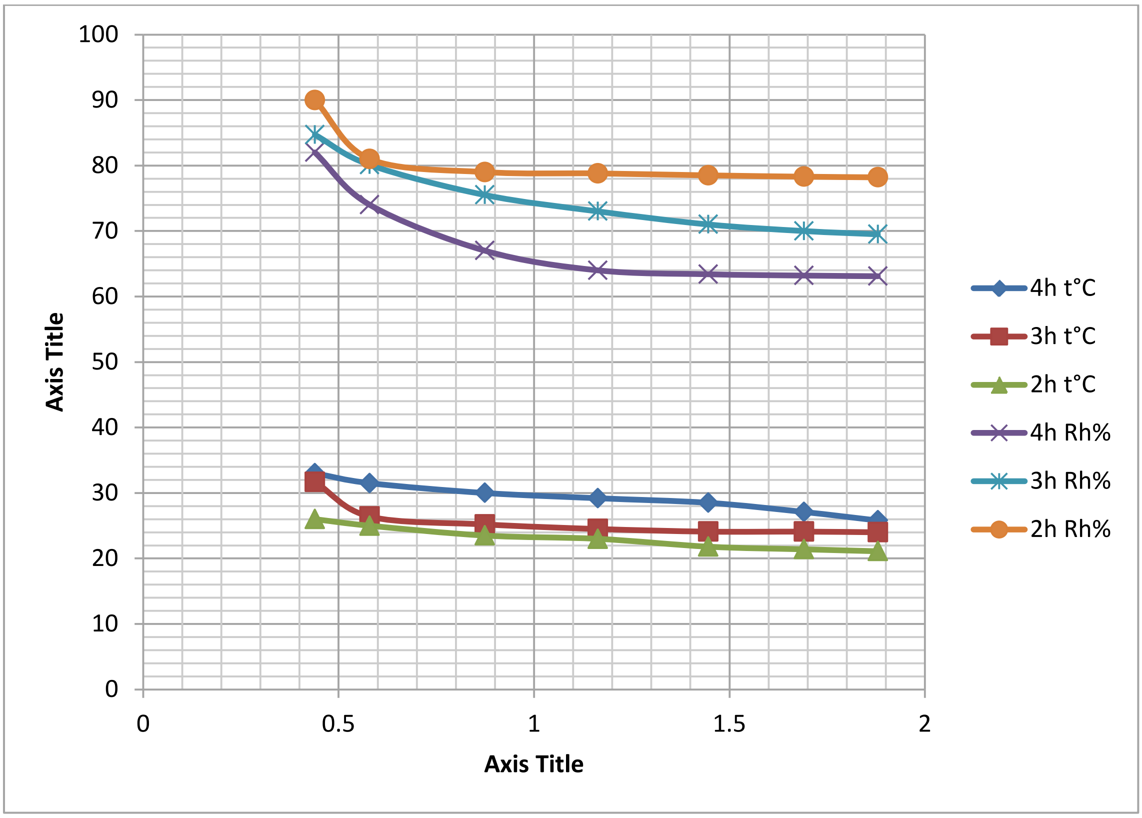




5. Performance progress



	Supply Duct Inlet flow m3/s	Supply Duct outlet flow m3/s	Volume Gain Ratio	Return air flow m3/s	Recycled supply air m3/s	Recycle ratio
Phase 1	0.658006224	0.920905875	0.399539763	0.386114001	0.71761205	0.538053954
Phase 2	0.81742203	0.9493875	0.161441049	0.538159055	0.757663206	0.710287963
Phase 3	0.94453232	0.998966625	0.057630961	0.701014526	0.885867965	0.79133071



6. artificial faults and data records

OBS	Air Pressure Pa	Env Room Load	Heater	Water flow Valve Openning %	Fresh air gate Openning %	Mix air gate openning Gate %	Speed HZ	Supply Air flow reading	Supply air m3/s	Volume Ratio of Return and fresh air	Fresh air t°C	Fresh air Rh %	Return air t°C	Return air Rh %	Mix air t°C	Mix air Rh %	Cooling coil exit t°C	Air dencity Kg/m2	Cooling coil exit Rh %
1	103500	Full	4	16	50	50	34	0.5	0.95	0.52	28.0	77.0	26.1	43.0	27.4	60.0	13.0	1.265	95.0
2	103500	Full	4	16	50	50	40	0.7	1.12	0.52	27.9	70.0	26.0	44.0	27.0	60.0	13.5	1.263	95.0
3	103500	Full	4	16	50	50	45	0.9	1.27	0.52	27.8	68.0	26.0	45.0	27.0	58.0	13.9	1.261	94.8
4	103500	Full	4	16	50	50	50	1.1	1.42	0.52	27.5	67.5	25.5	46.0	27.0	58.0	14.2	1.260	94.4
5	103500	Full	4	16	50	50	55	1.3	1.56	0.52	27.1	68.0	25.0	47.0	26.0	58.0	14.4	1.259	94.0
6	103500	Full	4	16	50	50	60	1.6	1.68	0.52	26.6	70.0	24.7	47.5	25.5	58.5	14.6	1.258	93.7
7	103500	Full	4	16	50	50	65	1.8	1.79	0.52	26.0	71.0	24.5	48.2	25.1	59.5	14.7	1.258	93.5

OBS	Air Pressure Pa	Env Room Load	Heater	Water flow Valve Openning %	Fresh air gate Openning %	Mix air gate openning Gate %	Speed HZ	Supply Air flow reading	Supply air m3/s	Volume Ratio of Return and fresh air	Fresh air t°C	Fresh air Rh %	Return air t°C	Return air Rh %	Mix air t°C	Mix air Rh %	Cooling coil exit t°C	Air dencity Kg/m2	Cooling coil exit Rh %
1	103500	Full	4	16	25	50	34	0.5	0.94	0.58	28.6	89.4	26.1	43.0	27.3	54.5	12.5	1.268	94.4
2	103500	Full	4	16	25	50	40	0.7	1.11	0.58	29.4	86.5	26.1	43.7	27.3	54.5	12.9	1.266	94.0
3	103500	Full	4	16	25	50	45	0.9	1.25	0.58	29.6	84.4	26.0	44.2	27.3	54.7	13.4	1.264	93.7
4	103500	Full	4	16	25	50	50	1.1	1.40	0.58	29.7	81.5	25.9	44.7	27.3	55.0	13.9	1.261	93.5
5	103500	Full	4	16	25	50	55	1.3	1.54	0.58	29.8	78.5	25.7	45.3	27.2	55.3	14.2	1.260	93.1
6	103500	Full	4	16	25	50	60	1.6	1.68	0.58	29.9	75.7	25.5	45.8	27.0	56.0	14.4	1.259	92.9
7	103500	Full	4	16	25	50	65	1.8	1.79	0.58	30.0	72.3	25.3	46.1	26.7	55.3	14.4	1.259	92.9

OBS	Air Pressure Pa	Env Room Load	Heater	Water flow Valve Openning %	Fresh air gate Openning %	Mix air gate openning Gate %	Speed HZ	Supply Air flow reading	Supply air m3/s	Volume Ratio of Return and fresh air	Fresh air t°C	Fresh air Rh %	Return air t°C	Return air Rh %	Mix air t°C	Mix air Rh %	Cooling coil exit t°C	Air dencity Kg/m2	Cooling coil exit Rh %
1	103500	Full	4	16	50	25	34	0.5	0.92	0.46	28.1	65.7	27.1	41.8	27.8	56.0	12.6	1.267	93.6
2	103500	Full	4	16	50	25	40	0.7	1.11	0.46	28.0	66.4	26.9	42.1	27.5	54.9	13.0	1.265	93.6
3	103500	Full	4	16	50	25	45	0.9	1.25	0.46	27.7	66.7	26.7	42.6	27.2	55.8	13.5	1.263	93.6
4	103500	Full	4	16	50	25	50	1.1	1.41	0.46	27.2	66.8	26.3	43.5	26.6	56.1	13.9	1.261	93.6
5	103500	Full	4	16	50	25	55	1.3	1.55	0.46	26.9	66.9	26.0	44.8	26.3	56.2	14.2	1.260	93.6
6	103500	Full	4	16	50	25	60	1.6	1.70	0.46	26.6	67.1	25.7	46.5	26.0	57.7	14.4	1.259	93.5
7	103500	Full	4	16	50	25	65	1.8	1.78	0.46	26.2	67.1	25.6	47.6	25.8	58.0	14.5	1.259	93.5

OBS	Air Pressure Pa	Env Room Load	Heater	Water flow Valve Openning %	Fresh air gate Openning %	Mix air gate openning Gate %	Speed HZ	Supply Air flow reading	Supply air m3/s	Volume Ratio of Return and fresh air	Fresh air t°C	Fresh air Rh %	Return air t°C	Return air Rh %	Mix air t°C	Mix air Rh %	Cooling coil exit t°C	Air dencity Kg/m2	Cooling coil exit Rh %
1	103500	Full	5	16	75	25		0.9	1.27	0.45	28.0	65.0	25.4	48.4	27.1	60.0	13.7	1.262	95.1
2	103500	Full	5	16	75	25		1.1	1.41	0.45	27.4	64.4	24.8	50.3	26.5	60.5	14.0	1.261	94.7
3	103500	Full	5	16	75	25		1.3	1.53	0.45	27.1	64.1	24.7	51.3	26.3	60.7	14.3	1.260	94.5
4	103500	Full	5	16	75	25		1.4	1.59	0.45	26.9	64.3	24.5	52.2	26.1	61.3	14.8	1.257	94.0
5	103500	Full	5	16	75	25		1.6	1.70	0.45	26.1	65.0	24.2	52.5	25.7	61.6	14.7	1.258	93.5
6	103500	Full	5	16	75	25		1.7	1.75	0.45	26.1	64.4	24.1	53.3	25.6	62.0	15.0	1.257	93.0
7	103500	Full	5	16	75	25		1.8	1.80	0.45	25.9	63.6	24.1	53.4	25.5	62.0	15.1	1.256	93.0

OBS	Air Pressure Pa	Env Room Load	Heater	Water flow Valve Openning %	Fresh air gate Openning %	Mix air gate openning Gate %	Speed HZ	Supply Air flow reading	Supply air m3/s	Volume Ratio of Return and fresh air	Fresh air t°C	Fresh air Rh %	Return air t°C	Return air Rh %	Mix air t°C	Mix air Rh %	Cooling coil exit t°C	Air dencity Kg/m2	Cooling coil exit Rh %
1	103500	Full	5	16	50	50		0.9	1.27	0.54	28.4	70.0	25.0	48.7	27.0	58.1	13.0	1.265	95.0
2	103500	Full	5	16	50	50		1.1	1.41	0.54	28.3	68.1	24.9	49.2	26.8	59.0	13.5	1.263	95.0
3	103500	Full	5	16	50	50		1.3	1.53	0.54	28.3	67.0	24.5	50.2	26.6	59.3	13.9	1.261	94.8
4	103500	Full	5	16	50	50		1.4	1.59	0.54	28.3	66.3	24.5	50.5	26.4	59.4	14.2	1.260	94.4
5	103500	Full	5	16	50	50		1.6	1.70	0.54	28.2	67.4	24.3	51.6	26.1	60.1	14.4	1.259	94.0
6	103500	Full	5	16	50	50		1.7	1.75	0.54	28.2	67.4	24.2	51.6	26.0	60.4	14.6	1.258	93.7
7	103500	Full	5	16	50	50		1.8	1.80	0.54	28.1	67.9	24.1	52.7	25.9	61.0	14.7	1.258	93.5

OBS	Air Pressure Pa	Env Room Load	Hea ter	Water flow Valve Openni ng %	Fresh air gate Openni ng %	Mix air gate openni ng Gate %	Speed HZ	Supp ly Air flow readi ng	Suppl y air m3/s	Volume Ratio of Return and fresh air	Fresh air t°C	Fresh air Rh %	Retur n air t°C	Retur n air Rh %	Mix air t°C	Mix air Rh %	Coolin g coil exit t°C	Air denci ty Kg/m 2	Cooli ng coil exit Rh %
1	103500	Full	5	16	25	50		0.9	1.27	0.60	29.3	79.0	24.5	48.7	25.7	55.0	12.5	1.268	94.4
2	103500	Full	5	16	25	50		1.1	1.41	0.60	29.4	78.3	24.3	49.5	25.5	55.0	12.9	1.266	94.0
3	103500	Full	5	16	25	50		1.3	1.53	0.60	29.4	78.3	24.2	50.2	25.5	55.3	13.4	1.264	93.7
4	103500	Full	5	16	25	50		1.4	1.59	0.60	29.4	77.6	24.2	50.2	25.5	56.0	13.9	1.261	93.5
5	103500	Full	5	16	25	50		1.6	1.70	0.60	29.2	76.7	24.1	50.9	25.4	56.6	14.2	1.260	93.1
6	103500	Full	5	16	25	50		1.7	1.75	0.60	29.2	75.9	24.1	50.7	25.4	56.5	14.4	1.259	92.9
7	103500	Full	5	16	25	50		1.8	1.80	0.60	29.2	75.8	24.1	51.2	25.4	57.6	14.4	1.259	92.9

OBS	Air Pressure Pa	Env Room Load	Hea ter	Water flow Valve Openni ng %	Fresh air gate Openni ng %	Mix air gate openni ng Gate %	Speed HZ	Supp ly Air flow readi ng	Suppl y air m3/s	Volume Ratio of Return and fresh air	Fresh air t°C	Fresh air Rh %	Retur n air t°C	Retur n air Rh %	Mix air t°C	Mix air Rh %	Coolin g coil exit t°C	Air denci ty Kg/m 2	Cooli ng coil exit Rh %
1	103500	Full	6	16	75	25		0.9	1.27	0.42	28.0	65.0	25.4	48.4	27.1	60.0	13.7	1.262	95.1
2	103500	Full	6	16	75	25		1.1	1.41	0.42	27.4	64.4	24.8	50.3	26.5	60.3	14.0	1.261	94.7
3	103500	Full	6	16	75	25		1.3	1.53	0.42	27.1	64.1	24.7	51.3	26.3	60.5	14.3	1.260	94.5
4	103500	Full	6	16	75	25		1.4	1.59	0.42	26.9	64.1	24.5	52.2	26.1	61.0	14.8	1.257	94.0
5	103500	Full	6	16	75	25		1.6	1.70	0.42	26.1	64.0	24.2	52.9	25.7	61.5	14.7	1.258	93.0
6	103500	Full	6	16	75	25		1.7	1.75	0.42	26.1	63.7	24.1	53.3	25.6	62.0	15.0	1.257	93.0
7	103500	Full	6	16	75	25		1.8	1.80	0.42	25.9	63.6	24.1	53.4	25.5	62.2	15.1	1.256	93.0

OBS	Air Pressure Pa	Env Room Load	Hea ter	Water flow Valve Openni ng %	Fresh air gate Openni ng %	Mix air gate openni ng Gate %	Speed HZ	Supp ly Air flow readi ng	Suppl y air m3/s	Volume Ratio of Return and fresh air	Fresh air t°C	Fresh air Rh %	Retur n air t°C	Retur n air Rh %	Mix air t°C	Mix air Rh %	Coolin g coil exit t°C	Air denci ty Kg/m 2	Cooli ng coil exit Rh %
1	103500	Full	6	16	50	50		0.9	1.27	0.53	29.7	68.8	25.2	47.0	27.7	57.5	14.1	1.260	95.0
2	103500	Full	6	16	50	50		1.1	1.41	0.53	29.5	67.0	25.0	47.8	27.4	57.5	14.1	1.260	94.4
3	103500	Full	6	16	50	50		1.3	1.53	0.53	29.3	65.5	24.7	49.2	27.0	59.0	14.8	1.257	94.0
4	103500	Full	6	16	50	50		1.4	1.59	0.53	29.2	64.8	24.6	49.7	26.9	58.2	14.8	1.257	93.7
5	103500	Full	6	16	50	50		1.6	1.70	0.53	29.0	64.5	24.5	50.0	26.1	59.0	14.9	1.257	93.5
6	103500	Full	6	16	50	50		1.7	1.75	0.53	28.9	64.2	24.4	50.3	26.6	58.8	14.8	1.257	93.3
7	103500	Full	6	16	50	50		1.8	1.80	0.53	28.9	64.0	24.4	50.5	26.6	59.3	14.9	1.257	93.0

OBS	Air Pressure Pa	Env Room Load	Hea ter	Water flow Valve Openni ng %	Fresh air gate Openni ng %	Mix air gate openni ng Gate %	Speed HZ	Supp ly Air flow readi ng	Suppl y air m3/s	Volume Ratio of Return and fresh air	Fresh air t°C	Fresh air Rh %	Retur n air t°C	Retur n air Rh %	Mix air t°C	Mix air Rh %	Coolin g coil exit t°C	Air denci ty Kg/m 2	Cooli ng coil exit Rh %
1	103500	Full	6	16	25	75		0.9	1.27	0.62	34.0	80.0	25.0	48.0	26.5	54.1	13.7	1.262	95.0
2	103500	Full	6	16	25	75		1.1	1.41	0.62	33.0	78.0	24.9	48.5	26.3	54.1	13.9	1.261	95.0
3	103500	Full	6	16	25	75		1.3	1.53	0.62	32.0	77.0	24.9	49.0	26.2	54.1	14.3	1.260	94.8
4	103500	Full	6	16	25	75		1.4	1.59	0.62	31.5	76.5	24.9	49.0	26.2	54.2	14.7	1.258	94.4
5	103500	Full	6	16	25	75		1.6	1.70	0.62	31.3	75.0	24.6	49.6	26.0	54.2	15.2	1.256	94.0
6	103500	Full	6	16	25	75		1.7	1.75	0.62	31.0	74.0	24.6	49.7	25.9	54.3	15.2	1.256	93.7
7	103500	Full	6	16	25	75		1.8	1.80	0.62	30.6	74.0	24.5	50.0	25.9	54.3	15.2	1.256	93.5

Appendix C: Nomenclature

A_{he}	Surface area of the heat exchanger (m^2)
A_w	Surface area of the walls, windows, etc. (m^2)
C_{cc}	Heat capacity of the cooling coil (J/˚C)
C_p	Constant pressure specific heat of air (J/kg·˚C)
C_w	Heat capacity of the side walls (J/˚C)
f_{mix}	Mixed air Volumetric flow rate (m^3/s)
f_o	Fresh air Volumetric flow rate (m^3/s)
H_{fg}	Enthalpy of water vapor (J/kg)
h_o	Outdoor heat transfer coefficient (W/m^2·˚C)
h_w	Heat transfer coefficient in the side walls (W/m^2·˚C)
l	Lewis relation
ρ	Air density (kg/m^3)
$P(T)$	Saturated Vapor pressure at temperature T (Pa)
P_s	Vapor pressure near heat exchanger (Pa)
P_{mix}	Vapor pressure of water in mixed air (Pa)
Q_{cc}	Thermal power from the cooling coil (W)
Q_{in}	Heat input provided by air conditioning system (W)
Q_{load}	Cooling load in the room (W)
Q_w	Thermal power from the wall (W)
r	System-to-fresh-air volumetric flow-rate ratio (*fmix*/*fo*)
RH_a	Relative air humidity in thermal space
RH_o	Relative air humidity outdoor
T_a	Temperature in thermal space (˚C)
T_{cc}	Temperature in the cooling coil (˚C)
T_{mix}	Mix Air temperature (˚C)
T_{mrt}	Radiant air temperature indoor (˚C)
T_o	Outdoor ambient temperature (˚C)
T_s	Supply air from the heat exchanger (˚C)
T_{wall}	Indoor wall temperature (˚C)
V_{air}	Relative air velocity indoor (m/s)
V'_{air}	Relative air velocity in the cooling coil (m/s)
V_{cc}	Effective cooling coil volume (m^3)
V_s	Effective thermal space volume (m^3)
W_a	Humidity mass ratio in thermal space
W_{mix}	Humidity mass ratio of mixed air
W_o	Outdoor humidity mass ratio
W_s	Humidity mass ratio of the supply air

END