The Pennsylvania State University

The Graduate School

Department of Architectural Engineering

BUILDING CONTROL KNOWLEDGE INFORMATION MODELING & CONTROL SELF-CONFIGURATION

A Dissertation in

Architectural Engineering

by

Yan Chen

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2015

The dissertation of Yan Chen was reviewed and approved* by the following:

Stephen Treado
Associate Professor of Architectural Engineering
Dissertation Advisor
Chair of Committee

Jelena Srebric
Adjunct Professor of Architectural Engineering
The Pennsylvania State University
Professor, Mechanical Engineering Department
University of Maryland

John I. Messner
Professor of Architectural Engineering

John Yen
Professor of Information Sciences and Technology

Chimay J. Anumba
Professor of Architectural Engineering
Head of the Department of Architectural Engineering

*Signatures are on file in the Graduate School

ABSTRACT

Building control systems play an important role in building energy conservation and indoor environment control. A large number of research studies have been focusing on the development of advanced control algorithms. These new algorithms have showed energy savings and other improvements in the studied cases. There are a few interrelated infrastructural needs among building control related research studies, applications and industrial practices, which may further facilitate the new technology deployment and standardization. First, there is a need for standard procedures to modularize control structure. Second, there is a need for standard representation formats and management tools for building control knowledge (BCK). Third, there is a need for an open-access HVAC dynamic simulation platform to develop and test advanced control algorithms. Further, there is a need for an automated control configuration mechanism to support reconfiguration of the control system, and to eliminate manual programming errors.

This study aims to bridge these gaps through an integrated study with an overall goal to develop a BCK representation and management framework, and utilize this framework to self-configure building control. More specifically, this study accomplishes a number of objectives, 1) developed a universal/standard data schema for the BCK based on analysis of existing representation formats; 2) created a database prototype to manage the BCK; 3) developed a self-configuration framework to configure control for specific building system based on interacting with BCK database; and 4) developed a dynamic HVAC simulation platform to serve as the testing bed for the control configuration.

Results from this study show that building control self-configuration can be achieved through interacting with a BCK database that enables the exchange of specific system information to initiate the self-configuration process. This study contributes to information

modeling for building control system (especially focus on the representation of BCK), the dynamic HVAC system modeling for control analysis, and the control self-configuration to support system reconfiguration and adaptation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AHU	air handling unit
ASHRAE	American Society of Heating Refrigerating and Air-Conditioning Engineers
BAS	Building Automation System
BCK	building control knowledge
BEDES	Building Energy Data Exchange Specification
BIM	Building Information Modeling
CBECS	Commercial Buildings Energy Consumption Survey
COBie	Construction-Operations Building Information Exchange
DDC	direct digital control
ER	entity-relationship
FDD	fault detection and diagnosis
GA	genetic algorithms
HVAC	heating ventilation and air conditioning
IEA	International Energy Agency
IFC	Industry Foundation Classes
MM	mixed-mode, means hybrid ventilation system
MPC	model-predictive control
O&M	operation and maintenance
ORDB	object-relational database
PID	proportional-integral-derivative
SQL	Structured Query Language
VAV	variable air volume
SIMBAD	SIMulator of Building And Devices

ACKNOWLEDGEMENTS

This is a product of a four-year effort, but without the supports and encouragements from my adviser, committee members, colleagues, friends, and families, this product could hardly be finished. Thus, I would like to express my sincerely thanks and best acknowledgement to them.

First, I would like to thank my adviser Dr. Stephen Treado for his passions and encouragement about my work. He gave me great freedom to explore the topics I am interested in and trained my ability to act independently and being self-motivated. Meanwhile, he offered me opportunities of leading course labs, giving lectures, and presenting my work at ASHRAE conferences, at the poster session during Department of Energy - Building Technology Office peer review. Without his initial offer of the research assistantship, I would have missed the opportunity of being here at Penn State Architectural Engineering program, and knowing those great faculty members and my fellow colleagues.

I would like to thank Drs. Jelena Srebric and John Messner for serving as my major committee members. Dr. Srebric gave me advices and critical comments about my research as well as careers. I learned from her about how to be a mentor and how to be passionate about research. Dr. Messner provided me insightful suggestions with detailed comments to refine my dissertation draft as well as the peer-reviewed manuscripts. His rigorous scholarship helped me to improve my scientific research, writing and editing skills. I would also like to thank Dr. John Yen from the College of Information Sciences and Technology (IST), who serves as my outside committee member. He gave me helpful assistant on understanding of terminologies and fundamentals in knowledge representation. He also provided forethoughtful comment on the self-configuration mechanism regarding the potential of control adaptation.

Meanwhile, I would like thank Drs. Chimay Anumba, James Freihaut, and William Bahnfleth for their advice on my career development. Dr. James Freihaut's positive attitude of "couldn't be better!" has enlightened me in some of my difficult times.

I would also like to extend my thanks to my fellow colleagues: Ms. Amanda Webb has been such a great friend and mentor to me. Her critical comments about my research were direct and sharp, which have significantly helped refining my work. I would also like to thank for her encouragement and support for my research exploration. I want to give my thanks to Dr. Andrew Windham for his comments and discussion as my officemate, especially during my simulation model development. Also thanks for Dr. Ke Xu for his suggestions and comments on my integrated case study. My thanks also go to Mr. Xiao Chen (from mechanical engineering), for his help with MATLAB/Simulink during my HVAC system and components model development. I would also like to thank Dr. Cynthia Reed for her suggestions about my career and the development of BCK database using Microsoft Access, which she used in her work at National Institute of Standard and Technologies (NIST). I would also like to thank Dr. Guiyuan Han for her comments on my simulation platform manuscript. I would also like to thank Mr. Yifan Liu, Ms. Li Jiang, Mr. Chong Zhou (from IST), and Mr. Fangxiao Liu for their suggestions and comments about database development and information modeling. I want to give my thanks to Mr. Xiaolong Xing for helping me debugging and teaching me developing the database SQL programming code. I would like to thank Mr. Corey Wilkinson for his logistic support with software installation, and computer maintenance.

Last, I would extend my thanks to my parents, grandparents, and Wenchi for their understanding and support, especially during my four-year Ph.D. student life.

DEDICATION

To my parents and grandparents

CHAPTER 1

INTRODUCTION

A tree as great as a man's embrace springs up from a small shoot;

A terrace nine stories high begins with a pile of dirt;

A journey of a thousand miles starts (with a single foot step) under one's feet.

The Tao Te Ching (Section: 64)

by Lao Tzu (the founder of philosophical Taoism, 6th century BC)

This chapter first gives a brief overview about challenges that are relevant to building HVAC systems associated controls development. Then, it further elaborates these challenges from several directions, i.e., building control hierarchical structure, building control knowledge representation, building control knowledge management, building control configuration process, and the dynamic simulation platform for control performance analysis. After those discussions, a research objective is proposed, and the outline of this dissertation is given at the end of this chapter.

1.1 Overview

In 1883, Warren S. Johnson invented the first electric room thermostat, which opened a new era for building industry with the capability to control indoor air temperature. Since then, revolutionary changes have been happening in buildings and their enclosed indoor environments. Building HVAC control applications vary in different types of buildings and their control variables. For example, it could be a typical office's temperature controlled through a thermostat, or a hospital operating room with its humidity, air pressure and temperature controlled by a more complex automation system. In those environments, there is no doubt that HVAC systems with their associated control systems have been playing the critical role in achieving the control objective, i.e., delivering appropriate indoor environmental quality, satisfying occupants' thermal comfort requirement, maintaining room operational setting requirement, and securing the buildings' safety against fire and other hazards. While achieving all these desirable control objectives, building HVAC systems and associated controls are also facing profound challenges:

In the whole world, buildings consumed 35% of the final energy in 2010, reported by International Energy Agency (IEA) (2014). Within U.S. alone, 41.1% of primary energy consumption is consumed by buildings, which is 6% higher than the world average ratio. This 41.1% is further divided into 22.5% from residential building sector and 18.6% from commercial building sector (U.S. DOE 2010). For a typical building life cycle, the operating energy would represent approximately 80-90% of its energy consumption based on the study by Ramesh, Prakash et al. (2010), and in colder climate, it is even higher (up to 95%) reported by Scheuer, Keoleian et al. (2003). Based on Commercial Buildings Energy Consumption Survey (CBECS) data, in 2010, 60% of operational energy end-use is related to building automation system for commercial buildings, namely, lighting (20.2%), spacing heating (16.0%), spacing cooling (14.5%), ventilation (9.1%), and refrigeration (6.6%) (U.S. DOE 2010). Thus, by summing up the spacing heating, spacing cooling, ventilation, and refrigeration, about 45% of the energy that is directly related to HVAC system. Within this 45% of energy, Akinci, Garrett et al. (2011) summarized that 25%-45% of them are wasted due to faults, including improper control logic and strategy, malfunction of controllers and controlled device, etc. A study by Roth, Westphalen et al. (2005) identified and evaluated 13 key faults in commercial building, and concluded that those faults increase commercial building primary energy consumption by about 11% of energy consumed by HVAC, lighting, and larger refrigeration system in commercial building sector (i.e., approximately one quad). If we assume the energy waste ratio did not change much from Roth's study, then the wasted energy flow in the commercial building sector can be estimated using the Sankey diagram. As shown in Figure 1-1, approximate 1.4% of the total energy was wasted in U.S., which does not include the energy waste in the residential sector. In order to promote energy efficiency in building operation, The White House (2011) has proposed a goal to achieve a minimum of 20% energy use reduction in commercial building sector in 2020.

Meanwhile, additional challenges come from the improving requirements for HVAC system to delivery appropriate indoor environmental quality with higher energy efficiency. Nowadays, people spend the vast majority of their time indoors. For example, average Americans spend 87% of their time indoors (Klepeis 2001). With people spending more time indoors, thermal comfort is becoming a universal requirement for human occupied indoor environment. Meanwhile, the understanding that some indoor environmental parameters are relevant to employees' productivities further triggers our society to recognize that appropriate indoor environmental control can be not only a health and comfort factor, but also an important economic factor. This concept has broaden the application range of HVAC and control system. However, among actual applications, there are also a series of issues about indoor environmental control that can cause discomfort, low productivity, health issues, and even morbidity and mortality (Sundell 2004). Failures or inappropriate design and operation of the HVAC system and controls could be one of the major causes for those issues. For example, inappropriate ventilation rate has been identified as one of the factors that contribute to sick building syndrome (SBS) (Sundell, Levin et al. 2011).

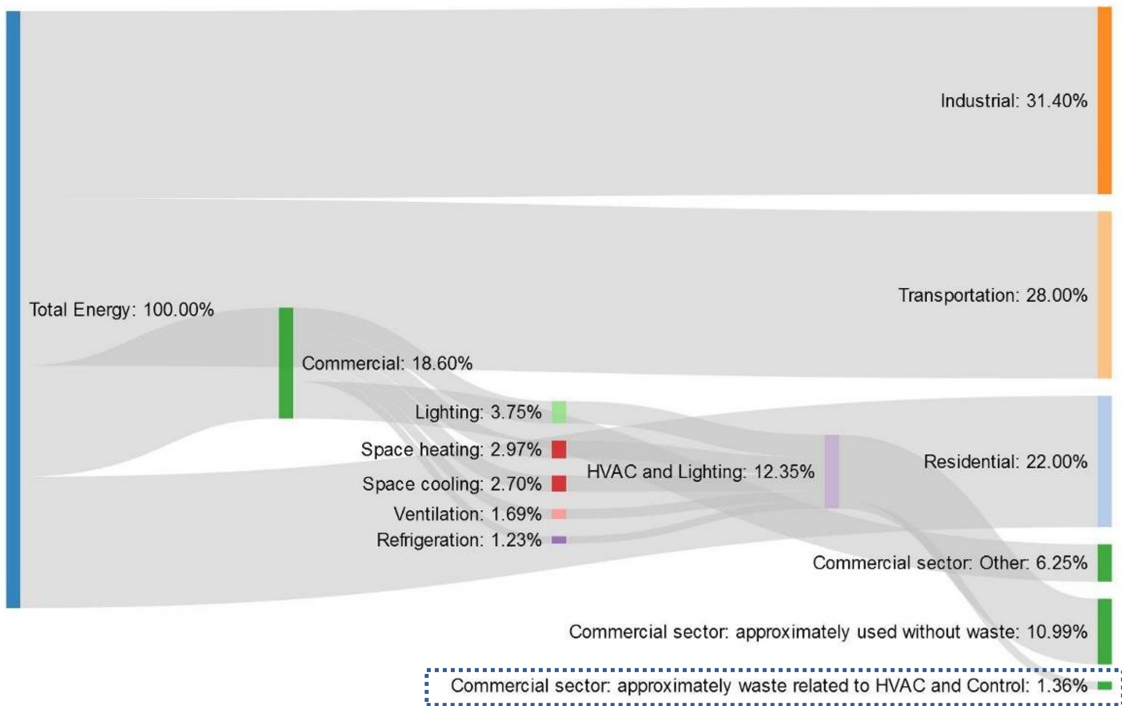


Figure 1-1 Wasted energy flow in commercial buildings sector in U.S. (estimated based on DOE data and the Roth's energy waste estimation).

1.2 Some Gaps in Building Control

A large number of studies in building control area have been exploring methods to address these challenges. For example, Dong (2010) presented an HVAC control strategy integrating weather forecasting and occupant behavior pattern predictions, which showed energy reduction of 17.8% (for cooling season), 26% (for heating season) compared to the scheduled temperature set-point control. May-Ostendorp, Henze et al. (2011) investigated model-predictive control (MPC) techniques for optimizing control sequences for window operation in mixed-mode (MM) buildings, and indicated a maximum of 40% cooling energy saving through near-optimal night cooling strategies. Ma, Anderson et al. (2011) applied a distributed MPC to control building temperature based on predictive weather and occupancy. Federspiel, Fisk et al. (2004) investigated the relationship between worker performance and ventilation rate setpoint to find a way to improve employee productivity.

These studies have made meaningful contributions to address challenges from the aspects of new control algorithm development, setpoint optimization, and FDD. Particularly, there are numerous studies on advanced control algorithms development for improving building operation, such as genetic algorithm (Huang and Lam 1997; Wright, Loosemore et al. 2002; Angelov and Buswell 2003; Lu, Cai et al. 2005; Congradac and Kulic 2009; Mossolly, Ghali et al. 2009; Khan, Choudhry et al. 2015), neural network (Soyguder and Alli 2009; Kusiak, Li et al. 2010; Kusiak, Xu et al. 2011; Kusiak and Xu 2012), fuzzy logic (Maeda and Murakami 1992; Dunlop, Burnham et al. 1995; Dounis, Tiropanis et al. 2011), and other advanced control algorithm development (Bi, Cai et al. 2000; Braun and Chaturvedi 2002). While these developed advanced control algorithms demonstrated energy savings in the case studies, the deployment of these algorithms is relatively challenging. This challenge might be associated with a number of gaps existed in the building control related infrastructural development, including:

- Representation of building control knowledge
- Management of building control knowledge
- Self-configuration of building control
- Dynamic simulation platform for control performance analysis

The following subsections further expound on these topics.

1.2.1 Representation of Building Control Knowledge

The concept of building control knowledge (BCK) is defined here as information that conveys the functionality, detailed control logic, algorithms, sequences, programming code, and the hierarchical structure of a (number of) controller(s). The representation of BCK can be analyzed from the control hierarchical structure, control modularization and module representation.

For the hierarchical structure, a two-level control structure is generally used. A discussion is given by ASHRAE Handbook - HVAC Applications (2011) as: "HVAC systems are typically controlled using a two-level control structure. Lower-level local-loop control of a single set point is provided by an actuator. For example, the supply air temperature from a cooling coil is controlled by adjusting the opening of a valve that provides chilled water to the coil. The upper control level, supervisory control, specifies set points and other time-dependent modes of operation." Similarly, Naidu and Rieger (2011) pointed out that the function of local control is to firstly ensure stability and then to ensure good set-point tracking, while the supervisory control is in response for coordinating various local controllers, and maintaining the overall operational objectives.

Then, the "two-level" control system can be further organized into several types, i.e., centralized, decentralized, distributed, and hierarchical system (Scattolini 2009; Vieira and Veiga 2009; Moroşan, Bourdais et al. 2010). From Figure 1-2 (a), it shows that centralized control system at the upper level has only one controller. Decentralized one (Figure 1-2 (b)) has multiple independent supervisory controllers. A distributed control structure (Figure 1-2 (c)) follows the decentralized structure except that there is communication among the individual controllers. A hierarchical structure (Figure 1-2 (d)) is similar to the distributed system but has an independent coordinator to manage the communication and information exchange among subsystem controllers. Among these structure types, Moroşan, Bourdais et al. (2010) concluded that a distributed control system had better control performance and lower computation requirement compared to the decentralized and centralized ones. Scattolini (2009) summarized that the decentralized control system is in lack of a stabilized algorithm. Cao, Chen et al. (2010) found that centralized control had the higher converging speed, while distributed control was more robust against packet loss, less actuation latency, excellent scalability because control relied only on local communications. For application, due to the diversity of HVAC system scale, configuration and other factors, one control structure type might work for certain HVAC system type, but not for other ones.

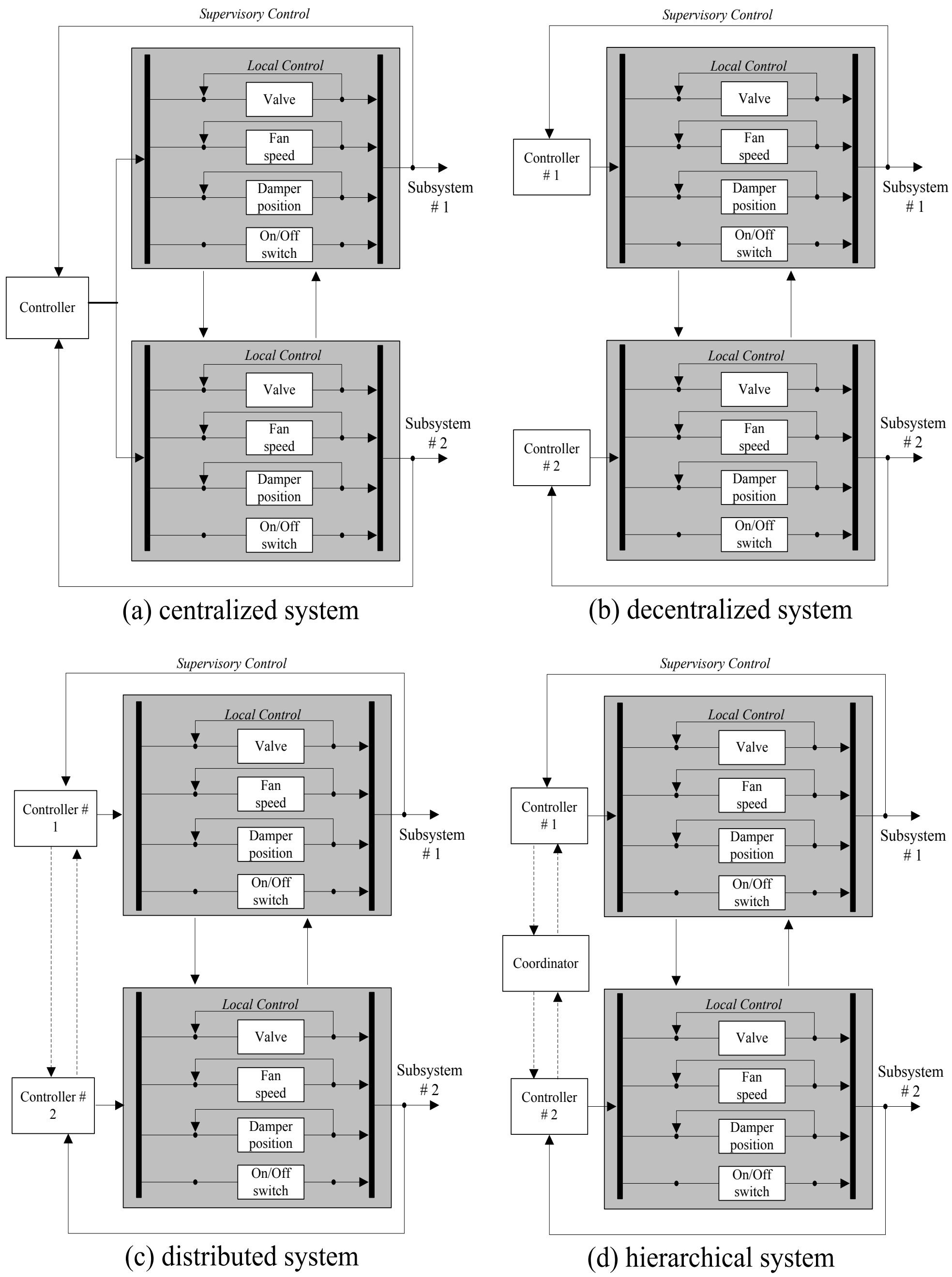


Figure 1-2 Control system structure types.

Despite the variation in control structure types of organizing multiple controllers, the representation elements of each individual controller can be identical, i.e., control sequences, programming code, control flow diagram, inputs, outputs, parameters, etc. This enables that a complex control system can be modularized into a number of individual control modules that are systematically organized to fulfill the control requirements and functionalities. There can be different ways for modularizing control knowledge representation, including modularizing based on control components, or operation modes, or control functionality, or mixed use of the previous three.

For the control module representation, a few publicly accessible resources provide collections of some levels of building control knowledge. Since there is no standard for the BCK representation, each resource represents the BCK differently. For example, CtrlSpecBuilder uses control sequence, system schematics, data points, and alarms (AutomatedLogic 2012). *Library of System Control Strategies* uses control sequence, system schematics, control flow diagrams, data points, alarms (Martin and Banyard 1998). EquipmentBuilder uses control sequences, system schematics, data points, alarms, and programming code (AutomatedLogic 2013). These different ways of BCK representation may create difficulty in understanding and interpreting the BCK documents. Besides, there are ambiguity issues in the control sequence documentation. Thus, there is a need to define an ontology for storing, retrieving and reusing structured HVAC control knowledge.

1.2.2 Management of Building Control Knowledge

Currently, there are a number of examples for managing the collection of building control knowledge. CtrlSpecBuilder developed by AutomatedLogic (2012), is a web-based tool that can generate narrative control sequences. It allows users to specify system and equipment types and then generate the templates for narrative control sequence, and data point list. However, this program does not allow users to enter new control logic, sequences, or algorithms through its interface. Some handbook and organizational guidelines also collected typical building control sequences/strategies, e.g., *Library of system control strategies* wrote by Martin and Banyard (1998), and *The sequence of operation* CD published by ASHRAE (2005). However, these resources and tools are restricted by one aspect or another, including lack of interactive interface, limited description of calculation algorithms, and lack of support for advanced algorithm deployment.

Meanwhile, a number of advanced control strategies and algorithms have been developed for achieving better building performance. For example, Huang and Lam (1997) introduced an adaptive learning algorithm based on genetic algorithms (GA) for automatic tuning of PID controllers in HVAC systems. Soyguder, Karakose et al. (2009) introduced a method with PID-type adaptive control, with self-tuning of PID parameters based on the fuzzy logic rules for different error and error change rates. Distributed model-based predictive control (MPC) was used to control building temperature based on predicted occupancy and weather (Moroşan, Bourdais et al. 2010; Ma, Anderson et al. 2011). These advanced control algorithms demands to be symmetrically collected and represented in a modern BCK resource center. Thus, the modern BCK resource center shall allow researchers to use interactive interface to contribute and publish newly developed control algorithms.

Therefore, to benefit the reusability of BCK and to facilitate the deployment of newly developed advanced ones, there is a demand to develop an interactive resource center that is not only capable of storing and managing the existing BCK, but also supports BCK upgrading with newly developed control algorithms.

1.2.3 Self-Configuration of Building Control

The existing control configuration is a manually based process. The configuration of building control system involves with building design, implementation and operation. It is related to HVAC system designer, control system designer, control system programmer and commissioner, and facility (HVAC system) operator and maintainer (Montgomery and McDowall 2008). During this process, problems happen such as inaccurate or incomplete control sequences, copy-and paste previously written sequences, sequences contradicted with other supplementary drawings, mistakenly specifying the control sequence, and programming error (Utterson 2006; Calabrese 2010). Meanwhile, the HVAC systems and components might facing the failures of systems and equipment, which requires immediate reconfiguration of the control programming code to accommodate the changes. The manual configuration process would have challenges to handle these problems.

The concept of self-configuration was primarily introduced to autonomic computing by Kephart and Chess (2003). It was used to describe when a new component introduced into an autonomic system, it will automatically learn about the system, and configure itself and modify its behavior appropriately. A number of patents and studies have been developed to realize the control self-configuration. These works may be divided into two categories, the self-configuration of the whole building control system (Baldwin, Bishop et al. 1994; Ryan and Shah 2007; Wruck 2008; Harrod, Rigg et al. 2011), and the self-configuration (or mostly called "self-tuning", or sometimes "self-organizing") of one specific control loop (Procyk and Mamdani 1979; Nesler 1986; Shao 1988; Maeda and Murakami 1992; Dunlop, Burnham et al. 1995; Xu, Wang et al. 2004; Bai, Wang et al. 2008; Seem 2013). However, the existing methods and products have some limitations. First, these self-configuration methods are mostly designed for the application of specific building HVAC types or one specific HVAC system component.

Thus, these methods are not generic, and each system type would need a dedicated self-configuration method. Second, these methods typically used fixed control algorithms and it would be difficult for them to support future advanced control algorithms upgrades.

Apparently, these limitations restrict the application of these self-configuration methods at some level, because the industry today is still largely depended on the manual configuration process for building control. Thus, it would be beneficial to develop a generic self-configuration method that can potentially support advanced algorithms upgrades.

1.2.4 Dynamic Simulation Platform for Control Analysis

The development of new control algorithms or structures requires the test of new algorithm either in field or through simulation. In most cases, field tests might not be suitable because it would be time-consuming, costly, and involves with safety risks. Simulation platform can provide an opportunity to evaluate and compare different control strategies in a virtual environment. As Trčka and Hensen (2010) said, it has also been seen as promising tools for establishing the baseline (or baseband) performance predictions, which can be used to monitor performance or detect fault.

Simulink based HVAC modeling has been popular and powerful for analyzing the dynamic response and control algorithm development (Underwood 1999; Riederer 2005). Recently, a large number of MATLAB/Simulink based HVAC control studies have been reported (Tashtoush, Molhim et al. 2005; Wu, Melnik et al. 2007; Congradac and Kulic 2009; Karmacharya, Putrus et al. 2012; Avci, Erkoc et al. 2013; Afram and Janabi-Sharifi 2015; Khan, Choudhry et al. 2015). However, few of these studies provide a shared Simulink HVAC dynamic model library. Due to this fact, researchers still have to redevelop the HVAC dynamic models, which have been developed by other researchers, but not shared. This process is usually time-consuming.

Considering the advantages of using MATLAB/Simulink for advanced control algorithm development, the cost of existing commercial library, and the redundancy in the part of developing MATLAB/Simulink models in the existing work, a free open-access library for the dynamic model of the HVAC system and components in MATLAB/Simulink would be beneficial.

1.3 Research Objectives

Based on the previous discussion, it is clear that a number of gaps exist among building control related research studies, applications and industrial practices. Namely, there are a lack of standard representation formats of BCK, a lack of interactive BCK resource center, a lack of open-access HVAC dynamic simulation platform in MATLAB/Simulink, and a lack of generic self-configuration process. These gaps may seem independent, but they are actually interrelated. Therefore, this study aims to bridge these gaps through an integrated study with an overall research goal to develop BCK information modeling and management framework, and then utilize it to self-configure building control.

More specifically, this study targets to accomplish a number of research objectives, including:

1) To develop a universal/standard data schema for the BCK based on analysis of existing representation formats.
2) To create a prototype BCK database.

3) To develop a self-configuration framework based on interacting with the BCK database.

4) To develop a dynamic HVAC simulation platform in MATLAB/Simulink to test the control configuration result.

1.4 Outline of the Dissertation

Based on the list of research objectives, this dissertation is divided into six additional chapters. Chapter 2 to 5 address the challenges from four independent yet highly related perspectives, i.e., the BCK data schema, the BCK database, the self-configuration process, and the dynamic HVAC system and control simulation platform (shown in Figure 1-3).

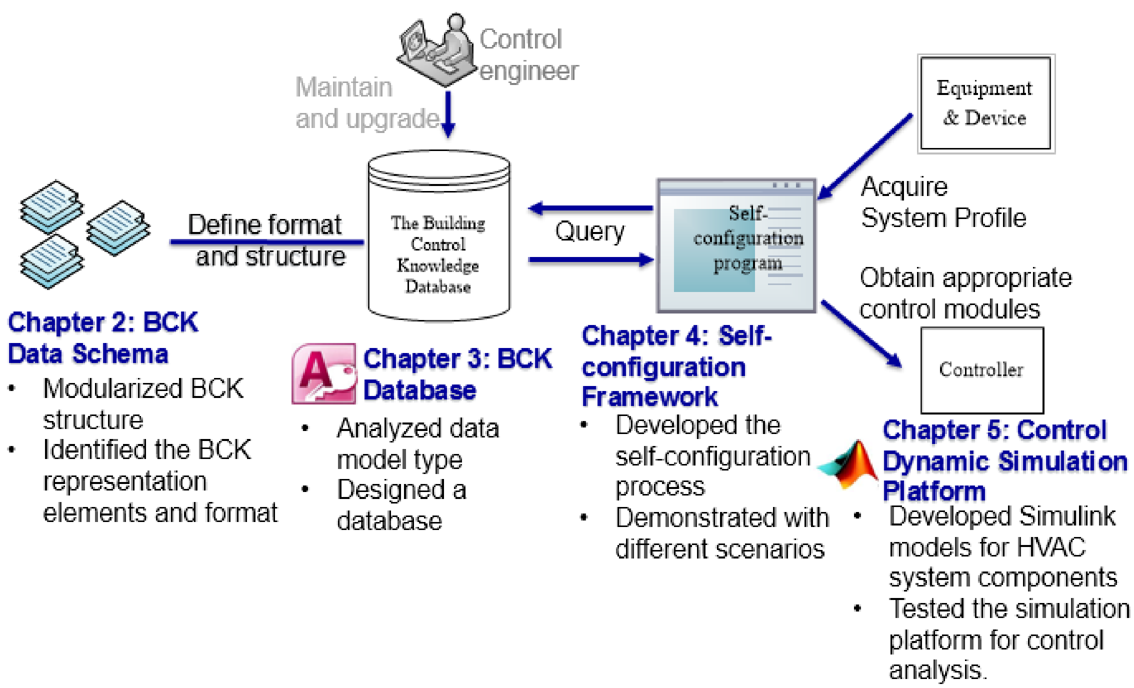


Figure 1-3 Overview of the major chapters in this dissertation.

Chapter 2 introduces a universal/standard data schema for BCK representation. First, it defines the scope of BCK. Next, it analyzes the existing literature and industry practices. Based on this analysis, it introduces a new approach to modularize the whole building control into multiple control modules; for each control module, a list of representing elements with standard format were proposed. Last, the new approach is implemented via a case study and compared with the existing representation to demonstrate its benefits.

Chapter 3 focuses on integrating identified BCK data schema into a database. Microsoft Access 2013 is used as the main tool for the first version database development. Meanwhile, a user-friendly interface has been created to support accessing the information and adding new data. Overall, this database is the foundation for the proposed self-configuration method. It not only servers as the storage house of BCK for different HVAC system types, equipment and components (discussed in Chapter 2 and 3), but also acts as part of the infrastructure for the self-configuration (discussed in Chapter 4).

Chapter 4 focuses the development of a generic self-configuration framework. This self-configuration framework is created based on interacting with the BCK database and the typical process of configuring specific BCK from current practices. Four self-configuration scenarios are developed and their processes are detailed through the process map.

Chapter 5 discusses the development of dynamic HVAC system and components simulation platform for HVAC system and control performance analysis. It utilizes the dynamic physical systems and components model. A library for HVAC systems and components models in MATLAB/Simulink is developed. This provides a simulation platform that can test the performance of different control strategies in a simulated scenario.

Chapter 6 presents a case study demonstration that utilizes the self-configuration framework to select the applicable BCK modules from the BCK database and configure them based on the case project specifications. Two self-configuration scenarios are demonstrated, i.e., the New System Configuration scenario and the Sensor Failure Reconfiguration scenario. Then, the configured BCK modules are evaluated on the dynamic HVAC system simulation platform to compare the performance of the two scenarios.

Chapter 7 summarizes this dissertation and presents recommendation for future work.

CHAPTER 2

BUILDING HVAC CONTROL KNOWLEDGE DATA SCHEMA – TOWARDS A UNIFIED REPRESENTATION OF CONTROL SYSTEM KNOWLEDGE

Nothing can be accomplished without norms or standards

-Proverb

This chapter discusses the development of a universal/standard data schema for BCK representation. First, it analyzes the existing literature and industry practices for BCK representation. Then, it introduces a new approach to modularize the whole building control into multiple control modules; for each control module, a list of representing elements with standard format were proposed. Last, the new approach is implemented via a case study and compared with the existing representation to demonstrate its benefits.

2.1 Introduction

The concept of building control knowledge (BCK) is defined here as information that conveys the functionality, detailed control logic, algorithms, sequences, programming code, and the hierarchical structure of a (number of) controller(s). BCK plays an indispensable role in the Building Automation System (BAS), realizing the functionality and operation of the building HVAC system and components. Practically, BCK is primarily created by the HVAC and control system designer, and represented in the format of narrative control sequences. For example, the control sequence for a thermal zone temperature setpoint may be described as "During the occupied mode, the setpoint shall be 23°C for cooling and 21°C for heating; during the unoccupied mode, the setpoint shall be 29°C for cooling and 13°C for heating". Then, narrative sequences are interpreted into programming code by a programmer, and implemented into BAS. Thus, control sequences and programming code might be viewed as the main representational artifacts for BCK. The degree of accuracy and unambiguity in the BCK representation would affect the quality of the control system design, implementation, commissioning, operation and maintenance, and further contribute to the overall building performance.

Efforts have been made for information modeling related to BAS, where the BCK are implemented. For example, Building Information Modeling (BIM) open information exchange standard *ISO 16739:2013 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries* have adopted information elements to represent BAS as part of their schema (International Organization for Standardization 2013). Also, there have been recent attempts to develop IFC model view definitions to document standard exchange elements for control information (e.g., the Building Automation Modeling information exchange (BAMie) Model View Definition (see buildingSMART International (2013)). The U.S. Department of Energy has also recently focused on the development of the Building Energy Data Exchange

Specification (BEDES) which included a section on “Control and Operation” (U.S. DOE 2014). Construction-Operations Building Information Exchange (COBie) has been developed to support the collection of information during the design, construction and operation stages to increase the data availability (East 2014). Lucas, Bulbul et al. (2013) proposed an object-oriented product model to support healthcare facility information management. Liu, Akinci et al. (2013) presents extensions to the Information Delivery Manual approach and the results of using it to identify information requirements for performance analysis algorithms of HVAC systems, especially for fault detection and diagnosis (FDD). Dong, O'Neill et al. (2014) developed a BIM enabled information infrastructure for FDD, which focused on data schema for the information exchange process. It is also notified that the current BIM effort lacks of detailed data schema for the control sequences representation. Kim, Jeong et al. (2015) developed a Modelica library for BIM-based building energy simulation. These efforts focused on the information modeling and analysis of building system. They provided documentation of the various physical building components that are used to build the HVAC and control systems in BAS. However, there was not a study that investigated the systematically representation of the BCK. This is critical for understanding the operational characteristics of the system.

Currently, there are noticeable issues in the existing styles and formats of BCK representation. First, there are no standard element lists for the elements within BCK. For example, CtrlSpecBuilder by AutomatedLogic uses control sequence, system schematics, data points, and alarms (AutomatedLogic 2012). *Library of System Control Strategies* uses control sequence, system schematics, control flow diagrams, data points, alarms (Martin and Banyard 1998). EquipmentBuilder uses control sequences, system schematics, data points, alarms, and programming code (AutomatedLogic 2013). The difference in selecting the representing elements creates difficulty in understanding and interpreting the BCK documents. It would be much easier to understand the BCK if a standard list of BCK representing elements can be defined.

Second, there is no standard format for describing each element in the BCK. Control sequences can be viewed as a representative for the format issue. Based on *ASHRAE Guideline 13-2007 Specifying Direct Digital Control Systems* (ASHRAE 2007), there were two methods of organizing control sequences: 1) by operation mode, and 2) by component, . However, in practice, mixed use of these two methods to organize control sequences and programming codes are common. Other problems include inaccurate or incomplete control sequences, copying and pasting previously written sequences, sequences contradicted with other supplementary drawings, mistakenly specifying the control sequence, and programming errors (Utterson 2006; Calabrese 2010). These problems can directly affect the normal operation of building system, which would have an impact on building energy efficiency, indoor air quality, and/or thermal discomfort. Thus, there is also a need to define standard format for each of the representing elements.

Third, BCK representation of a larger control system is usually more complex, and it is challenging to interpret and understand. In the U.S., approximately 90% of the building floor space is contained within medium and larger size commercial building, i.e., medium size (5,000-50,000 ft^2, 41%) and large size (+50,000 ft^2, 49%) (U.S. DOE 2003). As a result, the representation of BCK for large buildings becomes relatively complex. Most current documenting resources present the BCK of the entire HVAC system altogether without modularizing the BCK into smaller modules (e.g., in *Sequence of Operation for Common HVAC System* (ASHRAE 2005)). A few commercial products present HVAC components based upon modules. However, their modularization is incomplete and their control sequences and programming codes are still organized in a mixed format (e.g., EquipmentBuilder from AutomatedLogic). Thus, a complete and thorough modularization process is needed to reduce the complexity of BCK representation.

The objective of this chapter is to develop a universal/standard data schema for BCK representation. Specifically, the following sections present the analysis of existing literature and industry practices for BCK representation, introduce a new approach to modularize BCK representation, and discuss the format of attribute for each representing element. A test case has been implemented to evaluate the data schema and compare with existing methods.

2.2 Existing Representations and Formats

Before introducing the new style and format, this section further analyzes the existing representing method and formats in details. First, existing resources that represented the BCK are reviewed and compared. The resources include published guidelines, industrial application tools, and books. The following provides brief descriptions for the analyzed documents.

1. *ASHRAE Guideline 13-2007* is a guideline for specifying the DDC system (ASHRAE 2007). It focuses on the DDC system design procedure, but also discusses the formats of control drawings, sequences of operations, and objects list.
2. CtrlSpecBuilder is a web-based application tool developed by AutomatedLogic (2012). It provides control system design guideline and can generate detailed control sequences for most of the HVAC system types.
3. EquipmentBuilder for Educators 5.5 is a downloadable free software developed by AutomatedLogic (2014). It can generate control sequence, system schematics, and WebCtrl® programming code for a number of pre-defined HVAC systems and equipment.
4. *Library of system control strategies* is a reference book for specifying, developing and configuring control strategies by the Building Services Research and Information Association (Martin and Banyard 1998). It provides comprehensive sequence specification, highlighted by its detailed control flow diagram for easier understanding.

5. *ASHRAE Sequence of Operation for Common HVAC System* is a reference CD-ROM for specifying the control sequence provided by ASHRAE Technical Committee 1.4 Control Theory and Application (ASHRAE 2005).

Then, the representing elements and their format quality are summarized in Table 2-1. From this comparison, some elements were typically used among those documents, including controller name, control objective, operation mode, system schematic, inputs, outputs, trends, and alarms. Thus, those elements were included in the list of representing elements. Besides, the list also included a few less frequent used elements, including: control flow/logic diagram, functions/algorithms, and programming code. The following subsections discussed the functionality and format of each selected BCK representation element.

Table 2-1 Comparison of existing published documents for BCK representation.

Representing Elements	**1**	**2**	**3**	**4**	**5**
Control module name	-	**	*	**	***
Control objective	*	-	-	***	***
Operation mode	*	*	*	***	*
System schematic	***	***	***	***	***
Control flow diagram	*	-	-	***	-
Object list (inputs/outputs/parameters)	***	***	***	***	***
Alarms	***	***	***	*	***
Control sequence	**	**	**	**	**
Functions/algorithms	*	-	-	-	-
Programming code	*	-	***	-	-

***: listed with a standard format, clear and unambiguous
**: listed with a standard format, but not well-organized, and have ambiguity issues
*: listed without a standard format
-: not listed

Particularly, for the most important elements in the BCK representation (i.e., control sequences and programming code), there are problems in their documenting style and formatting in existing commercial templates. For example, here we used the EquipmentBuilder to generate the control sequences and programming code documents. For each of the major HVAC system and components, there are one control sequence document and one control code document (e.g., Variable Air Volume (VAV) terminal unit). Then, within each file (control sequence or programming code file), its layout has mixed use of documenting style (i.e., mixed arrangement of operation mode, setpoint configuration, and actuator control). For example, as can be seen in Table 2-2, the layout of control sequences document have mixed uses of operation modes (i.e., Zone Optimal Start, Zone Unoccupied Override), actuators (i.e., Reversing Variable Volume Terminal Unit, Reheating Coil Valve), and setpoint configuration (i.e., Run Conditions, Zone Setpoint Adjust, Demand Limiting). Similar problem happened for programming code. This represents the problem of mixed use of documenting style, which would confuse the interpretation and understanding of these control documents.

Table 2-2 Layout of control sequences and programming code from a template example in EquipmentBuilder for Educators.

<table>
<tr><th>Control Sequences
(Viewed in Word: the alarm and data point sections are excluded)</th><th>Programming Code
(Viewed in EIKON: the alarm and data point sections are excluded)</th></tr>
<tr><td>
Run Conditions – Scheduled
Minimum Ventilation on Carbon Dioxide (CO_2) Concentration
Demand Limiting – Zone Setpoint Optimization
Zone Setpoint Adjust
Zone Optimal Start
Zone Unoccupied Override
Reversing Variable Volume Terminal Unit – Flow Control
Occupied
Unoccupied
Reheating Coil Valve
</td><td>
Zone - Run Conditions – Scheduled
Outgoing Requests to Run Interlocked Equipment
Request Heat Source
Request Cool Source
Zone Occupied for - Minutes
Zone CO_2 Control
Airflow Control – Reversing – Internal Actuator
Outgoing Requests to Run Interlocked Equipment
Heating Control
</td></tr>
</table>

Thus, existing representing structure and formatting would barely satisfy the requirement of an explicit and logical BCK representation without ambiguity. Thus, there are issues for modularizing BCK, selecting BCK representing elements selection and formatting. This chapter tries to address those issues and standardize the BCK representation.

2.3 A New Modularization Approach

This approach includes two major parts (see Figure 2-1). The first part presents a process for completely modularizing BCK representation, by transforming the one-big-complex BCK thoroughly into many small control modules. The second part identifies a list of representing elements for a typical control module. Then, it defines appropriate format for each element's attributes. Existing documents are analyzed based on their format quality (i.e., degree of consistency and unambiguity). If an existing format is well-defined without ambiguity, it will be adopted to format the representing elements; otherwise, new format must be created. The following sub sections discuss each part in details.

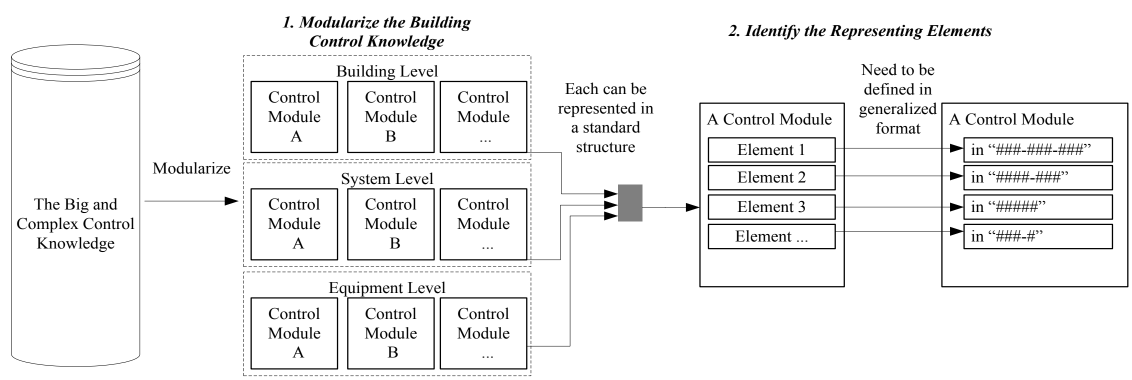


Figure 2-1 Overview of new control modularization method.

2.3.1 BCK Modularization

Modularization of BCK is intended to reduce the system complexity based on the structural or functional similarities in control system. This transforms a big and complex BCK into many small and simple modules organized in a rational structure. For a BAS, each control module can be viewed as an individual object. This object has its own inputs, parameters, embedded control logic, and the outputs. While it functions as an independent subsystem, the inputs might depend on other subsystem, or its outputs might be used as inputs for other subsystem. Usually, for a large system, a number of control modules can be organized in a hierarchical or distributed structure. Thus, it becomes natural and feasible to modularize the BCK representation.

An ontology analysis of the BCK illustrates this rational structured feature. Shown in Figure 2-2, the control modules are classified into three layers (i.e., building layer, system layer, and local layer). This expanded the ASHRAE-defined two-layer classification into three layers. Particularly, it expands the supervisory layer into building layer and system layer. The building layer manages control modules that functioned to determine the optimal point for building layer (e.g., operation mode, optimal point for building energy usage and operational cost). The system layer manages control modules that function to determine the optimal point for specific system (e.g., modify the setpoint based on system wide information). The local layer manages control modules that functioned to maintain the setpoint by actuating local components.

The modularization process is designed based on the determination of control variables via ontology analysis. As shown in Figure 2-3, for an HVAC system, the first major step is to identify the hardware control variables (e.g., supply fan speed). For each hardware control variable, a local control module is built by identifying its depended inputs, parameters, and virtual variable(s). Virtual variables are outputs from system or building control modules, and maybe used in local or system control modules as inputs. For each virtual variable identified, a system/building control module is built by identifying its dependent inputs, parameters, and virtual variable(s). The modularization process continues until finishing analyzing the last virtual variable. Implementation of the modularization process is further demonstrated in Section 2.4.1.

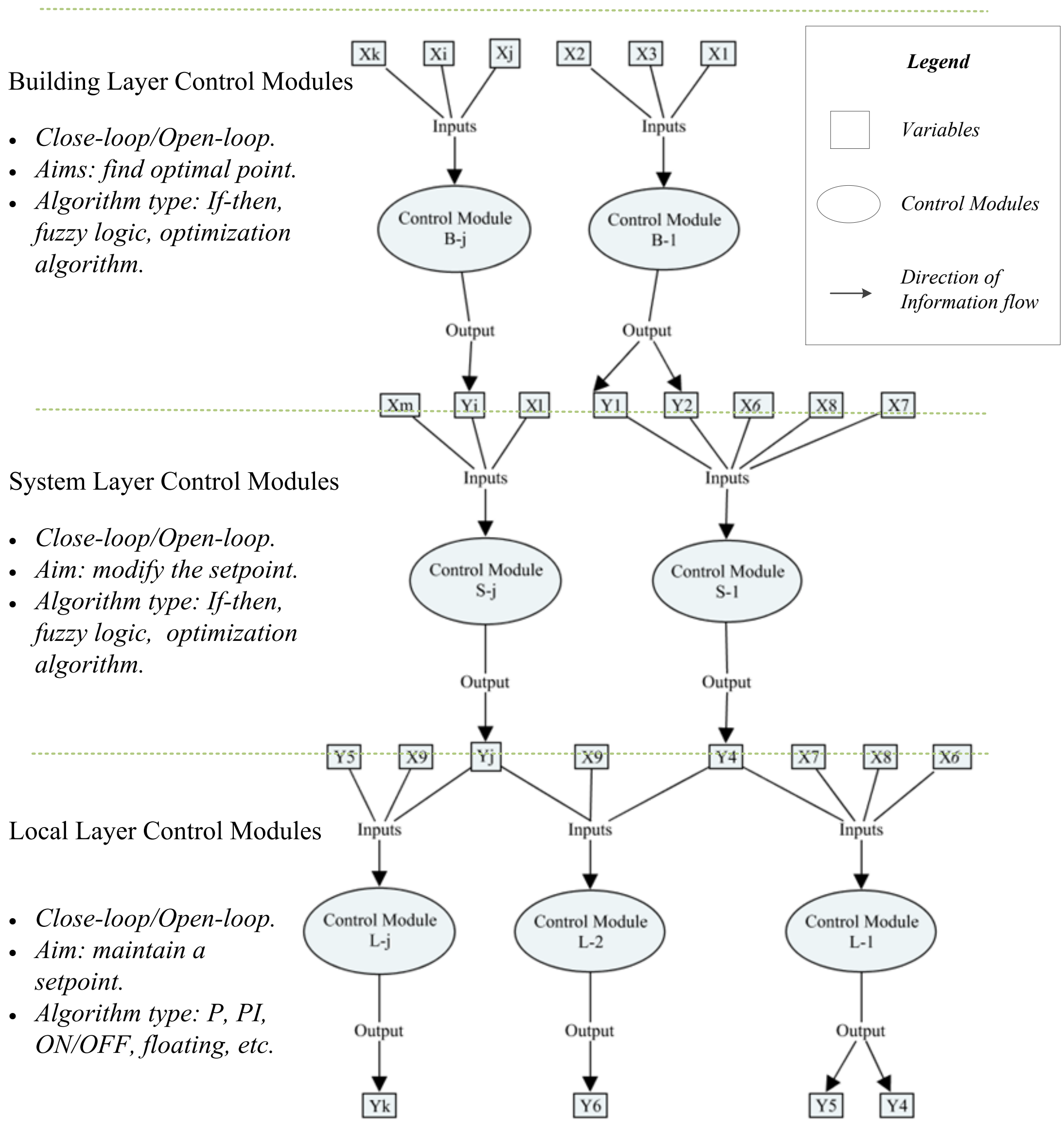


Figure 2-2 Ontology analysis of building control system structure.

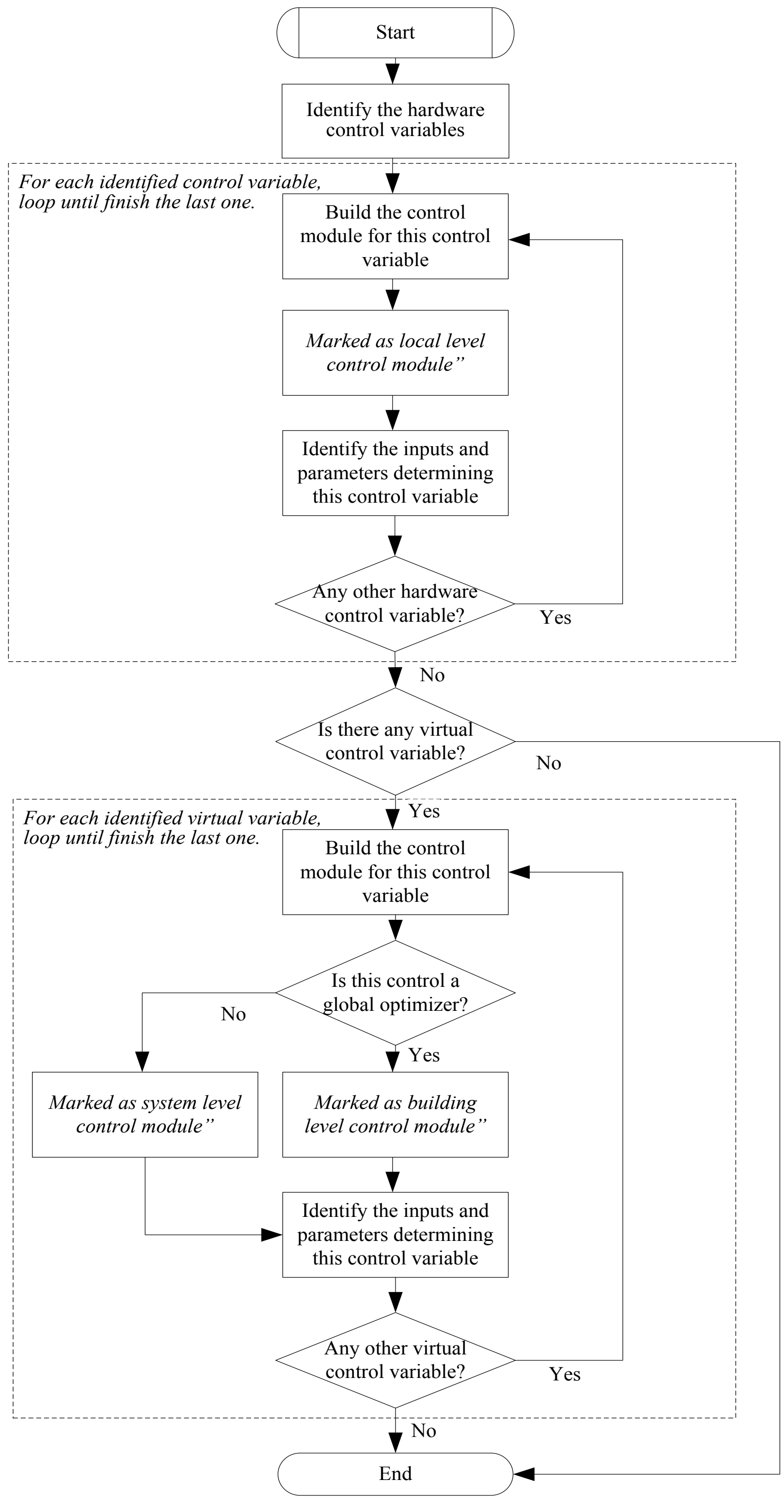


Figure 2-3 Control module classification process.

2.3.2 BCK Module Elements Identification and Formatting

After modularizing the whole BCK into a number of control modules, the next step is to determine the list of representing elements and their format for a typical control module. Based on the previous analysis of representing elements in existing resources (see Table 2-1), this section further discusses the reason of including/excluding each in the standard list and also defines the format (in Appendix A: Elements Formatting for A Typical Control Module).

2.3.2.1 Control Module Name

Control module name is an element given in most of the references by using the narrative name of the controlled components or systems. In ASHRAE (2005), the name was defined as a narrative name with special designation. This designation provided a unique ID, which can be helpful to store and manage related information in a database. Thus, the defined controller name includes narratives using the name controlled components, variables, or systems, (with the control functionality included) and a designated symbol.

2.3.2.2 Control Objective

Control objective element is used to help readers understand functionality of the control module. However, since the name of the module is given in both narratives and ID, the narratives have already briefly described the control objectives. Thus, the control objective is unnecessary to be included again.

2.3.2.3 Operation Mode

Operation mode element is used to describe the system operation for any given specified stage of operation. It helps the explanation and documentation of the control sequences by the operation mode. The format of Operation Mode was mentioned in the context of Control Sequences as a subtitle (or Clause Name). There are usually a list of typical operation modes, such as occupied, unoccupied, heating, cooling, optimal start, freeze protection, etc. However, those modes were either not well defined, or some of the operation modes might have overlaps (e.g., optimal start with heating). Thus, it might cause some ambiguity issues in practical understanding process. The format here follows the style in the *Library of system control strategies* by Martin and Banyard (1998), and defines a list of operation mode.

2.3.2.4 System Schematic

The system schematic element is usually included to explain the layout and connection of systems and components configuration. For the modularized BCK representation, it would be unnecessary to have separate system schematics for each of them. It is more meaningful to have some shared system schematics to illustrate the relationship and configuration at a higher layer. Thus, the System Schematic is included as one of the representing elements. The format of the drawings shall follow the ANSI/ASHRAE Standard 134 about the graphic symbols for HVAC systems (ASHRAE 2005).

2.3.2.5 Control Flow/Logic Diagram

The control flow diagram, also known as a control flow chart, is an element used to illustrate the systematic control decision process. It is more intuitive than narrative sequences, and this is usually helpful to assist programming and interpretation. It was recommended in the Guideline 13 to “use tables and diagrams where possible to assist in conveying sequence logic” (ASHRAE 2007). It was also suggested to be included for control sequence representation by Utterson (2006). However, only the *Library of control system strategies* included the control flow diagram (Martin and Banyard 1998). This study includes this control flow diagram as a necessary element. The format of the drawings follows the ANSI/ASHRAE Standard 134 about the graphic symbols for HVAC systems (ASHRAE 2005). The defined format follows the standard and adds a Symbol to represent it.

2.3.2.6 Control Sequence

The control sequence element is the narrative description of control knowledge without detailed calculation algorithms. The formats of control sequence can be typically classified as two types: by operation mode, or by components. Based on the reviewed documents, most of the control sequences had major sections organized by the components, and then each major section was organized by operation modes. Our format was using operation mode based structure within each control module, since the control modules have been organized by control functionalities and components through modularization. For the control sequence language, there were no standard languages. Researchers from computer science have studied and developed several formats to better represent general algorithm in narratives, including *pseudocode, prosecode, literate code* (Zobel 2004). Here *prosecode* is selected since it can concisely represent the control sequences with a structured way. Prototype and language for the control sequence is created (in Table 2-3).

Table 2-3 Control sequence organized by component output value vs by operation mode.

	By components output values	By operation mode
Rules	y_1 **SHALL** be $Y_1 \ldots y_m$ **SHALL** be Y_m a) **WHEN** x_1 is X_1 b) **OR WHEN** x_2 is X_2 **AND**,…**OR** x_n is X_n.	**IF** x_1 is X_1 **AND** (x_2 is X_2 …**OR** x_n is X_n), **THEN** y_1 is $Y_1 \ldots y_m$ is Y_m. a) **IF** **THEN**
	where x_1 stands for the operation mode variable x_2- x_n stand for the other conditions variables that need to satisfy when x_1 is enable. X_1, X_2, X_n, are the linguistic variables of x_1, x_2, x_n, respectively y_1 - y_m stand for control outputs Y_1- Y_m stand for control outputs values under this rule	
Usage	• Better guide for programming • Useful for trouble shooting and maintenance for specific components	• Generally easier to understand • Better explain operating concept • Highlight differences between the operating modes

2.3.2.7 Data Point

Data Point element is used to describe the BCK module related input, output, and parameter. All the listed documents (in Table 2-1) include this element with a standardized format. Thus, the format from the existing documents is adopted.

2.3.2.8 Alarm

Alarm element is used to describe the list of abnormal status that should be alerted to the system operators. The format of Alarm was in narratives within the context of Control sequences, as a separate section. The format can be extracted as Alarm Name: triggering condition. This format 1) added the symbol to represent the alarm in a programming language; 2) added “logic” to explain the triggering condition.

2.3.2.9 Function /Algorithm

Function /algorithm element is used to provide a detailed description of the logic/or algorithms used to determine control output. This element was recommended by Guideline 13 as “show the formulas in the sequences if they are to be used in calculations” (ASHRAE 2007), but few of the documents included this element. Having this element would also enable the explanation of new or advanced algorithms for further deployment. This might potentially put a few efforts to solve the deployment difficulty for advanced or optimal control strategies realized by a number of studies (Brambley, Haves et al. 2005; COOLTOOLS™ 2009). Meanwhile, O&M engineers preferr understandable control algorithms. Therefore, this element would likely provide the explanation that O&M engineers needed to understand the algorithm. Since algorithms/formulas were not mentioned in those documents, there was no format about the representation. The format for this element is defined to include the symbol, narratives, formula inputs, formula outputs, formula parameters, formula and references.

2.3.2.10Programming Code

Programming code is an element that represents the BCK in a programming language. Reviewed documents did not include this element except the EquipmentBuilder. The benefits of using this element included 1) there is future need for controller reconfiguration, and self-configuration as discussed in (Chen and Treado 2014), 2) if a standardized programming format is defined and provided, it would also decrease the programming errors. Therefore, this element was included in the lists of BCK module elements. The format of programming code generally depends on manufacture and programmer's style. There were three commonly-used types, namely, text-based, graphic-based, and menu-driven based (ASHRAE 2007). These three types are compared in Table 2-4.

Table 2-4 Comparison of programming types.

	Text-based	**Graphic-based**	**Menu-based**
Characteristic	Use high-level programming language (BASIC, C, Pascal)	Use graphic "functional blocks" for programming	Select control sequences pre-programmed by manufacturer
Flexibility	Highest	High	Low
Comprehensive	Highest	High	Low
Learning curve	Long (a few months)	Medium (a few days)	Short (a few hours)
Programming time	Medium	Long	Short

The first two types are more flexible and robust, while the menu-driven depends on pre-programmed sequences. In practice, each type is used by different manufactures (e.g., AutomatedLogic uses graphic-based in WebCtrl, Siemens uses text-based). Thus, the format of this programming code is not limited to any type, but depended on the users and manufactures, as long as it can satisfy the modularized programming requirement. For the testing purpose, the StateFlow block in Simulink is used to program the control sequences in this study.

Therefore, the full list of representing elements were identified as *control module name, control objective, operation mode, system schematic, control flow diagram, data point, alarm, control sequence, function, and programming code*.

2.4 Demonstration through an Example System

In this section, an example is used to demonstrate the BCK representation following the new approach. This example is based on the control system for single duct VAV multi-zone system. This type of system is common in commercial buildings and its major components include the AHU (heating coil, cooling coil, and supply fan), VAV terminal units (damper and reheating coil), and mixing box. The original control sequences and programming codes are obtained from the *EquipmentBuilder for Educators 5.5*, and used as raw data for the modularization and constructing the BCK representing elements.

2.4.1 Modularization

The modularization of the BCK for this example system followed the process described in Figure 3-1. Based on the given HVAC system and components, a list of hardware control variables were identified. Then, each hardware control variable was further analyzed using the ontology analysis (shown in Figure 2-2), which determines the related virtual control variable(s). In this example, the detailed analysis for the part of VAV terminal unit is presented in Figure 2-4 and the whole system ontology analysis is presented in Figure 2-5. As shown in Figure 2-4, in order to determine the VAV damper opening, the control module needed to know at least the value of four inputs/parameters, i.e., the minimum/maximum airflow rate setpoint, the reheat coil status, zone heating & cooling demand, manual override signal. Among them, the manual override signal and the reheat coil status were the sensor measurement or direct manual inputs, while the min/max airflow setpoints, zone heating & cooling demand were virtual control variable depending on another set of inputs/parameters. Therefore, zone heating & cooling demand was determined based on the actual zone temperature and setpoint. In addition, the zone setpoint was determined based on different operation model. Thus, for the VAV terminal units, the one hardware control variable (VAV damper) depended on at least three virtual control variables (i.e., zone heating & cooling demand, VAV min/max airflow setpoint, and zone temperature setpoint).

Using the same method, seven hardware control variables, and eighteen virtual control variables are identified (shown in Figure 2-5). Each control variable became a foundation for further constructing the control module packaging the associated inputs, parameters, logic, operation mode, sequence, control flow diagram, formula, etc. In this way, the whole BCK was modularized into 25 individual functional-independent yet strongly related control modules.

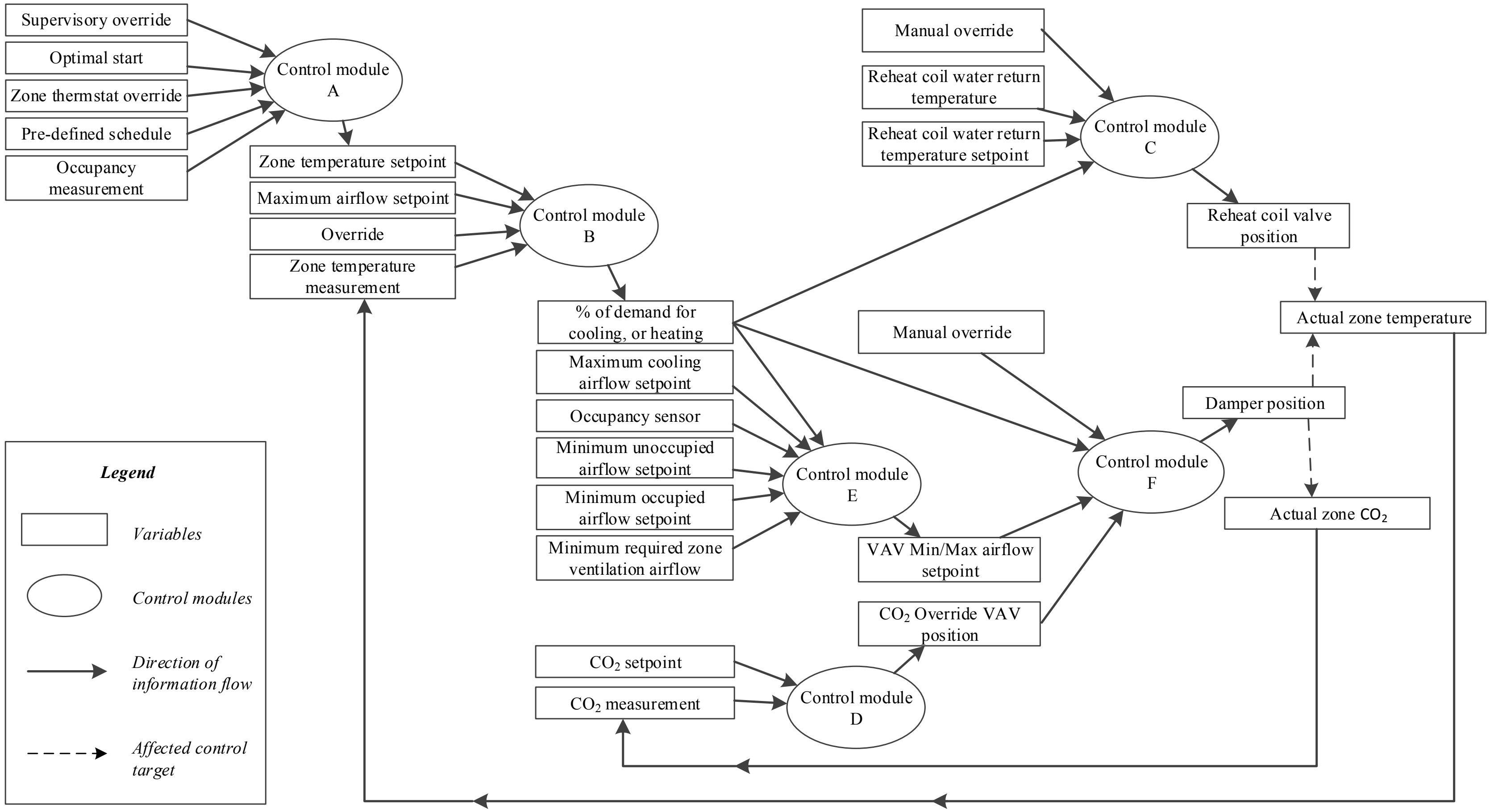


Figure 2-4 Detailed ontology analysis for the VAV box terminal unit (part of the whole system).

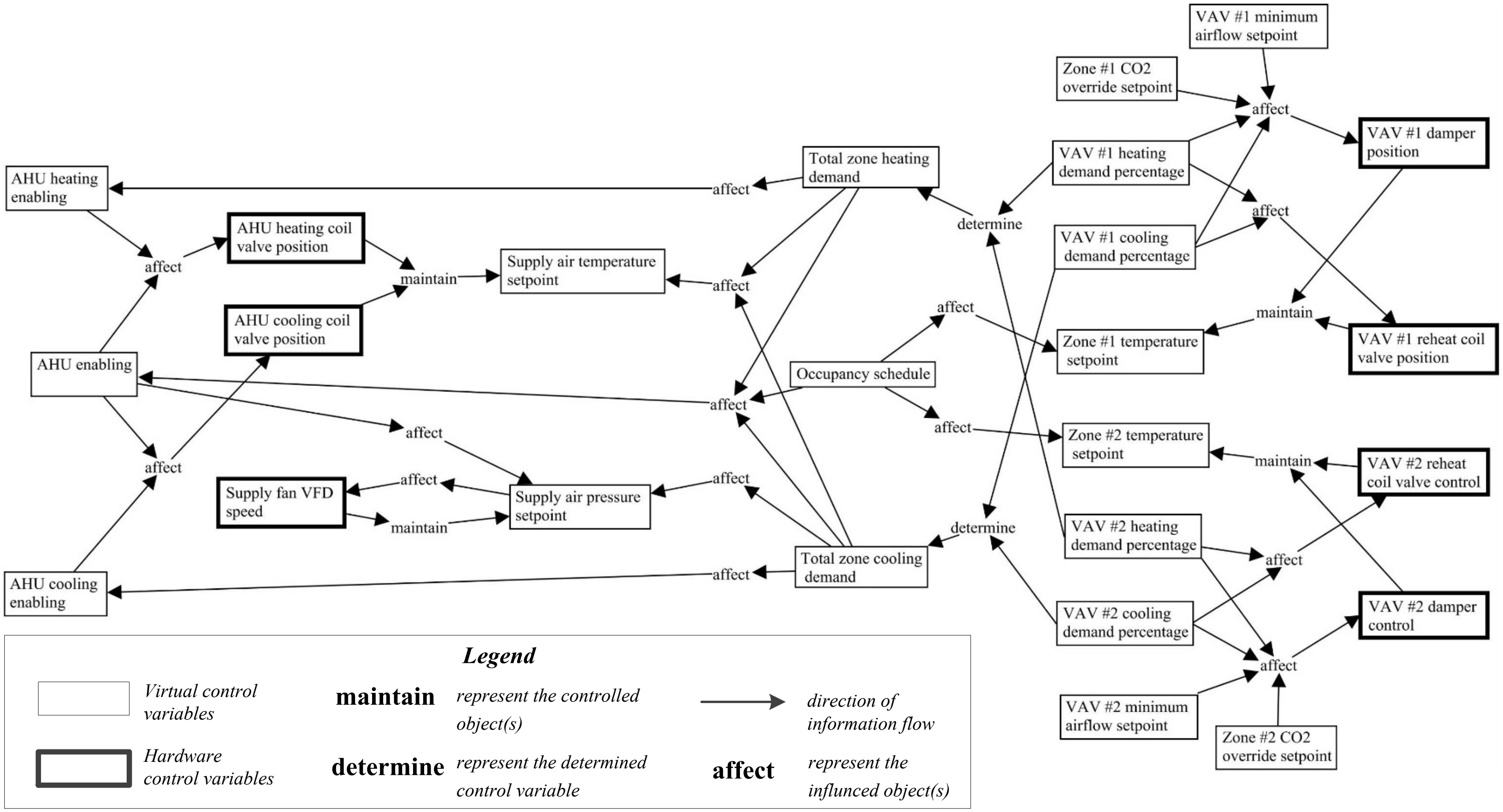


Figure 2-5 Ontology analysis of the control variables for the single duct VAV multi-zone system (the whole system).

2.4.2 Construction of the Representing Elements for Control Modules

After modularization, the format guideline (table in Appendix A: Elements Formatting for A Typical Control Module) is followed to construct the BCK representing elements for each control module. Each module has its own content, while sharing the same structure (list of representing elements) and formatting style. A control module with formatting in details is presented in Appendix B: Example-A VAV Box with Reheat Control Module. As shown in the diagram, *VAV #1 damper control module* (L_Z1_VAV_D) was selected to represent the typical control modules.

In *VAV #1 damper control module*, this *Module Name* was designated as L-Z1-VAV_D, which indicated as a local control module for Z1 VAV damper position control. The *Control Objective* indicated this control module is intended to maintain the zone temperature setpoint. In the *Operation Mode* section, there were five operation modes. In the *Object List* section, it indicated that there are five input variables and one output variable. The *Alarm* section indicated there are four alarms associated with this control module. In the *Sequence* section, the sequence is organized by the five operation modes, and it can be further interpreted with the *Control flow diagram* and *Programming code*. In the *Formula* section, two calculation formulas were identified and represented explicitly.

The other twenty-four control modules were represented using the same format. In this way, by applying the modularization process and formatting guide, the BCK for the example HVAC system was modularized and reformatted into twenty-five control modules. Each of them has been packaged as a functional independent module. All together, they have realized the functionality of heating, cooling and ventilation with the choice of optimization for different performance measures.

2.5 Comparison and Discussion

From this demonstration example, it can be seen that the new approach for BCK documentation has a number of new features that are different from the existing practices (Table 2-5).

Table 2-5 Summary of comparing the new approach to the existing ones.

Existing Approaches	The New Approach
• Different engineers and manufacturers used different lists of elements to represent BCK based on their own expertise	• Provided a guideline that engineers and manufacturers can follow to represent BCK
• Used different formats in different BCK documentations	• Used standard format for BCK representing elements
• Don't have a high level relationship analysis	• Provide clear ontology analysis at the high level of modularization
• Usually don't have control flow diagram	• Provided control flow diagram
• Organized by major devices at the high level, and then with mixed used of operation mode and component formats at lower level. The modulation rules and results were not explicit.	• Organized by major systems at the high level, then, modularized into a number of control modules with independent functionality at the lower level. Provided a standard process to simplify the complexity of larger control system BCK representation.
• Usually don't have detailed description of the calculation formula	• Constructed a function/algorithm element to represent the calculation formula
• Advanced algorithm is always represented as a black box	• Provides a space to represent the advanced control or algorithms

These features provide several benefits for the BCK documentation. First, the control variable based modularization method simplified the big and complicated BCK representation via an explicit process that can be standardized. Although the existing practices also divided the BCK into different elements, there was a lack of a standard modularization process, and different engineers/manufactures had done it differently. Second, the functional independent control module is easier for analysis and diagnosis. Because each module is constructed based on one control variable with its own control information, and would be convenient to build and verify the sequences, logic, programming codes, etc. Third, the newly added representing elements for formula and algorithm provided a platform to describe the calculation in details. For a long time, this formula, algorithm and detailed calculation explanation were usually ignored in the existing practices. This disconnection between algorithm designers and users has created difficulties for building operators/commissioners to understand the logic and to diagnose faults. Therefore, this new element for formula/algorithm not only reconnects the algorithm designers with users, but also creates a platform for deploying new algorithms in the field. Using this element, users can easily identify the differences between the baseline algorithm and the new one for achieving the same control functionality.

Another benefit of using a modular approach is that it may support more distributed control algorithm development and enable self-configuration and fault diagnostics and detection based on modularized control. For example, it could support the development of a new building control self-configuration that depended on a database that stores generalized BCK modules. Chen and Treado (2014) envisioned this type of new self-configuration control system will be different from conventional self-configuration control systems (such as the work by Ryan and Shah (2007), Wruck (2008), Baldwin, Bishop et al. (1994)). While conventional self-configuration control systems have been based on preinstalled fixed control algorithms to determine the control outputs, the new ones would interact with a BCK module database to select/update the applicable control algorithms for a specific system. In this way, the self-configuration process will not be limited by fixed control algorithms, but instead provide a wider application range and be easier to manage and update.

Meanwhile, this BCK data schema developed in this work can potentially contribute to the IFC BAMie. This data schema may be adopted under the Domain schemas -> Building

Control Domain -> IfcBuildingControlsDomain schema, serving as a new entity that document the control sequences, algorithms and logic for determining control outputs. In this way, it could expand the application of IFC Building Automation Modeling for a wider application range at the operational and maintenance stage (e.g., use these information for retrofitting control system, FDD, and on-going commissioning). This data schema may also be integrated into the BACnet Control Device modeling section as a set of new object types to enhance the representation of control knowledge that determine control outputs. This could enrich the information exchange among control devices and benefit applications and services that are based on this information (such as FDD, optimization, and performance analysis).

In summary, this chapter presented a new approach to represent generic BCK. A modularization process has been developed based on individual control variables (both hardware and virtual). A process map has been provided to guide the modularization process. For control module, a list of elements has been identified to represent the modularized BCK. Standard formats have been developed for each element. Lastly, the implementation of the new approach has been demonstrated using a single duct VAV multi-zone system. The demonstration showed an explicit representation of BCK for control module, which can be used as a guideline for industry practices. The proposed method provides clear representation of BCK comparing to the existing methods. It reduces the format inconsistence and ambiguity, which would potentially benefit the building control system design, implementation and operation maintenance processes. Meanwhile, the modularization also provided easier system configuration and re-configuration (by using plug-and-play features). In this way, individual control modules can be easily reconfigured or upgraded without affecting other modules. This will be further explored in the integrated BCK database study in Chapter 3, and the self-configuration process development in Chapter 4.

CHAPTER 3

BUILDING CONTROL KNOWLEDGE DATABASE - TOWARDS A UNIFIED MANAGEMENT OF CONTROL KNOWLEDGE

The previous chapter discussed and identified a list of representing elements for BCK. Based on this result, this chapter focuses on implementing the BCK data schema into a BCK database. The BCK database not only servers as the storage house of BCK for different HVAC system types, equipment and components (discussed in Chapter 2), but also acts as resource center for the self-configuration process (discussed in Chapter 4). Microsoft Access 2013 was used as the main tool for the first version database development. Meanwhile, a user-friendly interface has been created to support accessing the information and adding new data.

3.1 Introduction

Building control systems, which play a key role in delivering thermal comfort, adequate IAQ and achieving higher energy efficiency, function based on a collection of building control knowledge (i.e., control logic, sequences, algorithms, and related data points). This type of knowledge has been widely used in building control system design (with or without minor modifications). This high reusability has motivated the building industry and researchers to develop resource centers or platforms that can support systematic collections of building control knowledge.

Currently, there are a few public accessible resources, which provide collections of typical building control knowledge. One type is an internet-based resource center. An example is the CtrlSpecBuilder developed by (AutomatedLogic 2012). It is a web-based platform, which can generate narrative control sequences. This platform has been used by many control designers to specify control sequence for HVAC system. It allows users to specify system and equipment types and then generate the templates for narrative control sequence, data point list, and programming code (from *EquipmentBuilder*). However, this program does not allow users to enter new knowledge (i.e., control logic, sequences, and algorithms) through the interface. Another example is handbook, organizational guidelines for building control sequences/strategies (e.g., *Library of system control strategies* wrote by Martin and Banyard (1998), *The sequence of operation* CD published by ASHRAE (2005), and *Advanced VAV System Guideline* developed by Hydeman, Taylor et al. (2003)). However, these resources and platforms are facing several application challenges, lack of interactive interface, limited description of the calculation algorithms, and lack of support for the advanced algorithm deployment. They also use different styles and formats to represent the BCK, which further discourages the sharing and accumulation of building control knowledge.

On the other side, a number of advanced control strategies and algorithms have been developed for achieving higher performance. Huang and Lam (1997) introduced an adaptive learning algorithm based on genetic algorithms (GA) for automatic tuning of PID controllers in HVAC systems. Soyguder, Karakose et al. (2009) introduced a method with PID-type adaptive control, with self-tuning of PID parameters based on fuzzy logic rules for different error and error change rates. Distributed model-based predictive control (MPC) was used to control building temperature based on predicted occupancy and weather (Moroşan, Bourdais et al. 2010; Ma, Anderson et al. 2011). A fuzzy logic based coordinator-agent hierarchical control strategy was introduced for managing the users' preferences for thermal and illuminance comfort, indoor air quality and the energy conservation (Dounis, Tiropanis et al. 2011). Kusiak and Xu (2012) optimized a HVAC system based on a dynamic neural network. These advanced algorithms need to be deployed in real buildings. However, few of the advanced algorithms can be found in commercial control manufacturers' references. Most of these references only have the narrative control sequences and limited descriptions of algorithms and calculation formula. In some "fortunate" cases, an advanced algorithm was deployed in the commercial system, but explicit documentation of the calculation algorithms was missing. Thus, as realized by some researchers, deployment difficulty exists for advanced or optimal algorithms (Brambley, Haves et al. 2005; COOLTOOLS™ 2009).

Therefore, the building control industry is demanding an interactive platform and resource centers that are capable of not only storing and managing the existing building control knowledge, but also adding new ones. The kind of platform would benefit the reusability of building control knowledge and facilitate the deployment of newly developed advanced ones.

To achieve this goal, there were a few exploratory efforts. Schein (2007) developed an information model for building automation systems, and experimentally implemented for office building. This study also recommended investigating a software tool to parse the information model and then creating the configuration database and control application programs. However, this study did not provide detailed discussion about how to represent building control knowledge (e.g., sequences, operation mode, and algorithms. Akinci, Garrett et al. (2011) identified a list of functional requirements for integrated building control system information model based on the survey/workshop from experienced engineers and researchers. However, to author's best knowledge, there has not been mush follow-up researches in this field since 2011.

In order to extend this line of efforts, Chapter 2 of this work has analyzed the structural commons shared in building control knowledge and provided the modularization process as a foundation to develop the BCK database. Important commons and structures have been discussed in Chapter 2, including modularization of BCK representation into a number of modules based on control variables, and identification of the representation elements for typical control modules. This chapter thus focuses on the development of the BCK database. Specifically, this chapter first provides an overview of the BCK database design (in Section 3.2). Then, a detailed data model is presented in Section 3.3, and is implemented in database software in Section 3.4. Finally, a case demonstration is presented.

3.2 An Overview of Database Design

This overview section has two objectives: 1) discuss database functional requirements, and 2) analyze the data model of BCK and database type.

3.2.1 Database Functional Requirement

A list of functional requirements for integrated building control system information model had been identified through the work of Akinci, Garrett et al. (2011). The list includes, #1 representation of HVAC hardware components, control logic and algorithm, operational fault conditions, #2 information on climate, occupant behavior, #3 definition of data exchange framework, and metrics of system performance evaluation, and #4 capability of extension, upgrade, and adaption. Among then, Requirement #2 and #3 have been addressed by the BACnet Standard and other studies, such as the work by Schein (2007). Requirements #1 and #4 have not been explored. This work addresses requirement #1 and #4, with a consideration to support self-configuration of building control systems.

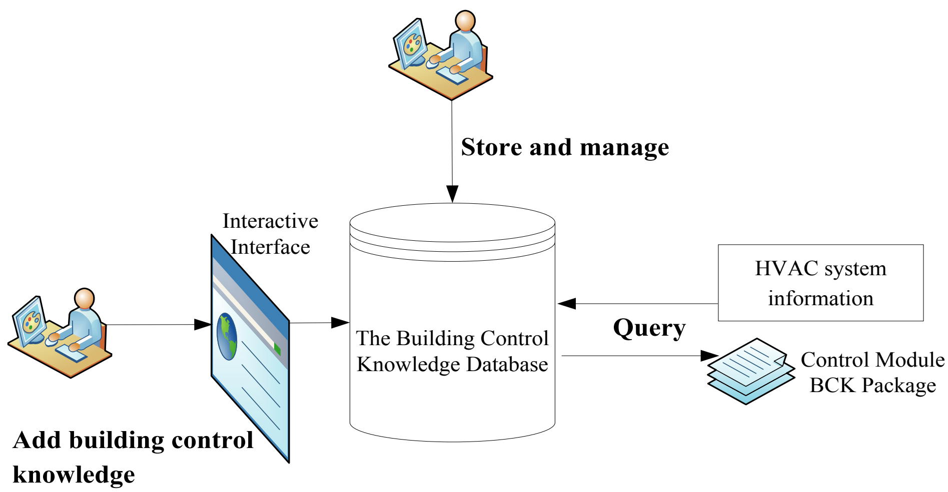


Figure 3-1 Functional requirement overview for the building control knowledge database.

Thus, as illustrated in Figure 3-1, initial consideration of the functional requirements for this database development include: 1) to serve as a database to store and manage building control knowledge; 2) to provide an interactive platform to support adding building control knowledge from previous projects or their expertise; and 3) to provide queries that can select and create specific building control knowledge to support self-configuration of building control system.

3.2.2 Data Model

As explained by Ma (2005), "Database modeling generally starts from the conceptual data models and then the developed conceptual data models are mapped into the logical database models". Therefore, based on concept of control module for BCK in Chapter 4, this chapter extended this concept and conducted ontology analysis using the entity-relationship (ER) modeling method by Chen (1976).

3.2.2.1 Entity Relationship Modeling

Based on the representing elements and formats identified in Chapter 4, a number of classes are defined, including control module, system schematics, flow diagram, sequence, function, data point and alarm. Each class is a template to define the formats of related data.

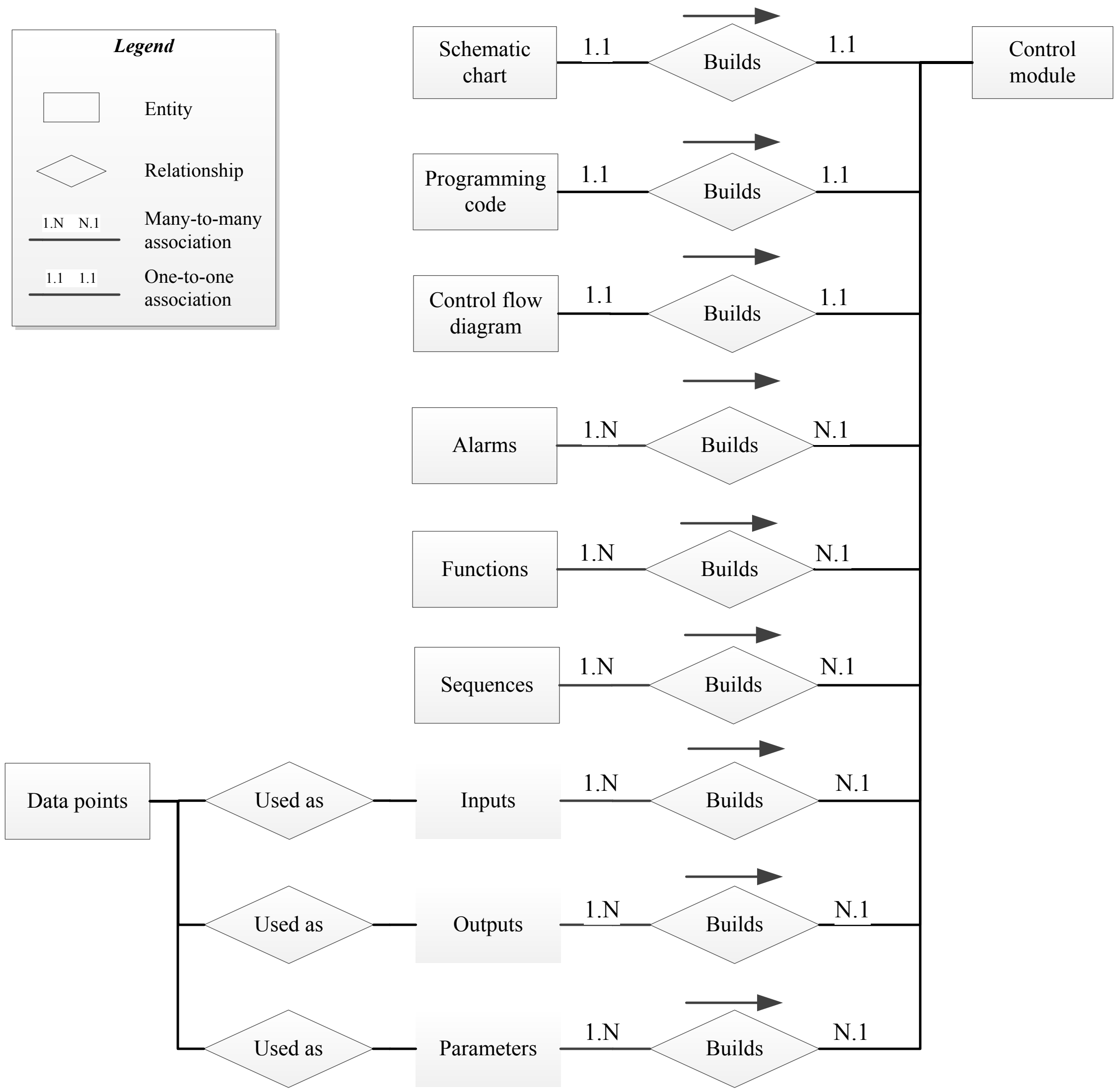


Figure 3-2 Entity relational diagram of a typical control module.

As illustrated in Figure 3-2, the control module is considered as a class that acts as a higher-level mask. It is composed with a number of sub classes, including schematic chart, programming code, control flow diagram, sequences, inputs, outputs, parameters, alarms, and functions. Each control module has one of schematic chart, programming code, and control flow diagram; and each module has one, or more than one of sequences, inputs, outputs, parameters, alarms, and functions. Thus, two types of association are designed between a control module object and its sub objects, i.e., one-to-one association (1.1-1.1), and many-to-many association (1.N-N.1). The detail of each class / subclass is discussed in Section 5.3.

3.2.2.2 Database Type Selection

To represent the data characteristics mentioned above, a reasonable database model type should be selected. Typical database type include flat files, relational databases, spreadsheets, hierarchical databases, XML, network database, object database, object-relational database, etc. (Stephens 2010). Object-relational database (ORDB) was selected for this database development because of a number of reasons. First, the database needs to perform complicated relational-style queries. Second, the programming environment and architecture favors using objects. Third, the database needs to perform relational-style data validation. Lastly, the object-oriented programing is more common for control sequence design, and the important of relationship among objects is obviously.

3.3 ORDB Data Model in Details

The ORDB data model is illustrated in this section. Three types of schema, namely, *Classes, Object relational tables*, and *List tables* are used to manage different types of data, and their relationships.

3.3.1 Objects Classes

This part explains the ORDB Classes in details. The definition of a Class serves as the template for how to specify an object under this class.

3.3.1.1 Control Module Class

Control Module Class is defined as a modularized shell that is in composed with a set of sub classes. Those subclasses including data point, alarms, sequences, functions, system schematics, control flow diagrams, and codes. Since those subclasses may be also associated to other control modules, the relationships between control module and these subclasses are "many-to-many association". This type of relationship was realized through objects relational tables (e.g., Input List). Thus, the attributes of the control module class include module ID, module name, control module type, building type, schematic ID, flow diagram ID, code ID, created date, and objects relational tables (i.e., input list, output list, parameter list, function list, sequence list, and alarm list)(see Table 3-1).

Table 3-1Control module classes and profile attributes.

Attribute	Data Type	Req'd?	Description
Module ID (CtrlID)	Short Text	Yes	Unique ID for this controller in format XX-XX-XX
Module Name (CtrlName)	Short Text	Yes	Narrative name defines the controller.
Controlled System (ControlledSys)	Short Text	No	The system controlled by this controller
Controlled Equipment (ControlledEquip)	Short Text	No	The equipment controlled by this controller
Control Module Type (CtrlType)	Short Text	Yes	The level that this controlled functioned
Schematic ID (SchematicID)	Short Text	Yes	Unique ID to link to the schematic in format XX-XX-XX

Flow Diagram ID (FlowDigramID)	Short Text	Yes	Unique ID to link to the flow diagram in format XX-XX-XX
Code (CodeID)	Short Text	Yes	Unique ID to link to the code file in format XX-XX-XX
Created Date (Date)	Date/Time	Yes	The date that this controller is created in format MM/DD/YYYY
Sequence List	N/A	Yes	See Relational Table: Controllers-sequences
Input List	N/A	Yes	See Relational Table: Controllers-inputs-data
Output List	N/A	Yes	See Relational Table: Controllers-outputs-data
Parameter List	N/A	Yes	See Relational Table: Controllers-parameter-data
Function List	N/A	Yes	See Relational Table: Controllers-functions
Alarm List	N/A	Yes	See Relational Table: Controllers-alarms

3.3.1.2 System Schematic Class

The System Schematic Class is defined as a class to store and manage the schematics of the related equipment, components, and systems.

Table 3-2 System Schematic classes and profile attributes.

Attribute	**Data Type**	**Req'd?**	**Description**
Schematic ID (SchematicID)	Short Text	Yes	Unique ID for this schematic in format XX-XX-XX
Schematic Name (CtrlName)	Short Text	Yes	Narrative name defines the schematic
File Location (FileLocation)	Attachment	Yes	Location where the schematic file is saved
Model Number (ModelNum)	Short Text	No	The model number for this system
Manufacturer (Manufacturer)	Short Text	No	The name of manufacturer for this system

3.3.1.3 Control Flow Diagram Class

The Control Flow Diagram Class is defined as a class to store and manage the schematics of the related equipment and systems.

Table 3-3 Control Flow Diagram classes and profile attributes.

Attribute	Data Type	Req'd?	Description
Flow Diagram ID (FlowDiagramID)	Short Text	Yes	Unique ID for this flow diagram in format XX-XX-XX
Diagram Name (DiagramName)	Short Text	Yes	Narrative name defines the flow diagram
File Location (FileLocation)	Attachment	Yes	Location where the flow diagram is saved

3.3.1.4 Programming Code Class

The Programming Code Class is defined as a class to store and manage the programming code for the control module.

Table 3-4 Control Flow Diagram classes and profile attributes.

Attribute	Data Type	Req'd?	Description
Programming Code ID (CodeID)	Short Text	Yes	Unique ID for this flow diagram in format XX-XX-XX
Code Name (CodeName)	Short Text	Yes	Narrative name for the programming code
File Location (FileLocation)	Attachment	Yes	Location where the code is saved

3.3.1.5 Function (algorithm) Class

Function (algorithm) Class is defined as a class that incorporates with mathematical functions (formulas or complex logic reasoning). This class aims to store and manage existing practical algorithms as well as the newly developed advanced control algorithm (e.g., model-predictive control algorithm, fuzzy logic control algorithm). It processes the signals from inputs, and determines the values of outputs using the calculation formula. Attributes of this class include ID, description, formula, inputs, outputs, parameters, and references. For one control variable (output), there could be more than one algorithms to realize the same control function. For example, one algorithm is supply air temperature setpoint determined by MPC, while another one algorithm is supply air temperature setpoint determined by zone thermal load directly. However, only one function should be selected in actual implementation.

Table 3-5 Function classes and profile attributes.

Attribute	Data Type	Req'd?	Description
Function ID (FcnID)	Short Text	Yes	Unique ID for the function in format XX-XX-XX
Function Name (FcnName)	Short Text	Yes	Narrative name of this function
Description (FctDescription)	Long Text	Yes	Briefly describe the function/algorithm to help users understand the calculation and functionality
Formulas (FcnFormulas)	Long Text	Yes	All calculation formulas in this function
Function Inputs (FcnInputs)	Long Text	Yes	Inputs for this function
Function Outputs (FcnOutputs)	Long Text	Yes	Outputs from this function
Parameters (FcnParameters)	Long Text	No	Parameters for this function
References (FcnReferences)	Long Text	No	Literature reference for this function

3.3.1.6 Data Point Class

Data Point Class is defined as a class to manage and store all the data points that can be used as inputs, outputs, and parameters for control module. A prototype attributes of each information node is listed in Table 3-6.

Table 3-6 Function classes and profile attributes.

Attribute	Data Type	Req'd?	Description
Data ID (DataID)	Short Text	Yes	Unique ID for the data point in format XX-XX-XX

Data Name (DataName)	Short Text	Yes	Narrative name defines the data point
Symbol	Short Text	Yes	Unique symbol to abbreviate the Data Name in the format XX-X-XX-XX
Unit	Short Text	Yes	Variable Unit
Source Type (SourceType)	Short Text	Yes	Refer to the DataSourceType Table
Lower Limit (LowerLimit)	Number	No	Lower limit value for this data point
Upper Limit (UpperLimit)	Number	No	Upper limit value for this data point
Location	Short Text	No	Related physical location in the system
Accuracy	Short Text	No	Accuracy only required for measurement

3.3.1.7 Sequence Class

Sequence Class is defined as a class that encapsulates the narrative description of control knowledge (generally called "operation of sequence", or "control sequences"). Sequence describes the order in which things happen or should happen; a group of things that come one after the other (Webster). As discussed in Chapter 4, those narratives are extremely important for programming as well as maintenance activities. The attributes of this class is designed to include sequence ID, name, operation mode, and the narratives (Table 3-7).

Table 3-7 Sequence class and profile attributes.

Attribute	Data Type	Req'd?	Description
Sequence ID (SequenceID)	Short Text	Yes	Unique ID for the sequence in format XX-XX-XX
Sequence Name (SequenceName)	Short Text	Yes	Narrative name defines the sequence
Mode	Short Text	Yes	Unique symbol to abbreviate the Data Name in the format XX-X-XX-XX
Narratives	Long Text	Yes	Narratives for the sequence of operation should be organized by the operation mode.

3.3.1.8 Alarm Class

Alarm Class is defined as a class to store and manage the building system alarms. These include the alarms for the equipment and system operations, as well as building level security. One control module can have multiple alarms. A multiple-to-multiple associate is existed between controller and alarm classes. This is realized through the controller-alarm relational table. An attribute of the alarm class is listed in Table 3-8.

Table 3-8 Alarm class and profile attributes.

Attribute	Data Type	Req'd?	Description
Alarm ID (AlarmID)	Short Text	Yes	Unique ID for the sequence in format XX-XX-XX
Alarm Name (AlarmName)	Short Text	Yes	Narrative name defines the sequence
Narratives	Long Text	Yes	Narratives describe the operation of alarm

3.3.1.9 HVAC System Class

HVAC System Class is used to categorize control module based on the application in different system types. In general, lists of system and equipment between different HVAC systems overlap. This indicates that one control module may be applicable to multiple HVAC system types. However, the classification of HVAC system itself would be another study, because there is a lack of explicit classification for HVAC system types.

Table 3-9 HVAC System class and profile attributes.

Attribute	Data Type	Req'd?	Description
HVAC Type ID (hvacID)	Short Text	Yes	Unique ID for this controller in format XX-XX-XX
Name (Name)	Short Text	Yes	Narrative name describes the type.
Description	Short Text	Yes	Briefly description of the system type

Heating Type	Short Text	No	Description of the system heating type
Cooling Type	Short Text	No	Description of the system cooling type

3.3.1.10 Equipment Class

Equipment class is defined as to document the specific HVAC system components. A relationship table is built up to record the association between equipment and its related control modules.

Table 3-10 Equipment class and profile attributes.

Attribute	Data Type	Req'd?	Description
Equipment (EquipID)	Short Text	Yes	Unique ID for this controller in format XX-XX-XX
Name (ObjectName)	Short Text	Yes	Narrative name describes the type.
Description	Short Text	Yes	Briefly description of the system type
Manufacturer	Short Text	No	The name of manufacturer for this system
Model	Short Text	No	The model number for this system

3.3.2 Objects Relational Tables

The objects relational tables have been created to represent the many-to-many association between classes (objects). This database includes a set of relational tables: Control Module-Inputs-Data Points, Control Module-Outputs-Data Points, Control Module-Parameters-Data Points, Control Module-Sequences, Control Module-Functions, Control Module-Alarms, and HVAC System Type-Control Module. Each relational table has a reserved field, which is designed for sorting or filtering the relationships based on other criteria in the future (e.g., optional, mandatory, advanced control).

3.3.2.1 Relational Table: Control Module-Inputs-Data Points

Table 3-11 lists the attributes for the Relational Table: Control Module-Inputs-Data Points. CtrlID and data point ID are served as the foreign keys, and these IDs are obtained by looking up the primary key value (IDs) in the control module objects and data points objects.

Table 3-11 Control Module-Inputs-Data Points.

Attribute	Data Type	Req'd?	Description
CtrlID	Short Text	Yes	"Lookup" referenced to CtrlID in Controller class.
Reserved Field	Short Text	No	For future development
DataName	Short Text	Yes	"Lookup" referenced to DataName in Data Points class

3.3.2.2 Relational Table: Control Module-Outputs-Data Points

Table 3-12 lists the attributes for the relational table: Control Module-Outputs-Data Points. CtrlID and data point ID are served as the foreign keys, and their values are obtained by looking up the primary key value (IDs) in the control module objects and data point objects.

Table 3-12 Control Module-Outputs-Data Points.

Attribute	Data Type	Req'd?	Description
CtrlID	Short Text	Yes	"Lookup" referenced to CtrlID in Controller class.
Reserved Field	Short Text	No	For future development
DataName	Short Text	Yes	"Lookup" referenced to DataName in Data Points class

3.3.2.3 Relational Table: Control Module-Parameters-Data Points

Table 3-13 lists the attributes for the Relational Table: Control Module-Parameters-Data Points. CtrlID and data point ID are served as the foreign keys, and their values are obtained by looking up the primary key value (IDs) in the control module objects and data point objects.

Table 3-13 Control Module-Parameters-Data Points.

Attribute	Data Type	Req'd?	Description
CtrlID	Short Text	Yes	"Lookup" referenced to CtrlID in Controller class.
Reserved Field	Short Text	No	For future development
DataName	Short Text	Yes	"Lookup" referenced to DataName in Data Points class

3.3.2.4 Relational Table: Control Module-Sequences

Table 3-14 lists the attributes for the Relational Table: Control Module-Sequences. CtrlID and SequenceID are served as the foreign keys, and their values are obtained by looking up the primary key value in the control module objects and sequences objects.

Table 3-14 Control Module-Sequences.

Attribute	Data Type	Req'd?	Description
CtrlID	Short Text	Yes	"Lookup" referenced to CtrlID in Controller class.
Reserved Field	Short Text	No	For future development
SequenceID	Short Text	Yes	"Lookup" referenced to SequenceID in Sequences class.

3.3.2.5 Relational Table: Control Module-Functions

Table 3-15 lists the attributes for the Relational Table: Control Module-Functions. CtrlID and Function ID are served as the foreign keys, and their values are obtained by looking up the primary key value in the controller, data points classes separately.

Table 3-15 Control Module-Functions.

Attribute	Data Type	Req'd?	Description
CtrlID	Short Text	Yes	"Lookup" referenced to CtrlID in Controller class.
Reserved Field	Short Text	No	For future development
FunctionID	Short Text	Yes	"Lookup" referenced to FcnID in Functions class.

3.3.2.6 Relational Table: Control Module-Alarms

Table 3-16 lists the attributes for the Relational Table: Control Module-Alarm. CtrlID and Alarm Name are served as the foreign keys, and their values are obtained by looking up the primary key value in the Control module objects (ID) and alarm objects (ID).

Table 3-16 Control Module-Alarm.

Attribute	Data Type	Req'd?	Description
CtrlID	Short Text	Yes	"Lookup" referenced to CtrlID in Controller class.
Reserved Field	Short Text	No	For future development
AlarmID	Short Text	Yes	"Lookup" referenced to AlarmID in Alarms class.

3.3.2.7 Relational Table: HVAC System Type-Control Module

Table 3-17 lists the attributes for the Relational Table: HVAC System Type-Control Module. CtrlID and Alarm Name are served as the foreign keys, and their values are obtained by looking up the primary key value in the Control module objects (ID) and alarm objects (ID).

Table 3-17 HVAC System Type-Control Module.

Attribute	Data Type	Req'd?	Description
HVAC Type ID (hvacID)	Short Text	Yes	Unique ID for this controller in format XX-XX-XX
Reserved Field	Short Text	No	For future development
CtrlID	Short Text	Yes	"Lookup" referenced to CtrlID in Controller class.

3.3.3 List Tables

As mentioned in the Section 3.3.1 Object Classes, some of the attributes are linked to lookup tables (the list tables). Here a few list tables are preliminary defined, including control module type list, and operation mode list.

3.3.3.1 List: Control module type

The type of control module is defined the type of the role that a control module plays in the overall building control hierarchy. Table 3-18 gives a list of attributes for this controller type class.

Table 3-18 Control module type attributes.

Attribute	Data Type	Req'd?	Definition
ID	AutoNumber	Yes	Unique ID automatically generated for each operation mode type

Control Type	Short Text	Yes	Name of the control type
Description	Long Text	No	Detailed description of the operation mode

3.3.3.2 List: Operation Mode

Operation mode describes the major conditions and functionalities that an equipment or system operates. A building usually has a list of operation modes, and its systems and equipment will cooperate to achieve the functional requirement at different modes. Thus, control sequences also match with operation modes. Thus, an operation mode list is created to assist identifying the "mode(s)" for each sequence.

Table 3-19 Operation mode attributes.

Attribute	Data Type	Req'd?	Definition
ID	AutoNumber	Yes	Unique ID automatically generated for each operation mode type
Operation mode (mode)	Short Text	Yes	Name of the operation mode
Description	Long Text	No	Detailed description of the operation mode

3.4 The Implementation of the Database Using Microsoft Access 2013

With the developed data model from previous section, this section discusses the implementation of the data model into a database software. This study uses the Microsoft Access 2013 as a preliminary database development tool. The following content explains the implementation of Class/Relational/List Tables, and the development of user interfaces.

3.4.1 The Relationship - Class/Relational/List Tables

Class tables, Relational tables, and List tables have been programmed in the MS Access 2013. Their relationships shown in Figure 3-3 indicate their complex associations (one-to-one, many-to-many, one-to-many). Each class table has a defined primary key. Among those keys, the primary key of Control module's ID (CtrlID) has been frequently referenced.

3.4.2 Data Entering Interfaces and Algorithm

For this version of database, several interfaces have been created to realize the functional requirements for this database (discussed in Section 3.2.1). The design of interface followed the "Three Click Rule" used in web design (Zeldman 2001) to ensure the effective navigation. Among these interfaces, the control module interface (shown in Figure 3-4) serves as the main interface for data entering.

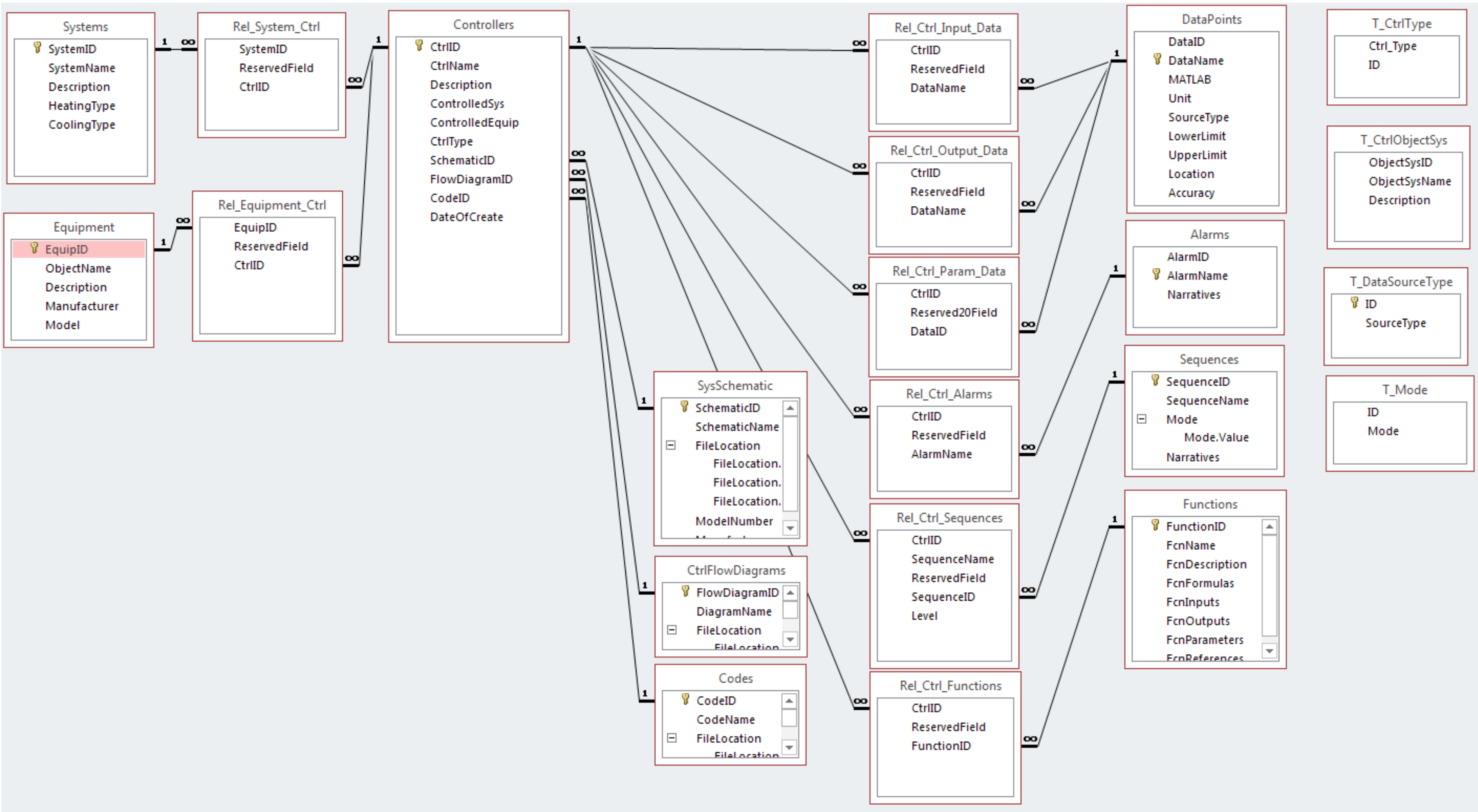


Figure 3-3 Relationships of the classes in BCK database (1-one entity, ∞-many entities, 🔑-primary key).

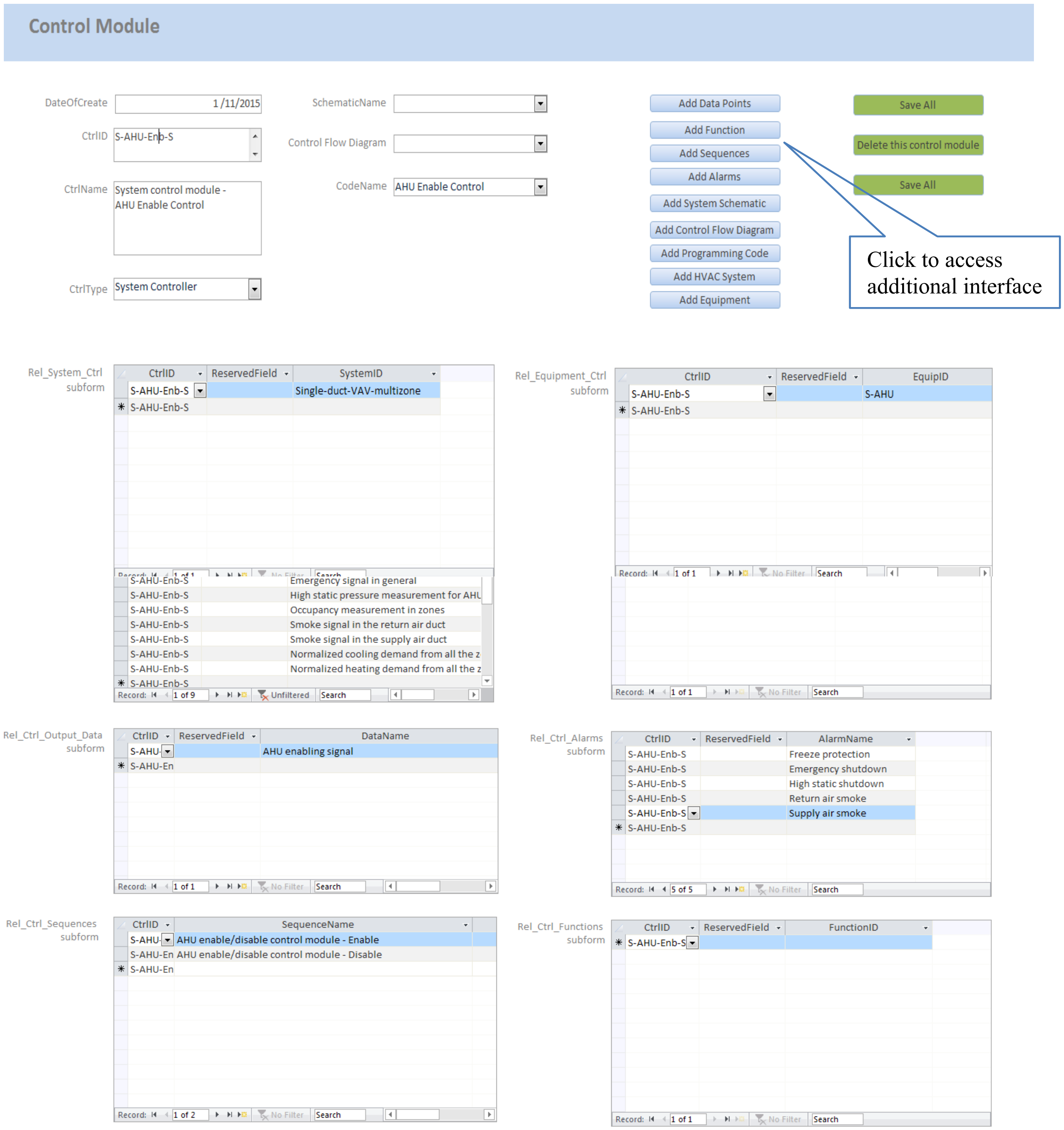


Figure 3-4 Screenshot of the Database Control Module Interface.

As shown in Figure 3-4, this interface incorporated all the attributes of a typical control module in BCK representation. It included eight sub-forms to configure its related systems, equipment, inputs, output, parameters, alarms, functions, and sequences. In each relational table, there is a reserved field for future relationship upgrades.

Meanwhile, this interface provides the capability for users to upgrade data points, functions, alarms, and sequences by accessing additional interface at the main interface. For example, if a data point is not listed in the existing list, the user can enter a new data point. By clicking the button "Add Data Points" in the main interface (shown in Figure 3-4), a new interface will pop-up (shown in Figure 3-5). Once the new data has been created and "Save the Record" button has been clicked, the interface will switch back to the main control module interface. The new data point list will include the newly added data point. Similar interfaces have been created for adding new functions, sequences, alarms, system schematic, control flow diagram, programming code, HVAC systems types, and equipment.

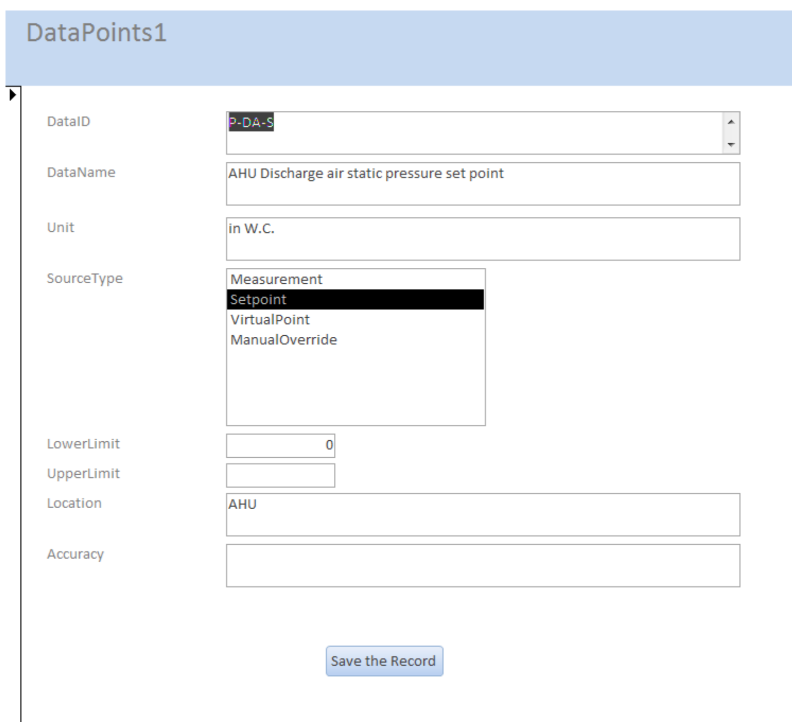


Figure 3-5 Screenshot of the Data Point Interface.

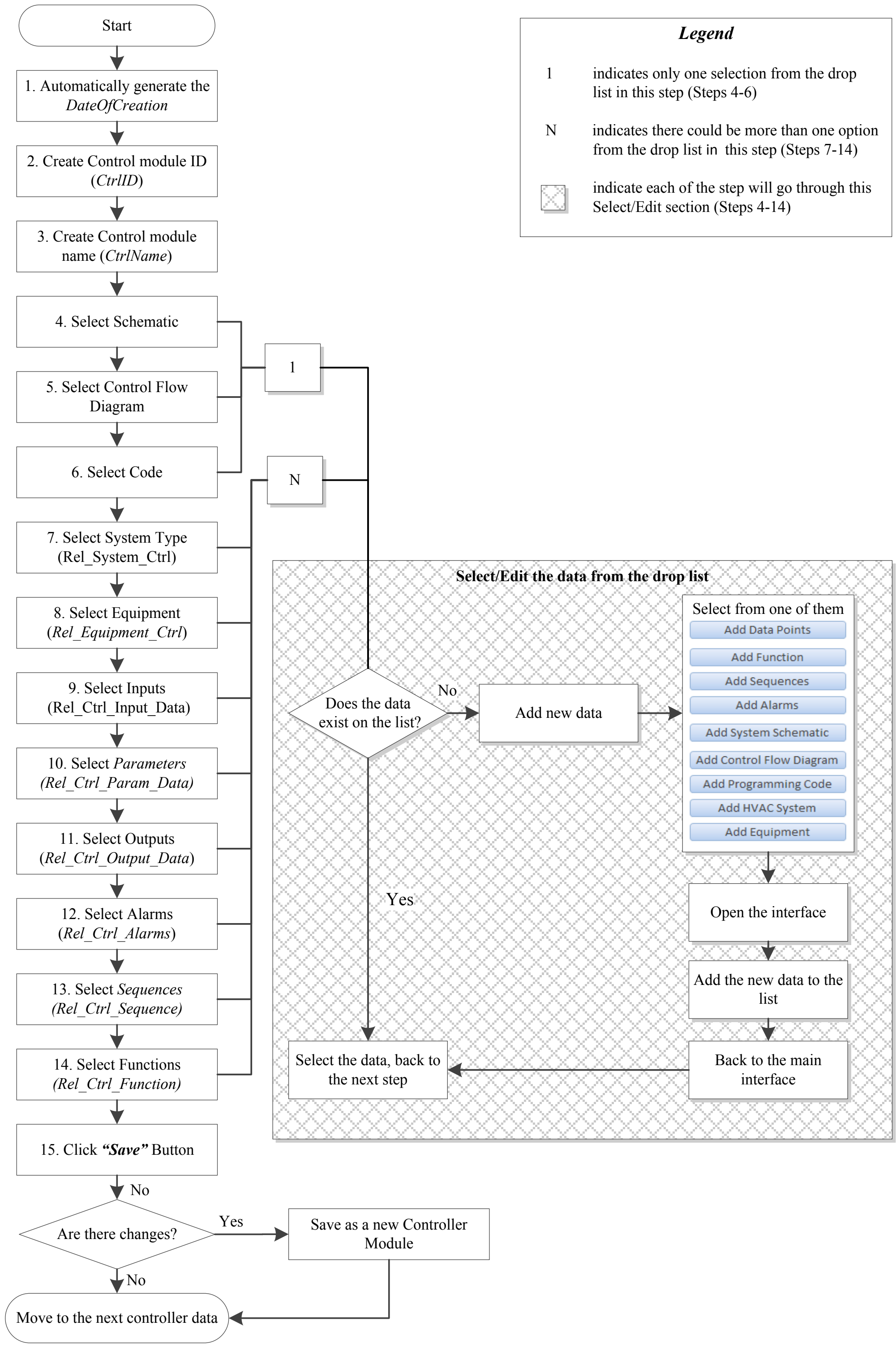


Figure 3-6 Flow diagram of developing a new control module in the database.

To explain the process of adding a new BCK control module, a systematic flow chart was created to illustrate this process. As shown in Figure 3-6, the process of adding new control module incorporates 15 major steps. Among these steps: from steps 4 to 6, command at each step will be executed once; from step 7 to 14, multiple objects might exist for each of these classes, so command at each step might be executed more than once, until the data entering is finished. By going through these steps, the data for a specific control module can be added into the database.

3.4.3 Data Query Interface and Algorithm

To search and find the appropriate control knowledge based on given criteria, an interface for querying control module has been created (in Figure 3-7). A set of queries have been developed in this database. They are used to support basic searching functions, including searching for relevant control knowledge (e.g., code, sequences) for a specific HVAC system types, a HVAC equipment, and searching relevant equipment for a specific data point.

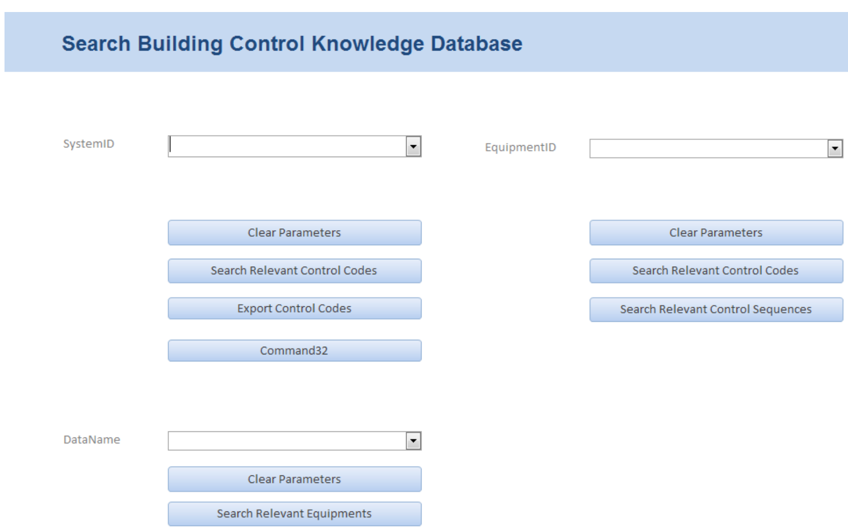


Figure 3-7 Screenshot of the query interface.

The Structured Query Language (SQL) codes were developed using the Query Wizard function in Microsoft Access 2013. A quote of the SQL code for searching the relevant control codes for a specific system type is given in Figure 3-8.

```
SELECT Systems.SystemID, Controllers.CtrlID, Controllers.CtrlName, Codes.FileLocation
FROM Codes INNER JOIN (Controllers INNER JOIN (Systems INNER JOIN Rel_System_Ctrl
ON Systems.[SystemID] = Rel_System_Ctrl.[SystemID]) ON Controllers.[CtrlID] =
Rel_System_Ctrl.[CtrlID]) ON Codes.CodeID = Controllers.CodeID
WHERE (((Systems.SystemID)=[Forms]![Search1]![SystemID]));
```

Figure 3-8 Code for querying the relevant control code based on given SystemID – programmed in SQL.

The code for exporting attachment is given in Figure 3-9. This code was developed in Microsoft Visual Basic for Applications (VBA) based on programming template for the Field2.SaveToFile Method (DAO), provided on discussion forum by Microsoft (2011).

```
Private Sub Command31_Click()
    Dim dbs As DAO.Database
    Dim rst As DAO.Recordset2
    Dim rsA As DAO.Recordset2
    Dim fld As DAO.Field2
    Dim strFullPath As String
    Dim strFullPath2 As String
    Dim Form As Form
    Set Form = Form_Search1
    Dim combo As ComboBox
    Set combo = Form_Search1.SystemID

    Set dbs = CurrentDb
    Set rst = dbs.OpenRecordset("SELECT Codes.FileLocation FROM Codes INNER
JOIN (Controllers INNER JOIN (Systems INNER JOIN Rel_System_Ctrl ON
Systems.[SystemID] = Rel_System_Ctrl.[SystemID]) ON Controllers.[CtrlID] =
Rel_System_Ctrl.[CtrlID]) ON Codes.CodeID = Controllers.CodeID WHERE
(((Systems.SystemID)= '" & [Forms]![Search1]![SystemID] & "'))")
    Set fld = rst("FileLocation")

        combo.SetFocus
        strFullPath2 = "C:\Users\BCKresearcher\Control Information Knowledgebase
Data structure\TestFolder\" & combo.SelText
        MkDir strFullPath2

     Do While Not rst.EOF
        Set rsA = fld.Value

        combo.SetFocus
        strFullPath = "C:\Users\BCKresearcher\Control Information Knowledgebase
Data structure\TestFolder\" & combo.SelText & "\" & rsA("FileName")
        If Dir(strFullPath) = "" Then
        rsA("FileData").SaveToFile strFullPath

        End If
        rsA.Close
        rst.MoveNext
    Loop

    rst.Close
    dbs.Close

    Set fld = Nothing
    Set rsA = Nothing
    Set rst = Nothing
    Set dbs = Nothing
    End Sub
```

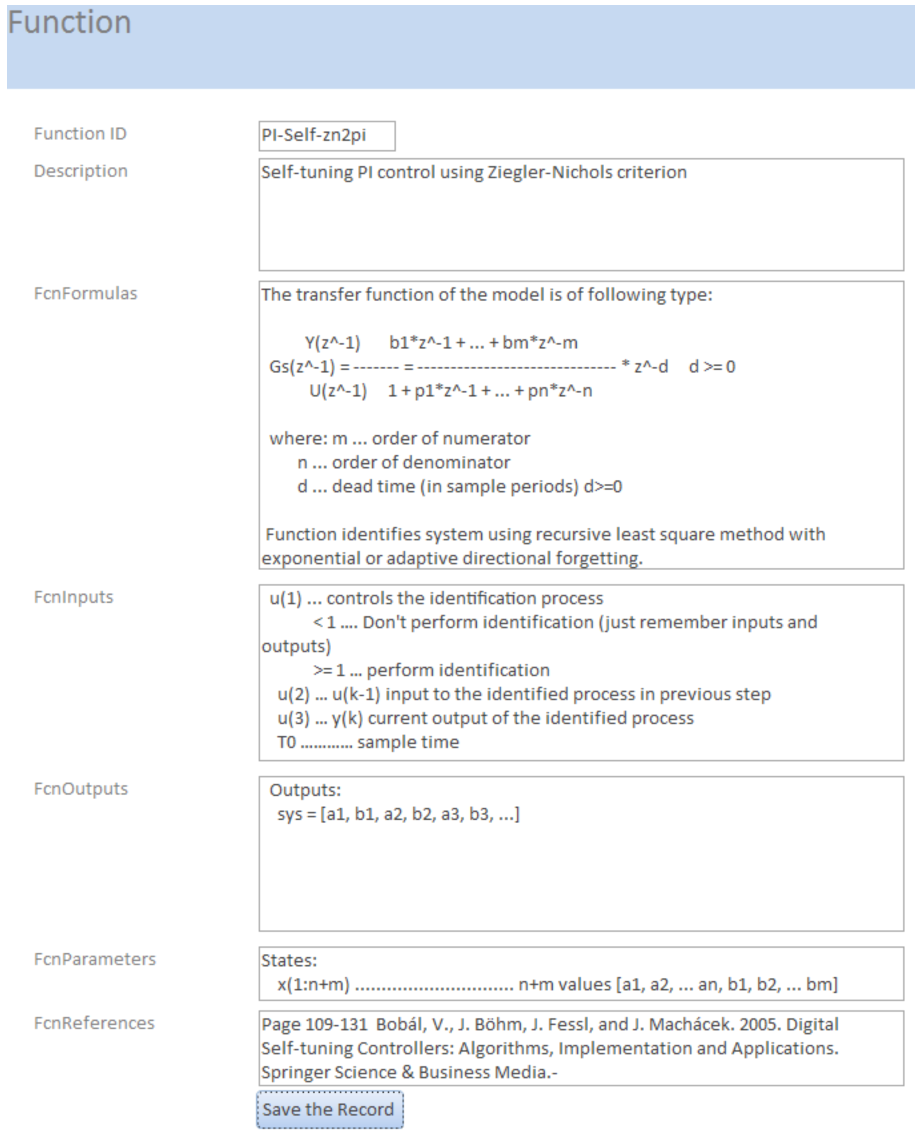
Figure 3-9 Code for exporting selected control codes (attachments) – programmed in Microsoft Visual Basic for Applications.

3.5 Demonstration Cases

This section demonstrates the application of this database. The first part shows the interface of adding a new control function (algorithm) into the database. The second part illustrates the process of finding appropriate control modules for a given HVAC system type.

3.5.1 Add Self-Tuning PI Control As a New Function.

An example is given in Figure 3-10. It shows the details of adding the self-tuning PI control algorithms to the database. The formulas and illustration come from the work by Bobál, Böhm et al. (2005). In this way, it provides a clear explanation of the calculation details of this method.



Function

Function ID PI-Self-zn2pi

Description Self-tuning PI control using Ziegler-Nichols criterion

FcnFormulas The transfer function of the model is of following type:

Y(z^-1) b1*z^-1 + ... + bm*z^-m
Gs(z^-1) = ------- = ----------------------------- * z^-d d >= 0
U(z^-1) 1 + p1*z^-1 + ... + pn*z^-n

where: m ... order of numerator
n ... order of denominator
d ... dead time (in sample periods) d>=0

Function identifies system using recursive least square method with exponential or adaptive directional forgetting.

FcnInputs u(1) ... controls the identification process
< 1 .... Don't perform identification (just remember inputs and outputs)
>= 1 ... perform identification
u(2) ... u(k-1) input to the identified process in previous step
u(3) ... y(k) current output of the identified process
T0 .... sample time

FcnOutputs Outputs:
sys = [a1, b1, a2, b2, a3, b3, ...]

FcnParameters States:
x(1:n+m) .... n+m values [a1, a2, ... an, b1, b2, ... bm]

FcnReferences Page 109-131 Bobál, V., J. Böhm, J. Fessl, and J. Machácek. 2005. Digital Self-tuning Controllers: Algorithms, Implementation and Applications. Springer Science & Business Media.-

Save the Record

Figure 3-10 Screenshot of the newly added self-tuning PID control function.

3.5.2 Finding the List of BCK Control Modules

This section demonstrates the process of finding the appropriate control modules (shown in Figure 3-11). This demonstration uses "Single-duct-VAV-multi-zone" system type as an example. In order to find the relevant BCK control modules for this system type, the specified

HVAC system type is firstly selected at ①. Then execute the query by click the "Search Relevant Control Codes" at ②. After successfully execution of the query, a result window would pop up and display the list of relevant control module. If this result needs to be exported and saved as a specific folder, then click the "Export Control Codes" at ③. In this way, the users can find the relevant control modules for the specific system types and equipment. The related building control knowledge can be packaged for individual modules.

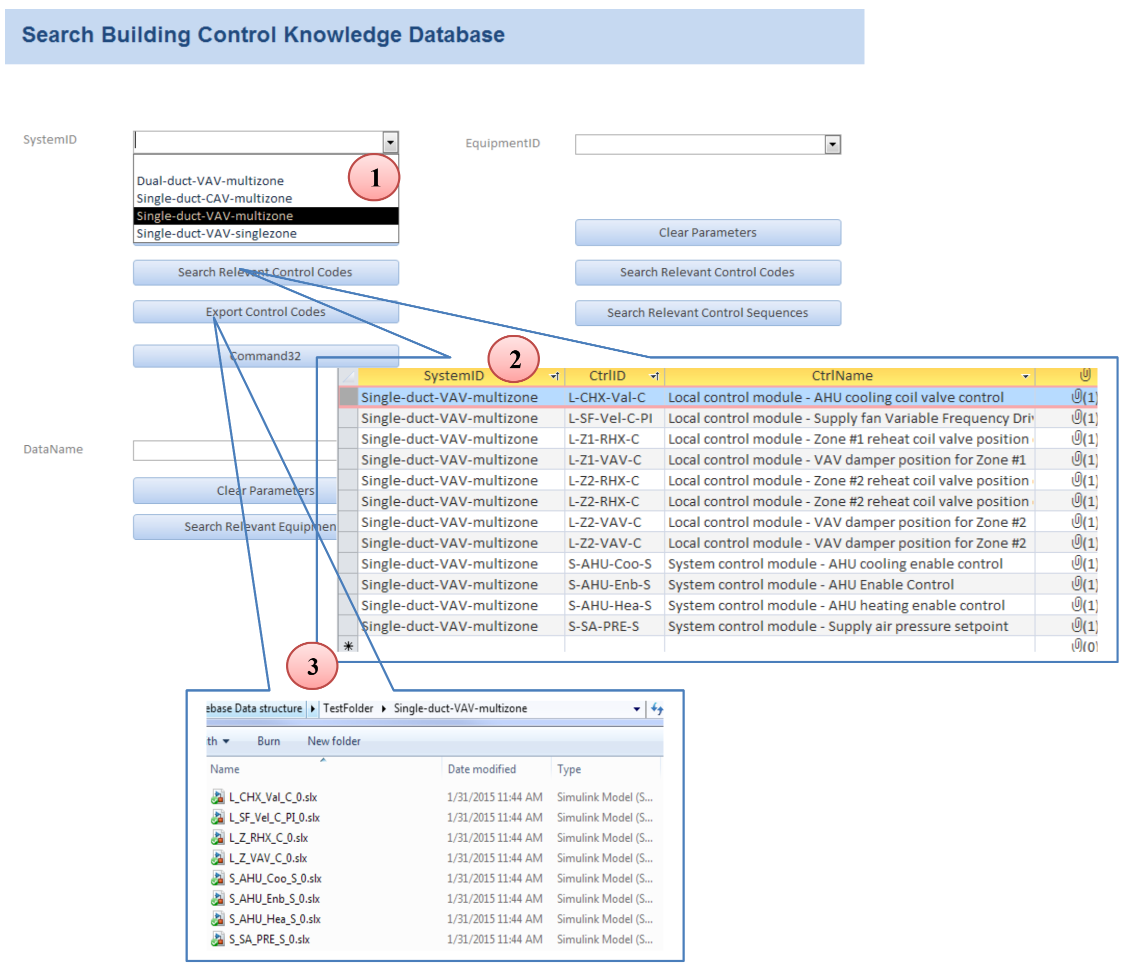


Figure 3-11 Demonstration of searching relevant building control knowledge for "Single-duct-VAV-multi-zone".

3.6 Discussion

In summary, prior work has documented the building control knowledge employed in the existing building control system. Examples of these collections include CtrlSpecBuilder (AutomatedLogic 2012), *Library of system control strategies* (Martin and Banyard 1998), and *The sequence of operation* CD by ASHRAE (2005). However, these documentation and collections have neither a shared standard format of representing the control knowledge, nor an interactive platform to support adding new data. Lacking of these two features in the current BCK resources may have obstructed the deployment of high performance control algorithms as well as the explicit interpretation of building control knowledge.

To address these issues, this chapter presented a data model for representing building control knowledge with well-defined format. It implemented this data model into an interactive database in MS Access, with the demonstrations of querying relevant control knowledge and adding advanced control algorithms. In addition, function (algorithm) has been added in the database as a dedicated class to provide explicit description of the calculation details. It enables the developers to provide clear description of control algorithms in the database so that the users can understand and interpret these algorithms, and maybe use it for fault detection and diagnosis. This chapter has made an effort to bridge the gap that existed between advanced control algorithms development in simulation and their deployment. It further contributes to the infrastructure development that are currently lacking in building control knowledge storage and management.

Most notably, this is the first study to the author's best knowledge that has investigated the method for storing and managing building control knowledge via a generic standard format and structure. The demonstration provides compelling evidence for this database to be able to be adopted by the building control industry to standardize BCK representation. This database also serves as a foundation for the self-configuration of building control described in Chapter 4.

However, limitations are also worth noting. First, this database is in its first version of design. It was only tested via the MS Access software and the control programming codes were developed using the MATLAB based Simulink/StateFlow. Future work may consider including surveys from the building control industry to improve the data model and format design. Second, only a few numbers of control modules have been developed and stored in this version of BCK database. In the future, result from the on-going researches about building control knowledge (such as *RP 1455: Best of Class Control Sequences for Air Systems* (Hydeman and Eubanks 2015)) should be continuously incorporated into the database content. Thus, in the author's vision, the BCK database shall be a living and growing warehouse for building control knowledge accumulation and be continuously upgraded, but not a one that "once and for all".

CHAPTER 4

SELF-CONFIGURATION OF BUILDING CONTROL KNOWLEDGE

Man takes his law from the Earth; the Earth takes its law from Heaven; Heaven takes its law from the Tao. The law of the Tao is its being what it is.

The Tao Te Ching (Section: 25)

by Lao Tzu

This chapter focuses on the development of the self-configuration framework based on interacting with the BCK database. Four self-configuration scenarios are created based on the typical application scenarios in the field, including new system configuration, retrofit reconfiguration, BCK database upgrade reconfiguration, and sensor & equipment failure reconfiguration. Detailed processes (flow charts) for each scenario have been developed.

4.1 Introduction

The configuration of a building control system involves the stages of building design, implementation and operation. It is related to HVAC system designer, control system designer, control system programmer and commissioner, and facility (HVAC system) operator and maintainer (Montgomery and McDowall 2008). As Baldwin, Bishop et al. (1994) pointed out, "Many problems which are found in prior systems result from human error in programming or identifying the system configuration to the controller. Other problems occur because the system configuration changes due to component failure, recovery or to modification of the system by the owner". To address these issues, researchers have been working on developing self-configurable control systems.

First of all, the concept of self-configuration was primarily introduced to the field of computer and memory system in the late 1970s (Chesley 1977; Chesley 1980). It was used to describe a method that can identify the good CPU, ROM, and RAM to form a computer and memory system, by sequentially testing of a plurality of CPU, ROM and RAM circuits on a semiconductor wafer. Now this concept is used a lot in the automatic computing field and described as "when a new component introduced into an autonomic accounting system, it will automatically learn about and take into account the composition and configuration of the system. It will register itself and its capabilities so that other components can either use it or modify their own behavior appropriately" (Kephart and Chess 2003).

This terminology was brought to the building industry in the 1990s, symbolized by the invention from Baldwin, Bishop et al. (1994). Since then, a number of patents and studies have been developed to realize the HVAC control self-configuration. These works may be divided into two categories, self-configuration of whole building control system, and self-configuration (or sometimes also called self-tuning, self-organizing) of one specific control loop.

Within the first category, Baldwin, Bishop et al. (1994) describes the process of how to configure or reconfigure the operation mode based on the validity check result of the input devices. This self-configuration process is designed for compressor and economizer. Ryan and Shah (2007) described a self-configuration process for a centralized controller unit. This controller unit utilizes a microprocessor to process characteristics profiles, provided by individual units and, then self-configure its control algorithm based on the profiles. Wruck (2008) developed algorithms (in the format of narratives and diagrams) to automatically select the appropriate control logic based on the availability of inputs. Those algorithms were designed for economizer control. Harrod, Rigg et al. (2011) developed a method that may configure and reconfigure the setting of building control circuits based on the available HVAC system characteristics. Certain options of control algorithms configurations were given in the control devices.

Within the second category, Xu, Wang et al. (2004) utilized a split-range PI gains scheduling strategy to modify the PI gains (parameters) for air handling unit control. Nesler (1986) described the automatic tuning adaptive controller using the recursive least squares (RLS) algorithm. Seem (2013) introduced a method using control strategies switching module to select appropriate control module (either model-based controller, or self-optimizing controller). Bai, Wang et al. (2008) developed an adaptive Smith predictor-based self-tuning PI controller for an HVAC system. A number of studies used fuzzy logic for self-configuration of the control parameters (Procyk and Mamdani 1979; Shao 1988; Maeda and Murakami 1992; Dunlop, Burnham et al. 1995).

Among these work, the self-configuration of the control systems (first category) are expected to facilitate the configuration speed (automatically create the control code, documents, etc.) and ideally have a wide applicable range; and the self-tuning controllers (second category) are expected to eliminate the requirement for control parameters tuning, and reduce the system instability issues. However, the existing methods and products for the first category have some limitations: first, these methods are only applicable to specific building HVAC types or one specific HVAC system component. There is a lack of a general self-configuration approach that can be applied to a broader range of HVAC system. Second, among these methods, there is a lack of consideration for future upgrade with advanced control algorithms.

These limitations may restrict the application of these self-configuration methods and products. Today, the industry is still largely depended on manual interpretation of the control system design and programing of the control code based on the control sequences. It would be beneficial to have a generic self-configuration framework that covers typical HVAC system types as well as supports upgrades.

Therefore, to advance the domain of building control self-configuration (for the first category), this chapter focuses on the development of a generic self-configuration framework. This framework interacts with the BCK database to configure the control programming code and documentations for specific HVAC system. Specifically, Section 4.2 presents the self-configuration framework based on the discussion of functional requirement, different self-configuration scenarios, and the prerequisite. The processes (flow charts) are also developed. Section 4.3 discusses the implementation status and issues.

4.2 The Self-configuration Framework

This section further illustrates the new approach to self-configure /automatically select the BCK modules. The discussion includes an overview, functional requirement, scenarios design, prerequisite and processes.

4.2.1 Overview

Contrasting with the existing self-configuration methods, the concept for this new approach stems from the how the knowledge has been generated, and documented in a book/documentation, and then stored in a library and then read by other people. As shown in Figure 4-1, this BCK self-configuration approach mimics a library, and a process that configures the BCK modules based on an existing BCK database.

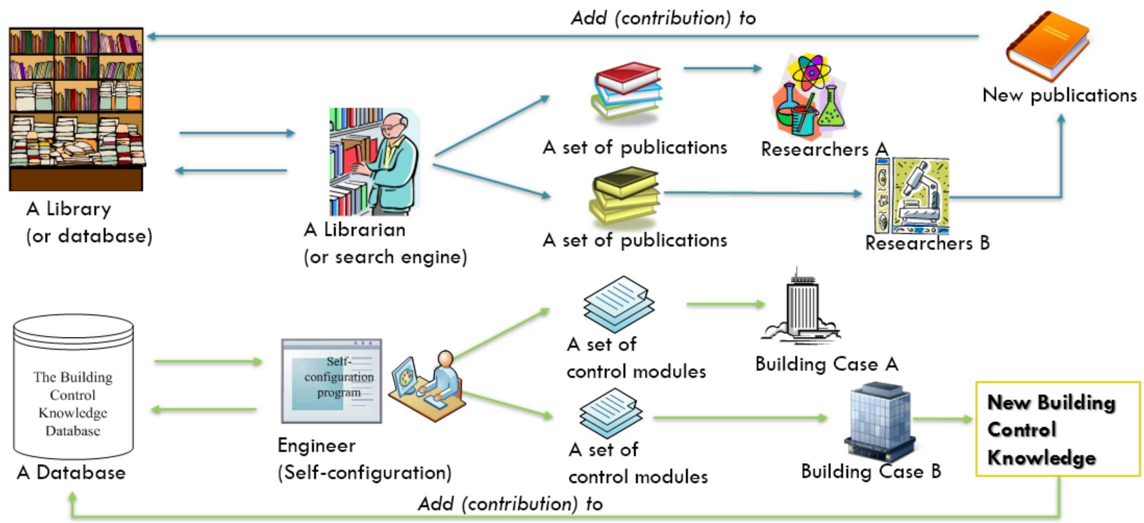


Figure 4-1 Inspiration of the method.

Typical BCK have been modularized into individual control modules based on their control variable (in Chapter 2). These modules are organized based on their associated HVAC system types and equipment and stored in a BCK database (in Chapter 3). Therefore, this part of work is intended to create a framework to interact with the BCK database based on specific HVAC system profile, and automatically configure the appropriate BCK modules (see Figure 3-1).

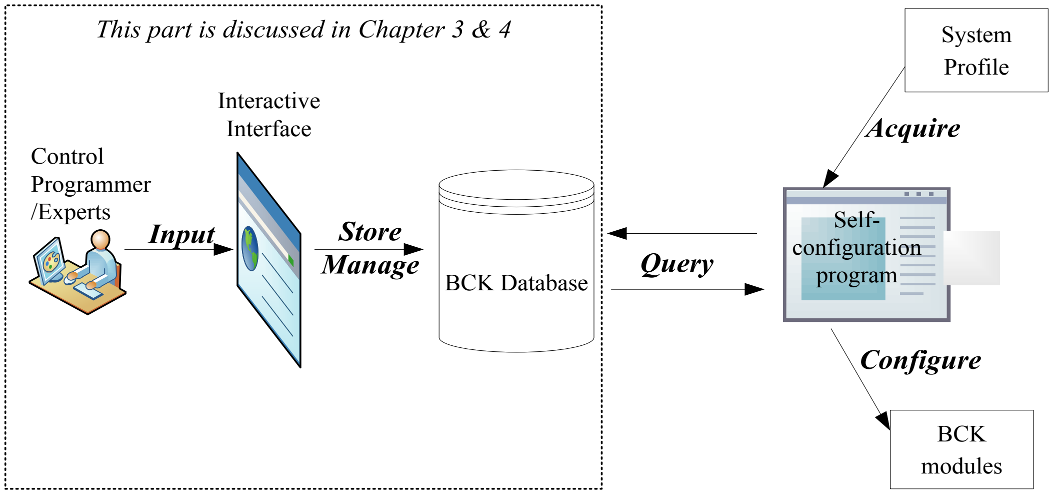


Figure 4-2 Overview of the self-configuration process interacting with the BCK database.

The concept of self-configuration is built upon the ontology of data points using the list of local level control variables to first identify the local level control modules. Then, further identifying the required upper level control modules based on their control inputs. This approach automatically filters the unrelated control modules from the top down, and then builds the control modules from the bottom up, as shown in Figure 4-3.

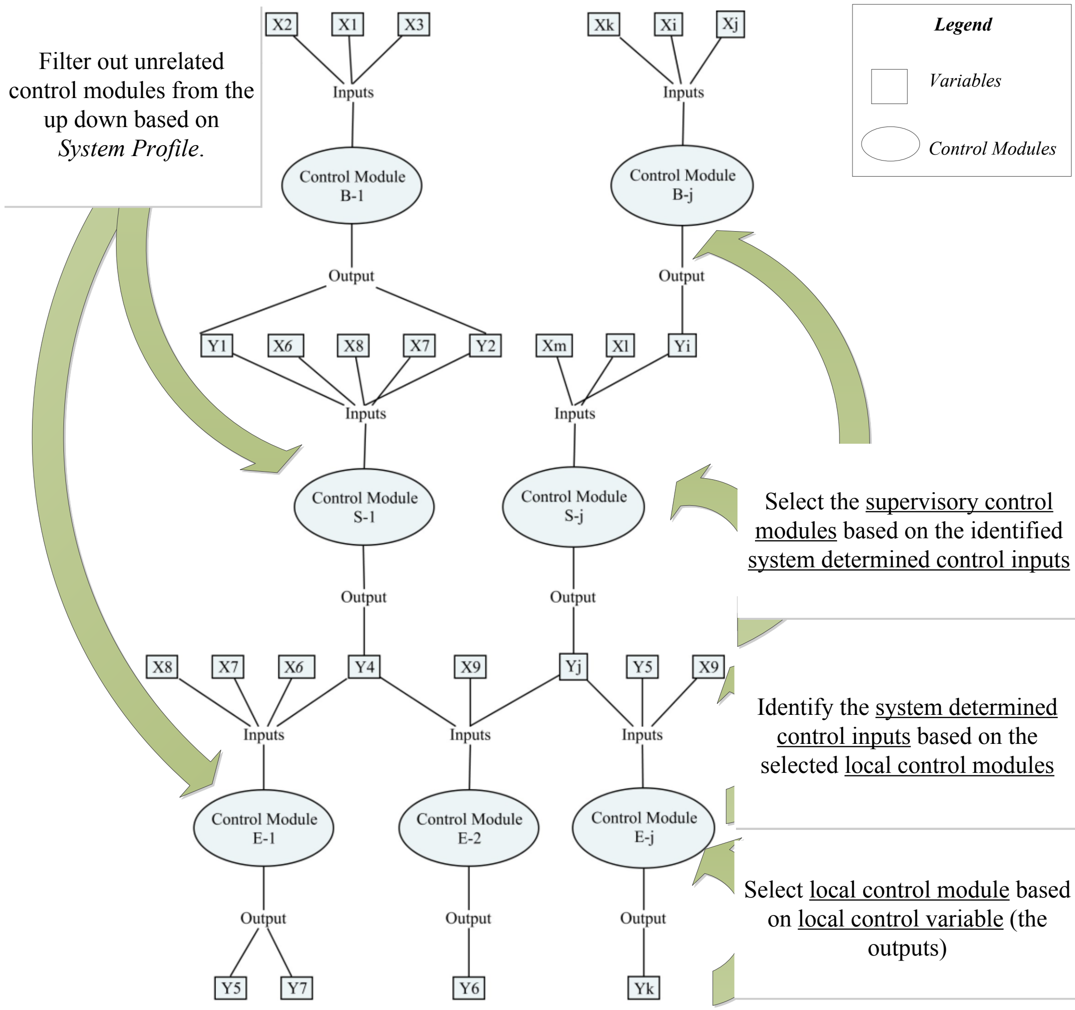


Figure 4-3 A bottom-up approach to self-configure control modules based on variables' ontology.

4.2.2 Functional Requirement

It is important to discuss the functional requirement for the control self-configuration. Akinci, Garrett et al. (2011) investigated the functional requirement for building self-configuration through a workshop survey analysis. Their work identified the functional requirements for self-configuring HVAC system, including:

1. Be able to talk *with* neighboring systems.
2. Be able to generate and evaluate alternative configurations.
3. Pass the equivalent of the tuning test.
4. Use less resource.
5. Recover from any failure.
6. Be able to learn and unlearn.
7. Change its own state in relation to external influences.
8. Be modular (and have parts).
9. Be able to explain why it made the change.
10. Permit external manipulation.

This study focuses on the self-configuration of building control, instead of the self-configuration of HVAC system. Meanwhile, in the author's opinion, not all the functional requirements listed above are achievable within the timeframe of a doctoral research. Some of them are out of the scope of this work, or within the scope of other studies: Specifically, item #1 addresses the inter-compatibility of different devices; this is within the research scope of device communication network protocol such as *BACnet - A Data Communication Protocol for Building Automation and Control Networks* by (ASHRAE 2010). For #2, it partially includes the evaluation of control algorithms, and the evaluation of building control system itself is another new topic, but generating alternative configuration should be addressed in this work. For #5, the description itself is problematic. In practice, it is impossible for any device to be able to recover from all failures. However, the feature of recovering from typical failures should be considered within #2 as generating an alternative configuration. For #6, 7, they are more related to adaptive control algorithms for specific control loops, which can be found in the advanced control algorithms research. The adaptive control algorithms can be applied to specific control modular(s) as needed. For #8, it has been addressed in the previous work (Chapter 2 and Chapter

3), which serves as a foundation of this chapter as well. Thus, this work will try to achieve the functional requirements listed in #2, 3, 4, 8, 9, and 10.

4.2.3 Self-configuration Scenarios

Four preliminary self-configuration scenarios are envisioned by the author, including new construction, retrofit system, BCK database update, and sensors & equipment failure/change reconfiguration. As shown in Table 4-1 , each scenario covers a range of applications. This work focuses on the procedure development and testing for the first scenario - New System Configuration, because this is the foundation of other three scenarios. Then, it briefly discusses the framework of other three scenarios.

Table 4-1 Four configuration scenarios.

Name	Scope
New System Configuration	The new configuration is to configure the control modules based on system profile, which has not been configured before.
Retrofit Reconfiguration	This scenario is to reconfigure the control modules based on the retrofit of HVAC systems and equipment. It utilizes the new system profile to reconfigure the control modules.
BCK database Update	This scenario enables the control modules upgraded based along with newly developed control algorithms to achieve higher performance. This scenario should be performed periodically through a cloud-based BCK database.
Sensors Failure Reconfiguration	This scenario covers the control reconfiguration due to the failures of sensors and equipment. It requires the status of failed sensor(s) be updated with the System Profile, and then SCP enables the reconfiguration based on new System Profile.

4.2.4 Prerequisites

In order to initiate the self-configuration process, basic information must be provided by the self-configuration process targeted system. That basic information should be formatted in a standard way and named System Profile. As shown in Table 4-2, this profile is composed of the four types of data, including HVAC system types, sensors (data points) list, local control variables list, and specific parameters list and their values.

Table 4-2 A System Profile template.

Items	Details
1. HVAC system type	A system type represents a category of HVAC system. Currently, there is no standard classification. The topic of HVAC system classification itself shall be investigated separately.
2. Sensors (data points) list	A list includes all the name of the hardware measuring data points for this system
3. Local control variables list	A list represents all the control variables from the systems and equipment. For example, Supply fan speed control is a local control variable, while supply air pressure setpoint is not a local control variable.
4. Specific parameters list and their values	A list includes all the parameters specified by system designer for this specific case. The values of these specified parameters will be passed on to the relevant control modules, while non-specified parameters will use their default values from the BCK database.

Meanwhile, the BCK database should have the typical BCK modules for the type of HVAC system that will be self-configured. Within the BCK database, it may happen that there is more than one BCK module capable to realize the same control function (control the same variable). This indicates there is more than one applicable BCK module for this control purpose. Thus, the self-configuration process must be able to select the best performance BCK module among them.

4.2.5 Processes

The overall process includes the determination of the type of self-configuration scenarios, the new system configuration, retrofit configuration, BCK database update reconfiguration, and the sensors & equipment failure reconfiguration. Flow charts are provided to illustrate the steps of each scenario.

4.2.5.1 Determine the scenario

The self-configuration process starts with the determination of the self-configuration scenarios. Figure 4-4 shows the steps of determining the appropriate scenarios. After power up, the self-configure program (SCP) imports the System Profile. Then, SCP determines if this is a new configuration scenario based on the availability of the Configuration Profile file from the system. If there is a Configuration Profile file, SCP compares the similarity between the old system configuration and the new system profile. If these two profiles (sensors list, HVAC system types, local control variables, and parameters) do not match with each other, SCP assigns this configuration to the retrofit reconfiguration scenario. Otherwise, there is no hardware change in the system configuration. Thus, SCP checks to see if there are any updates in the BCK database to upgrade the control algorithms. After that, SCP checks the failure cases of hardware and reconfigures the alternative BCK modules to eliminate the effect of failure cases. If none of the configuration is performed, SCP updates the date on the *configuration profile* to the current date at the *checked on YYYY-MM-DD, and* exits the self-configuration process.

(Note 1: each time, when a system is successfully configured, SCP shall generate a C*onfiguration Profile*, including the hierarchical layout, (sensors list, HVAC system types, local control variables, and parameters.

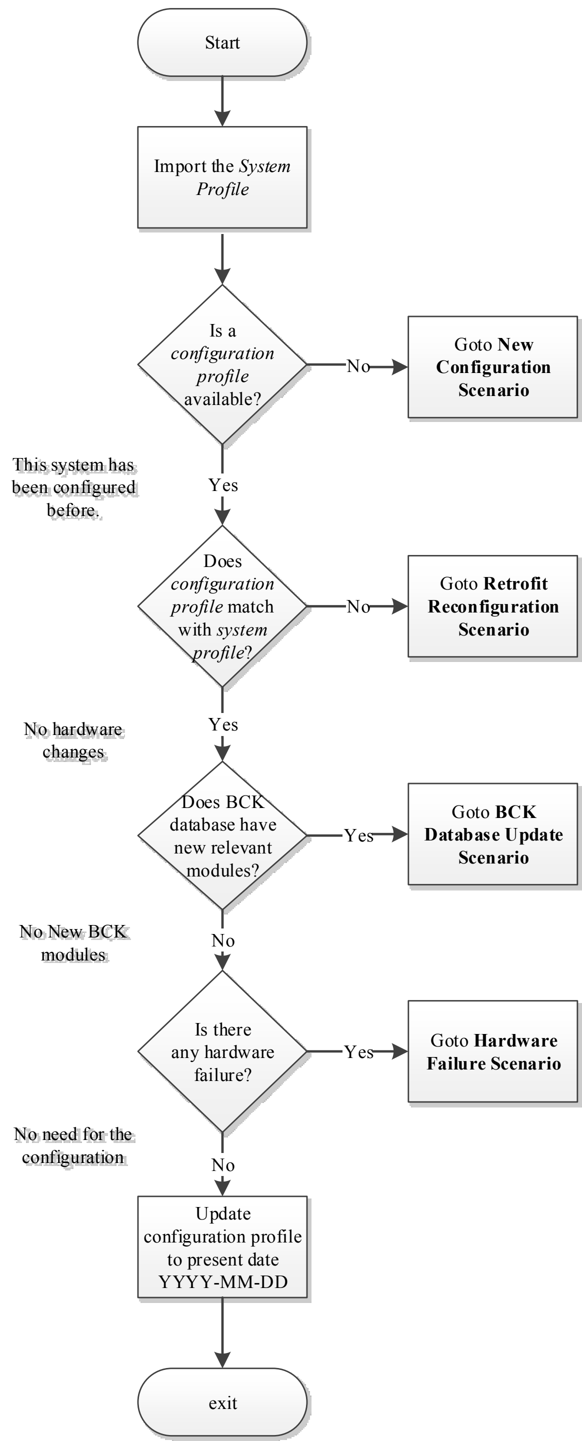


Figure 4-4 Process of determining the configuration scenarios.

4.2.5.2 Scenario 1: New system configuration

Part A: filtering BCK database and configuring the local level control modules:

As shown in Figure 4-5, the new system configuration scenario starts with the preliminary screening of the BCK database. It firstly utilizes the HVAC system type information from the *System Profile* to select the relevant BCK data.

The next step is to select the local control modules (shown in Figure 4-6). The local control modules were selected based on the local control variable from the *System Profile*. In this way, a list of local control module (called "local" because it determines the value of local control variable) can be selected.

Among those control modules, it is possible that one control variable will have multiple options of control modules that each have different control algorithms (e.g., PI, Self-tuning PI, P, etc.), but achieve the same control function. Particularly, because new control algorithms are being developed in research studies to optimize the control performance, this situation would likely happen at the supervisory levels of control variables. Thus, the BCK database design (in Chapter 4) has considered this situation and assigned an attribute to represent the version of control algorithms based on their control module ID, i.e., the last digit of the ID represents the version of its control algorithm, the latest version has a highest number. Therefore, the ID criterion is added to the selection process. In this way, the local control modules set can be assembled by the selected local control module, which satisfies the requirements.

Part B: configure the system level control modules

With the identified local control module set, the next step is to identify the system level control module set. In this part (shown in Figure 4-7), SCP firstly utilizes the input variable information from each local control module, selects the input variables that are "setpoint", and excludes the "sensor" ones. These "setpoint" variables are identified as belonging to the system control variables. Additional system control variables are identified in the process of configuring system control modules. Because input variables for system control modules could be "setpoint". In this way, the "setpoint" variables are looped identified during the selection process for system module until the last one. A list of system control module (called "system" because it determines the value of setpoint variable at the supervisory level) can be selected from the data set. The system control modules set can be assembled by the selected system control module that satisfies the requirements above.

Part C: Quality check and validation

This part is to check the configuration validity. Quality check and validation process shall be executed to sort through the complete set of selected modules. There are two rules that the configuration result shall follow: 1) each control output is solely controlled by one control module; 2) each input is provided either by the sensor reading from the System Profile, or by the control output from system/local control module. More rules can be added in a later version based on development requirement. Thus, to make sure that each control module controls a unique control variable and every input data is provided, this part is added in the process map (Figure 4-8) to check the redundancy and completeness of control configuration.

Part D: Export and configure each control module

This is the last part of the self-configuration process for the new system configuration scenario. As shown in Figure 4-9, SCP sorts through the database and uses the control module ID lists from "NewConfig_local" and "NewConfig_sys" to export each BCK control module and its associated data into a specified folder location. During this process, based on the specified parameters from the System Profile, parameters for individual control modules are configured with the specific value or the default value. Each module's associated data include the inputs, outputs, parameters, control flow diagrams, control sequence, control programming code, and alarms. In general, all the exported data are organized based on the control modules, which are based on the control variables' ontology analysis. For example, control sequences are generated automatically, and organized based on the operation mode at the lower level, and then organized based on control modules at the higher level. Thus, it can be easily sorted into the operation mode based version or control variable based version.

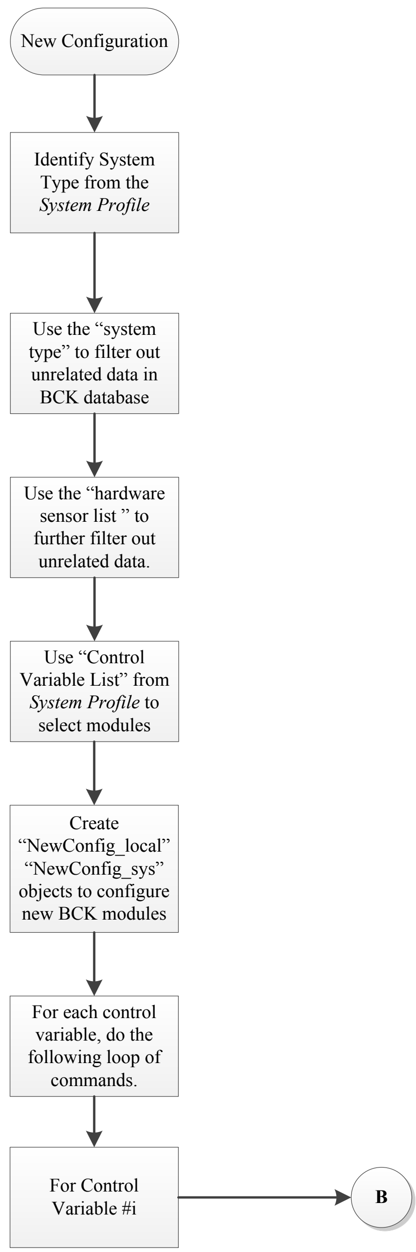


Figure 4-5 Process of new system configuration (Part A-1).

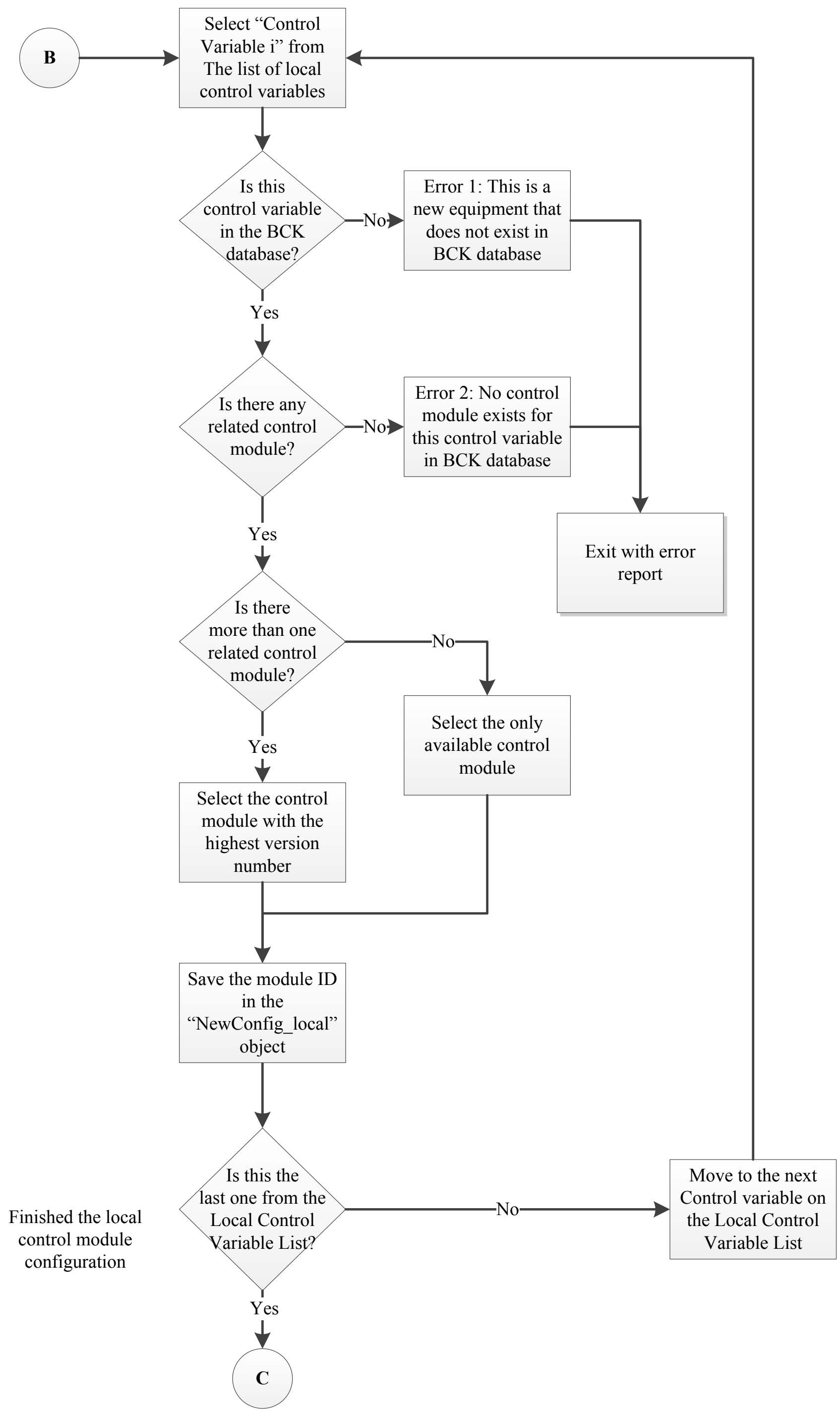


Figure 4-6 Process of new system configuration (Part A-2).

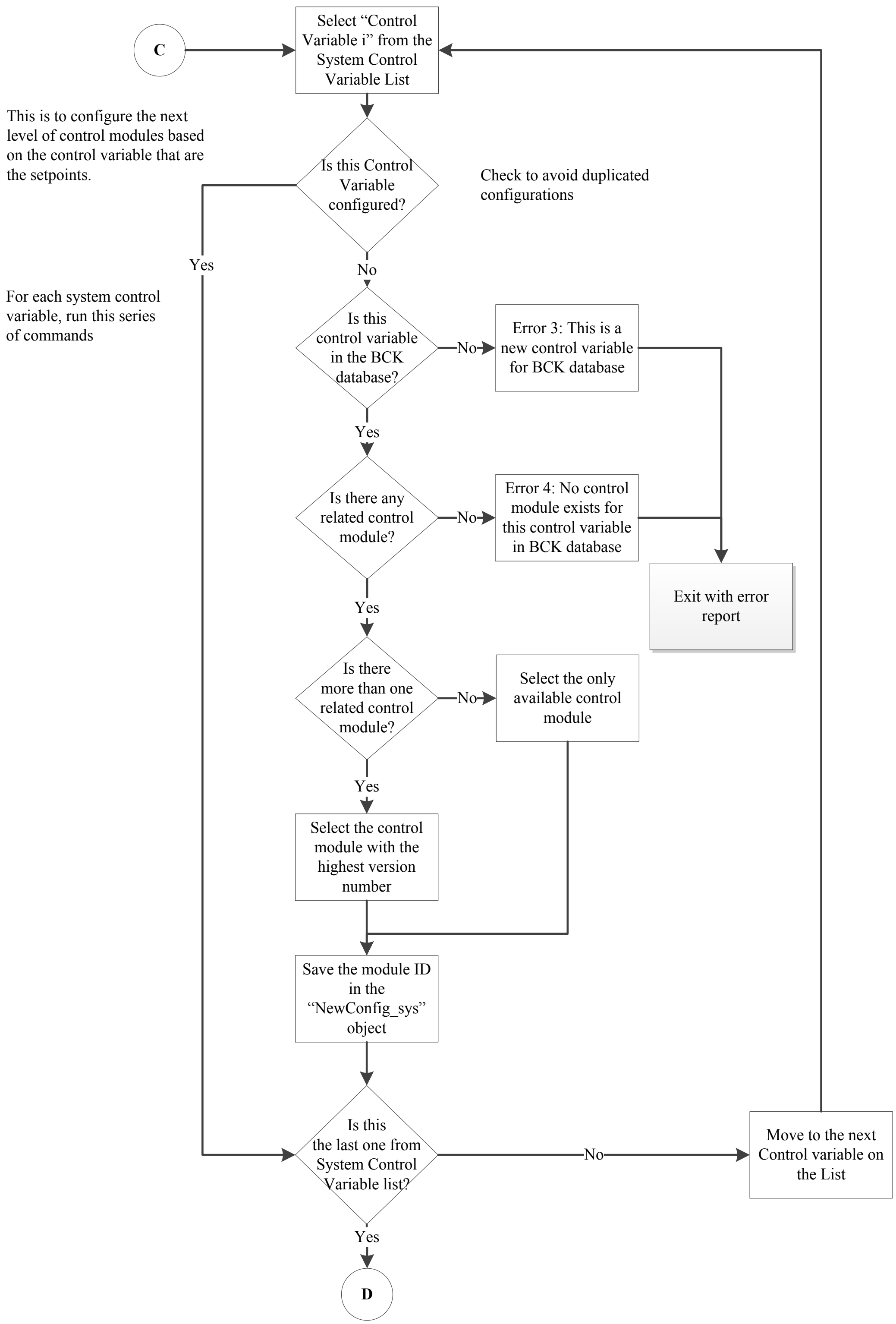


Figure 4-7 Process of new system configuration (Part B).

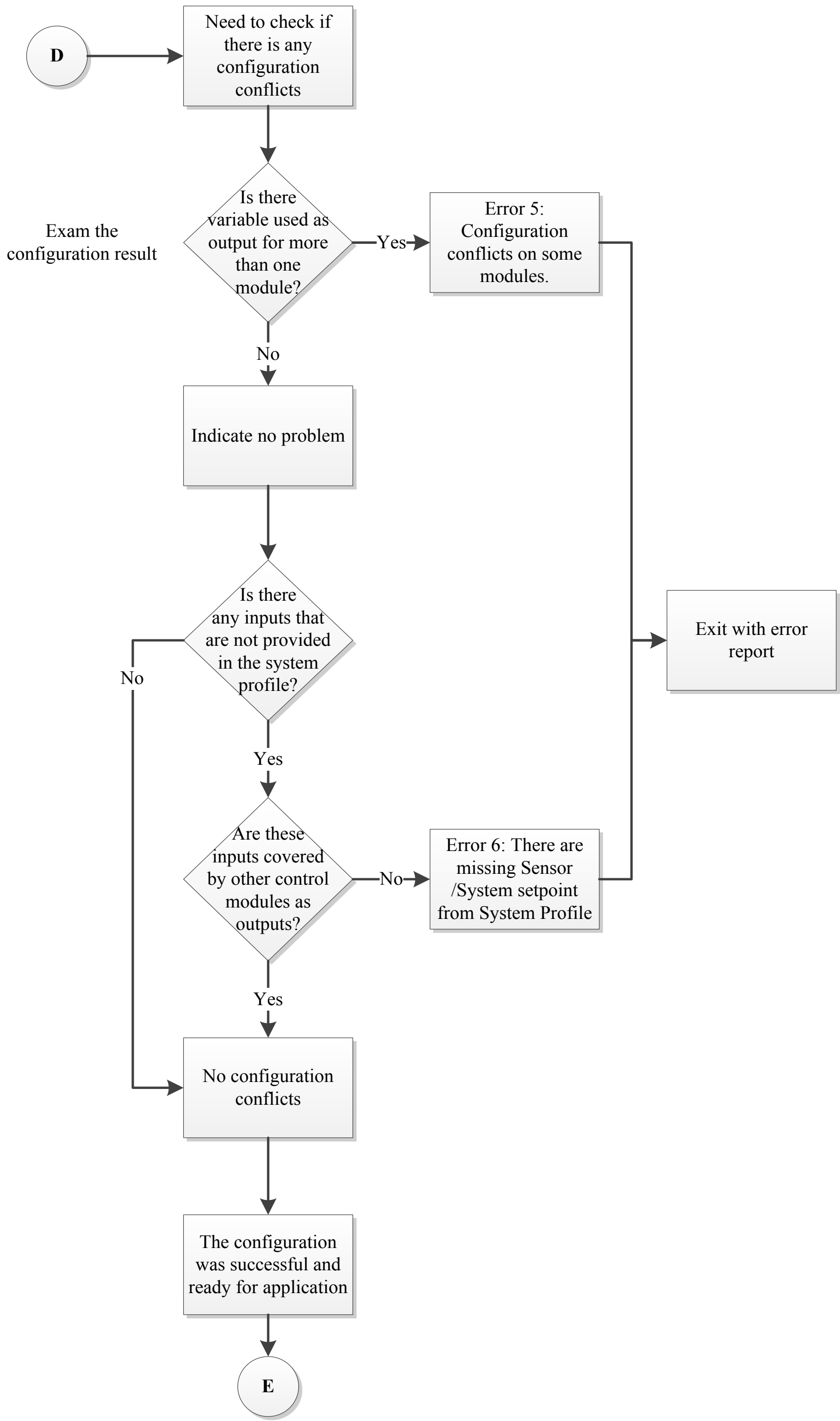


Figure 4-8 Process of new system configuration (Part C).

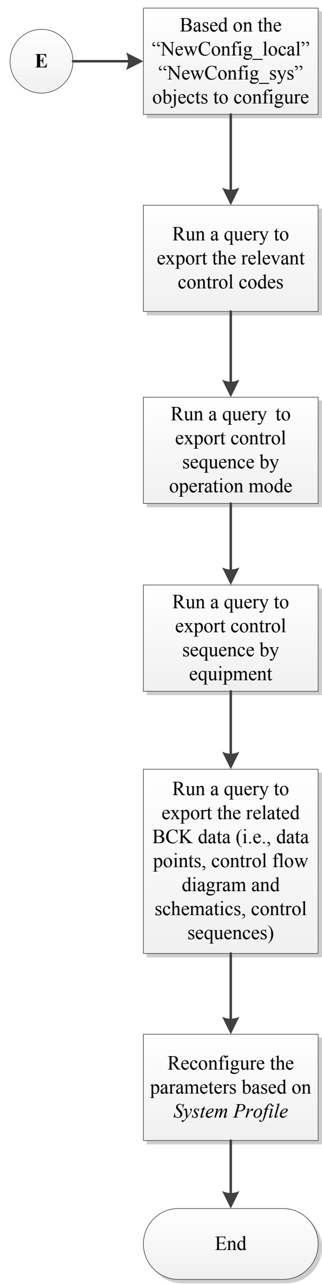


Figure 4-9 Process of new system configuration (Part D).

4.2.5.3 Scenario 2: Retrofit reconfiguration

As shown in Figure 4-10, the retrofit configuration scenario starts with the comparison between old System Profile with new System Profile. This helps to identify the changes that happened to HVAC system hardware configuration (i.e., HVAC system types, sensor list, and equipment list). Then, it repeats the new system configuration process to reconfigure control modules.

4.2.5.4 Scenario 3: BCK database upgrade reconfiguration

As shown in Figure 4-11, this scenario starts with the identification of the applicable control modules from the BCK database, based on HVAC system types and the list of control variables (of the System Profile). This step is to check if any upgrade can be applied to the existing configuration. Then, if there are applicable control modules, SCP will check the compatibility and requirement of hardware and parameters. If the check result indicates that the configuration can be upgraded, then SCP will upgrade that control modules. During this process, it might affect other control modules due to the requirements of inputs. Thus, reconfiguration of the system-level control modules has to be conducted if there is upgrade among the local control modules. Therefore, the next step follows the process described in the Part B, C, and D of the New Configuration scenario.

4.2.5.5 Scenario 4: Sensors & equipment failure reconfiguration

As shown in Figure 4-12, the failure reconfiguration scenario starts with the identification of failed devices (sensors or equipment), and then updates the System Profile and repeats the new system configuration process to reconfigure the control modules.

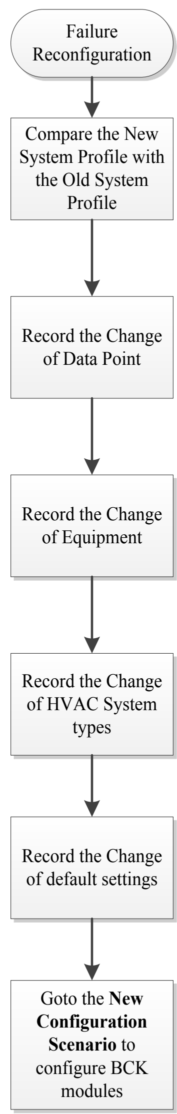


Figure 4-10 Process of retrofit reconfiguration.

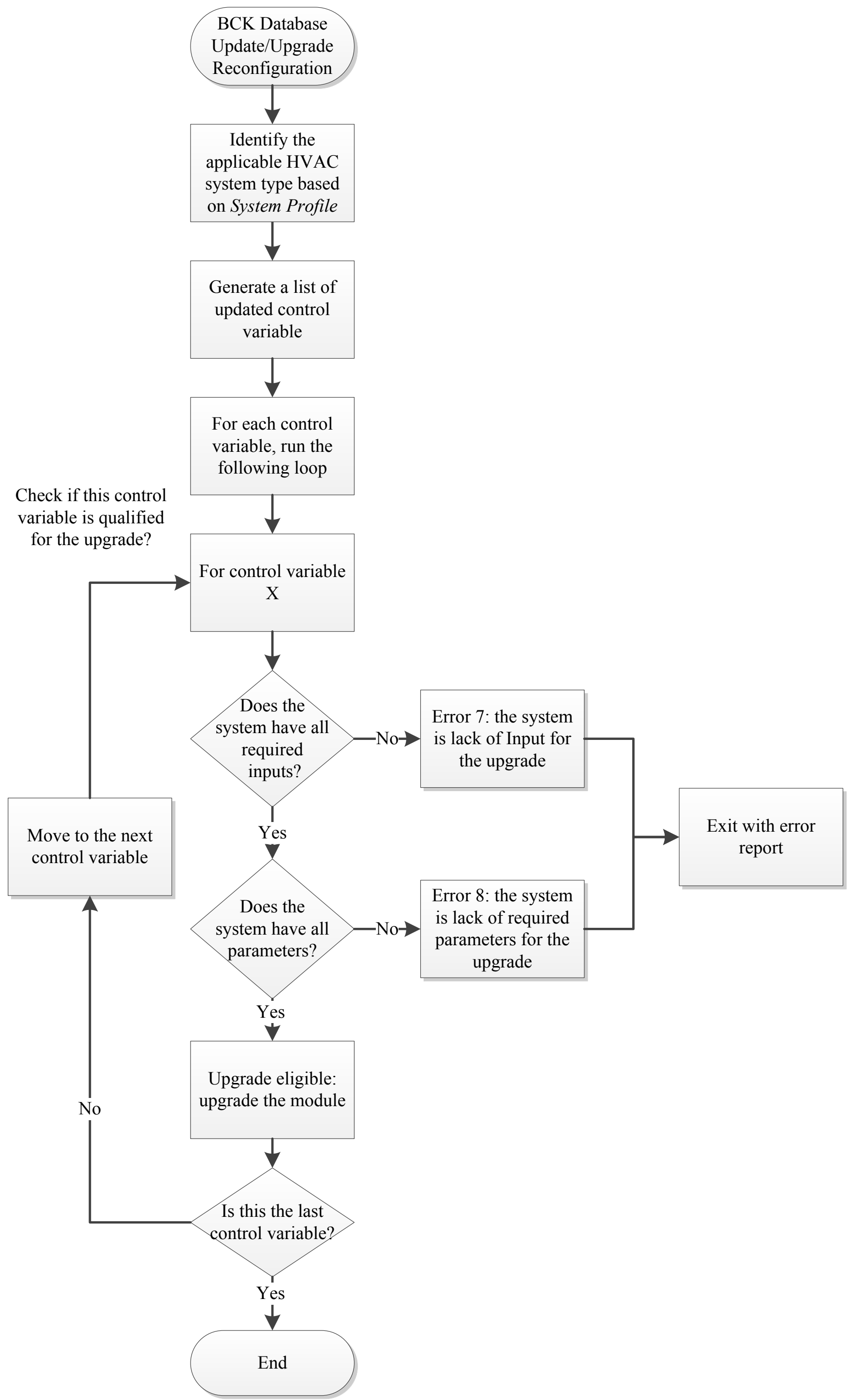


Figure 4-11 Process of BCK database upgrade reconfiguration.

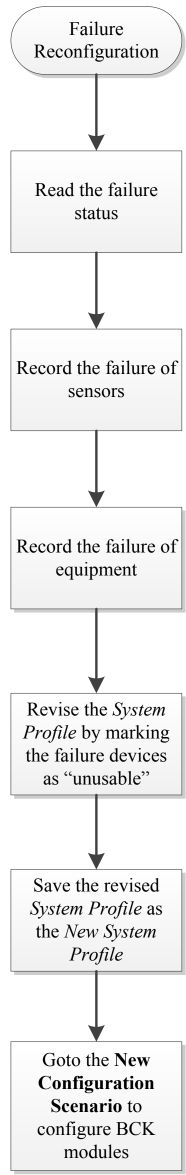


Figure 4-12 Process of failure case configuration.

4.3 Implementation

The Python programming language is envisioned as the primary programming language to develop the SCP. The coding structure needs to be developed based on the process flow diagrams in Section 4.2. However, due to time limitations, the Python based programming has to be left as future work.

4.4 Summary and Discussion

In this chapter, a new control self-configuration framework has been developed. This framework is based on the BCK database and given system profile. The current version of the self-configuration framework is categorized into four different scenarios: new system configuration, retrofit configuration, BCK database upgrade reconfiguration, and the Sensors & equipment failure reconfiguration. This work focuses on the first scenario since it is the foundation of other three scenarios. Detailed process maps have been developed, which illustrates each steps of the self-configuration process. Due to the limitation of time, the programming of the self-configuration program (SCP) is left as future work. The SCP will be developed in Python and it will utilizes the system profiles to self-configure the building control modules by interacting with the BCK database.

Meanwhile, the self-configuration process developed in this work is based on the observation of how human knowledge is accumulated, stored, managed and reused. Thus, although this process is developed for HVAC control knowledge self-configuration, the concept might be potentially applicable to other types of systems' self-configuration.

CHAPTER 5

DEVELOPMENT OF A SIMULATION PLATFORM BASED ON DYNAMIC HVAC SYSTEM AND COMPONENT MODELS

This chapter presents the development of a dynamic HVAC simulation platform in MATLAB/Simulink for testing control configuration results. As an initiating effort, the simulation platform is composed of basic modular HVAC components, including conduit, damper/valve, fan/pump, flow merge, flow split, heating coil, cooling coil, and zone. These modules are developed into Simulink customized blocks. The simulation platform is capable of calculating flow rates of fresh air, exhaust air, and return air based on system characteristic and fan curve. This platform also can support the simulation of customizable basic/advanced control strategies. This customized Simulink block library is then used to construct the simulation platform for the integrated case study in Chapter 6. A revised version of this chapter has been published in *Energy and Buildings*, 2014, 68, Part A, 376-386.

Nomenclature

$\dot{C}$	heat exchange rate (kJ/s.K)
C_p	specific heat (kJ/kg.K)
K_0	a function of authority A
a_{0-4}	coefficients for head vs. flow curve
e_{0-4}	coefficients for efficiency curve
r_{fin}	ratio of fin area to total surface area
ΔP	pressure differential, $P_{\mathrm{o}} - P_i$
h	convection coefficient (W/m^2K) or specific enthalpy (kJ/kg)
A	area (m^2), or authority of a valve
C	heat capacitance (kg/K), or dimensionless coefficient with subscript *f, or h*
D	fan blade diameter (m)
E	electricity consumption (kW)
I	integral gain
K	flow resistance coefficient (1/kg m)
L	length of the conduit (m)
N	rotational speed (rps)
NTU	number of transfer units
P	pressure (kPa), or proportional gain
R	heat transfer resistance (K/W), or reference value
T	temperature (°C)
U	overall heat transfer coefficient (kW/m^2 °C)
V	volume (m^3)

W	humidity ratio (kg_w/kg_{da})
X	output of PI controller
Y	actual value for PI controller
d	relative position ranging from 0-1.
m	mass flow rate (kg/s)
q	heat gain from occupants, or light, or equipment, etc. (W)
t	simulation time (second)
w	weighting factor for linear term of flow resistance coefficient K
ε	heat exchange effectiveness
η	efficiency
λ	leakage parameter
ρ	density (kg/m^3)

Subscripts

0	center of component
a	air
amb	ambient air
c	coil
cc	cooling coil
con	conduit
dps	dampers
duc	duct
ea	exhaust air

f	flow
fa	fresh air
fan	fan
fil	filter
fin	fin
h	pressure head
hc	heating coil
i	inlet
ia	inlet air
iw	water inlet
j	numerical term
l	lighting
o	outlet
oa	outlet air
oass	outlet air at steady state
ow	water outlet
owss	outlet water at steady state
p	people
r	roof
ra	recirculated air
sa	supply air
sur	surface, or surroundings
w	water
w1	East, and West walls
w2	North, and South walls

z	zone
zs	setpoint at the zone
z0	uncontrolled zone

Superscripts

*	wet condition

5.1 Introduction

In most cases, tests of new control algorithms in the field might not be suitable because it would be time-consuming, costly, and involved with safety risks. Simulation tools/platform can provide an opportunity to evaluate and compare different control strategies in a virtual environment. It has also been seen as promising tools for establishing the baseline (or baseband) performance predictions, which can be used to monitor performance or detect fault (Trčka and Hensen 2010). Furthermore, simulation analysis could help identify potential operational errors due to any ill design, before actual implementation of new control strategies in the field. Specifically, dynamic HVAC equipment models are useful in energy management for studying the optimal control strategies (Bourdouxhe, Grodent et al. 1998). In order to simulate the dynamic performance of HVAC systems, there would be two critical steps.

The first step is to develop dynamic models, which are a major focus among HVAC research fields. In past decades, a large number of dynamic models have been developed. The reference manual for the HVACSIM+ software documented the dynamic models for 26 types of HVAC components (Clark 1985). The *ASHRAE reference guide for dynamic models of HVAC equipment* reviewed the dynamic models developed before 1998 (Bourdouxhe, Grodent et al. 1998). The engineering reference for the DOE software EnergyPlus provide steady-state/discrete model equations for most of the HVAC components. Recently, (Zhou and Braun 2007) developed a simplified dynamic model for chilled-water cooling and dehumidifying with low computation requirement. Therefore, the availability of those models provides an opportunity to study dynamic performance of HVAC system through simulation as well as developing optimal control strategies.

The second step is to program the dynamic models into a simulation model through software. To implement the dynamic models for simulation, computational tools need be selected. There are two types of simulation approaches. The first type is the signal-oriented modeling, applied in most of the traditional modeling tools. One of the representatives is Simulink®. It is developed by MathWorks™, and used as a commercial tool for modeling, simulating and analyzing multidomain dynamic systems (MathWorks 2012). It provides customizable block for modular development and control system simulation toolbox, which mostly used by researchers and developers for control system analysis. Its solver is procedural based. Based on Simulink, SIMBAD (SIMulator of Building And Devices) has been developed for more than a decade (Husaunndee, Lahrech et al. 1997), but it is not free (cost about €3000 in 2013). Besides, it failed to provide easy accesses to view embedded equations or to upgrade the HVAC system component libraries. The second type is the object oriented modeling approach, the most popular tool is Dymola based on Modelica language. Modelica-based libraries for building energy and control systems have been developing by Lawrence Berkeley National Laboratory, and it is free (Wetter 2009). Nevertheless, the Dymola software itself is neither free, nor as popular as Simulink for control studies. Dymola can solve a systematic of equations without considering the computational order, but the models can become complex, and may cause simulation problems (Zupančič and Sodja 2011). On the other hand, Simulink modeling requires considerable engineering skills, but it is more suitable for design and implementation of control algorithms (Zupančič and Sodja 2013). Although many of the modeling methodologies used in Simulink® can be translated into a form for use in Modelica, it is not always possible to provide an equivalent methodology in Modelica (Dempsey 2003). In the development of controllers the signal-oriented modeling, as applied in MATLAB/Simulink, can show its strengths against the object-oriented modeling. Therefore, Simulink® is chose for the modeling purpose.

In practices, Simulink based HVAC modeling has been popular for analyzing the dynamic response and control algorithm development (Underwood 1999; Riederer 2005). Recently, several examples of Simulink based simulation studies have also been conducted. Dynamic models have been developed by (Tashtoush, Molhim et al. 2005), with cooling coil model based on empirical regression equations. (Soyguder, Karakose et al. 2009) compared the PID, fuzzy-PD and self-tuning PID-type fuzzy adaptive controllers through Simulink based simulation. (Congradac and Kulic 2009) simulated the performance of genetic algorithm based carbon dioxide (CO_2) concentration control, and validated the result using EnergyPlus. (Wu, Melnik et al. 2007) developed a mathematical model to simulate airflow control system of ventilation units in Simulink (with no flow split or flow merge for any recirculate air), and compared with lab-scale experiment result. However, most of those studies failed to consider several practical control components (e.g., damper, valve, etc.). More critically, none of the studies considered the flow rates determination by matching the HVAC system characteristic based on dampers openness together with the fan curve in a recirculated air loop. This pressure balance was not yet considered in the previous studies, even in the SIMBAD.

Therefore, this chapter discusses the development of dynamic HVAC system and components simulation platform for HVAC system and control performance analysis. It initializes an effort to create an open-access library in Simulink and demonstrates how to build up simulation models through a case study. In this simulation platform, airflow rates are determined based on HVAC system characteristics and fan curve. Specifically, Section 5.2 provides an overview of the simulation platform development. Dynamic equations for the basic HVAC components are included in Section 5.3. Section 5.4 described the method of flow calculation based on system characteristics and fan curve. Section 5.5 used part of the developed HVAC modules to demonstrate the process of developing dynamic simulation.

5.2 Simulation Platform Overview

The simulation platform proposed here includes two major parts, the physical plant modules, and the control sequence modules. Since the physical plant modules are fundamental for any further control algorithm simulation, the initial effort focuses on the development of physical plant modules.

In order for extensible application and modular reuse, each Simulink block for HVAC component is developed in a generalized modular manner. Those modules are included in a customized Simulink library and are easily to access, implement, and update. Since it was impossible to develop all the available models into Simulink modules at this point, this chapter started with a basic system consists of a number of major components, including conduit, damper/valve, fan/pump, flow merge, flow split, heating coil, cooling coil, and zone. Therefore, this work is not intended to be a completed project, but instead, is to initiate a platform and a demonstration of creating and sharing of Simulink modules among researchers, developers, and designers.

The information for each module is arranged as model input, output, related equations, control variable, manufacture parameters, module adjustable parameters, and calculated parameters. This classification is aimed to distinguish different variables based on their functions for modeling and future operational optimization. For each Simulink module, the information flow is programmed from model inputs, model parameters, control variables, model outputs. For most Simulink modules of the HVAC components, there are inputs and outputs for air properties, which is the major information flow. For example, the air-water cooling coil includes both the air property and water property information flows. Those two portions of the dataflow are purposely merged together for each module. This will provide a clear look and easier understanding for later implementation.

5.3 Customized Simulink Library for HVAC System

This section describes the initial effort to create a customized Simulink library for basic HVAC component models, including heating coil, cooling coil, zone, fan, flow split, flow merge, air duct, etc. Detailed mathematical models are also included.

5.3.1 Mathematical Models for Modular HVAC Components

5.3.1.1 Module 1: Fan or Pump (variable speed drive and constant speed drive)

Fan or pump model equations are extracted from the Type 1: Fan or Pump in HVACSIM+ reference book (Clark 1985). Those equations are used in the Simulink block programming. The pressure change from the fan (or pump) is calculated based on mass flow rate and the configuration of fan (or pump) (1). Since this model can be applied for either pipe or duct, so no subscript for the air or water is provided here.

$$P_o = P_i + 0.001 C_h \rho N^2 D^2 \tag{1}$$

where the dimensionless pressure head coefficient $C_h = a_0 + a_1 C_f + a_2 C_f^2 + a_3 C_f^3 + a_4 C_f^4$; a_{0-4} is the coefficients for head vs. flow curve; the dimensionless flow coefficient $C_f = \frac{m_i}{\rho N D^3}$.

Temperature change through the fan (or pump) is also considered based on fan (or pump) efficiency through conservation of energy (2).

$$T_o = T_i + \frac{P_o - P_i}{\rho C_p}\left(\frac{1}{\eta_{fan}} - 1\right) \tag{2}$$

Electric power (kW) and fan (or pump) efficiency are calculated in (3) and (4).

$$E_{fan} = \frac{m_i \Delta P}{\eta_{fan} \rho} \tag{3}$$

$$\eta_{fan} = e_0 + e_1 C_f + e_2 C_f^2 + e_3 C_f^3 + e_4 C_f^4 \tag{4}$$

where e_{0-4} is the coefficients for efficiency curve.

5.3.1.2 Module 2: Mixing box (flow merge)

For the mixing of two flow streams (either water or air), equations from Type 4: Flow Merge (Clark 1985) were used. The flow rate and temperature are determined based on mass and energy balances though equations (5) and (6).

$$m_o = m_{i1} + m_{i2} \tag{5}$$

$$T_o = \frac{T_{i1} \cdot m_{i1} + T_{i2} \cdot m_{i2}}{m_o} \tag{6}$$

Pressure change is determined based on the flow rate and duct (or pipe) characteristics (7).

$$P_o = P_{i1} - \frac{K}{2}\left[sign(m_{i1})m_{i1}^2 + sign(m_o)m_o^2\right] \tag{7}$$

Particularly, if it is airflow, the humidity ratio can be calculated using (8).

$$W_{oa} = \frac{W_{ia1} \cdot m_{ia1} + W_{ia2} \cdot m_{ia2}}{W_{oa}} \tag{8}$$

5.3.1.3 Module 3: Splitting box (flow split)

Multiple flow split at the splitting box, and the equations from Type 6: Flow Split by (Clark 1985) were used. Firstly, one of the split flow rate is determined by pressure difference (9).

$$m_{o1} = \sqrt{\frac{|P_0 - P_{o1}| m_i}{0.5K}}\, sign(P_0 - P_{o1}) \tag{9}$$

Then, the mass balance is used to determine the flow rate (10), while assuming the air is well mixed and the temperature and humidity ratio remain the same through splitting.

$$m_{o2} = m_i - m_{o1} \tag{10}$$

Pressure change is determined based on the flow rate and duct characteristic (11) and (12).

$$P_0 = P_i - 0.5Ksign(m_i)m_i^2 \quad (11)$$

$$P_{o2} = P_0 - 0.5Ksign(m_{o2})m_{o2}^2 \quad (12)$$

where P_0 represents the pressure at the center of component.

5.3.1.4 Module 4: Heating coil

Heating coil related equations are summarized based on simplified heating coil model from (Clark 1985). The steady state air and water outlet temperatures are calculated based on heat balance by (13) and (14).

$$T_{oass} = T_{ia} + \frac{(T_{iw}-T_{ia})\varepsilon C_{min}}{C_{p,a}m_{ia}} \quad (13)$$

$$T_{owss} = T_{iw} - \frac{(T_{oass}-T_{ia})C_{p,a}m_{ia}}{C_{p,w}m_{iw}} \quad (14)$$

where heat exchange effectiveness $\varepsilon = 1 - exp\left\{\frac{\left[exp\left(-\frac{C_{min}}{C_{max}}n\{NTU\}\right)-1\right]}{\frac{C_{min}}{C_{max}}n}\right\}$; $n = (NTU)^{-0.22}$; $NTU = \frac{U_cA_c}{C_{min}}$; $C_{min} = min(C_{p,a}m_{ia}, C_{p,w}m_{iw})$; $C_{max} = max(C_{p,a}m_{ia}, C_{p,w}m_{iw})$.

Then, the dynamic outlet temperatures are determined by and (16).

$$\frac{dT_{ow}}{dt} = \frac{T_{owss}-T_{ow}}{\tau}(16)$$

where coil time constant $\tau = (\tau_c^{-1} + \tau_x^{-1})^{-1}$; τ_c is the capacitive term of coil time constant; and $\tau_x$is the coil flush time, $\tau_x = \frac{\rho V_{hc}}{m_w}$.

Pressure changes are determined by flow resistance coefficients (17), (18).

$$P_{ow} = P_{iw} - K_w m_{iw}^2 \quad (17)$$

$$P_{oa} = P_{ia} - K_a m_{ia}^2 \quad (18)$$

5.3.1.5 Module 5: Cooling coil

The cooling coil dynamic model was programmed based on the simplified cooling coil module developed in ASHRAE RP-1194 project (Zhou and Braun 2007; Zhou and Braun 2007). A coil with four rows in a counter flow fashion was selected. It includes two parts of heat exchange, water-coil, and coil-air. The water-coil heat exchange process is given by (19).

$$C_w \frac{dT_{ow}}{dt} + \dot{C_w}(T_{ow} - T_{iw}) + \frac{1}{R_w}(T_{iw} - T_{coi}) = 0 \quad (19)$$

where total heat capacitance of water is C_w; total heat capacitance associated with the water flow is $\dot{C_w}$; total thermal resistance for heat transfer between water and coil material is $R_w = \frac{1}{\varepsilon_w \dot{C_w}}$; water-side heat transfer effectiveness for the row is $\varepsilon_w = 1 - e^{-NTU_w}$; Number of Transfer Units for the water-side is $NTU_w = \frac{h_w A_w}{\dot{C_w}}$; total internal surface area of the tubes within the coiling coil row is A_w; water-side heat transfer coefficient is h_w.

For the coil-air exchange process, the calculation is given based on coil dry/wet conditions. When the coil is dry, the dynamic process of air-coil heat exchange is described in (20) and (21).

$$C_c \frac{dT_c}{dt} + \frac{1}{R_a}(T_c - T_{ia}) + \frac{1}{R_w}(T_c - T_{iw}) = 0 \quad (20)$$

$$T_{oa} = T_{ia} + \varepsilon_\mathrm{a}(T_c - T_{ia}) \quad (21)$$

where total heat capacitance of the coil is C_c; total thermal resistance for heat transfer between air and coil material is $R_a = \frac{1}{\varepsilon_a \dot{C_a}}$; air-side heat transfer effectiveness for the row is $\varepsilon_a = 1 - e^{-NTU_a}$; Number of Transfer Units for the air-side is $NTU_a = \frac{\eta_a h_a A_a}{\dot{C}_\mathrm{a}}$; convection coefficient for air-side heat transfer is h_a; overall fin efficiency for heat transfer is $\eta_a = 1 - r_{fin}(1 - \eta_{fin})$; the ratio of fin area to total surface area is $r_{fin} = \frac{A_{fin}}{A_{sur}}$; individual fin efficiency is η_{fin}.

When the coil is wet, (20) and (21) are rewritten as (22) and (23).

$$C_c \frac{dT_c}{dt} + \frac{1}{R_a^*}\left(h_{s,c} - h_{ia}\right) + \frac{1}{R_w}(T_c - T_{iw}) = 0 \tag{22}$$

$$h_{oa} = h_{ia} + \varepsilon_a^*(h_{s,c} - h_{ia}) \tag{23}$$

where total heat and mass transfer effectiveness for wetter surface is $R_a^* = \frac{1}{\varepsilon_a^* m_a}$; the effectiveness for combined heat and mass transfer is $\varepsilon_a^* = 1 - e^{-NTU_a^*}$; total air mass flow rate is m_a; Number of Transfer Units for the air-side is $NTU_a^* = \frac{\eta_a^* \mathrm{h}_a^* A_a}{\dot{\mathrm{C}}_a}$; overall fin efficiency for combined heat and mass transfer is $\eta_a^* = 1 - r_{fin}(1 - \eta_{fin}^*)$*; i*ndividual fin efficiency (wet condition) is η_{fin}^*; convection coefficient for air-side heat transfer under dehumidification is h_a^**;* inlet and outlet air enthalpy are h_{ia} and h_{oa}; saturation air enthalpy at the mean coil temperature (T_c*)* is $h_{s,c}$.

To calculate the heat exchange process, firstly, coil is assumed dry, and the dynamic variables will be calculated through equation (19), (20) and (21). Secondly, the condensation status at the i^{th} row of coil will be evaluated based on if the dew point temperature of the air outlet T_{dew} (calculated from each row based on dry assumption) is lower than the coil temperature of that row T_c. If it is lower, the i^{th} row of coil will be considered as wet, and then recalculated using enthalpy balance equation under wet condition with (19), (22), and (23). Otherwise, the coil will be considered as dry on the surface, and keep the dry coil calculation result.

Besides, to convert among air enthalpy, temperature, and humidity ratio, Equation 6, 32, 38 from Chapter 1 of the *2009 ASHRAE Handbook: Fundamentals* were used (ASHRAE 2009).

5.3.1.6 Module 6: Zone

The zone dynamic model is summarized based on zone model by (Tashtoush, Molhim et al. 2005). There was some unit inconsistency in the original equations, i.e., original unit given for the thermal capacitance (C_z, C_{w1}, C_{w2}, C_r) is kJ/°C, while the unit for heat transfer coefficient ($U_{w1}, U_r, U_{w2}, q(t)$) is W/m^2°C. Therefore, it is corrected here by convert W/m^2°C to kW/m^2°C. The dynamic heat balances in the zone, through the east and west walls, south and north walls, roof are calculated in (24), (25), and (27) separately.

$$C_z \frac{dT_z}{dt} = m_{sa}C_{p,a}(T_{sa} - T_z) + 2U_{w1}A_{w1}(T_{w1} - T_z)$$

$$+U_rA_r(T_r - T_z) + 2U_{w2}A_{w2}(T_{w2} - T_z) + q(t) \tag{24}$$

$$C_{w1} \frac{dT_{w1}}{dt} = U_{w1}A_{w1}(T_z - T_{w1}) + U_{w1}A_{w1}(T_{amb} - T_{w1}) \tag{25}$$

$$C_{w2} \frac{dT_{w2}}{dt} = U_{w2}A_{w2}(T_z - T_{w2}) + U_{w2}A_{w2}(T_{amb} - T_{w2}) \tag{27}$$

Internal heat gain from the occupancy, lighting fixtures, other devices are given in (28)

$$q(t) = q_p + q_l \tag{28}$$

5.3.1.7 Module 7: General Conduit

Based on the conduit model in HVACSIM+ (Clark 1985), the length of conduit (L) were added into the original models. Pressure change is calculated by (29).

$$P_i = P_o + sign(m)K_{con}m^2L \tag{29}$$

Steady state air temperature is firstly calculated by (30), and time constant is applied to calculate the dynamic air temperature (31).

$$T_{ss} = T_{amb} + (T_i - T_{amb})\exp(-\gamma) \tag{30}$$

$$\frac{dT_o}{dt} = \frac{T_{ss} - T_o}{\tau} \tag{31}$$

where, $\gamma = \frac{UAL}{mC_p}$, $\tau = \left[\frac{h_i}{h_i + h_o}\right] \frac{C_m}{mC_p} L$.

5.3.1.8 Module 8: Damper or Valve

The dynamic model for the damper/valve is based on Type 5: Damper or Valve in HVACSIM+ (Clark 1985), pressure changes is calculated in (32).

$$P_o = P_i - sign(m) K_{dps} m^2 \tag{32}$$

where flow resistance coefficient $K_{dps} = \frac{wK_0}{[(1-\lambda)d+\lambda]^2} + (1 - \mathrm{w})K_0\lambda^{2d-2}$; K_0 is a function of authority A; w is the weighting factor for linear term of K; λ is the leakage parameter; d is the relative damper position ranging from 0-1.

5.3.1.9 Module 9: PI controller

The Simulink library has already implemented PI controller. The output of the PI controller (X) is calculated from the difference of Reference value (R) and the actual value (Y) in (33) (MathWorks 2012).

$$X = P\left((bR - Y) + I\frac{1}{s}(R - Y)\right) \tag{33}$$

where P, I are the proportional, integral gains, separately, and b is the setpoint weight.

In practice, the tuning of P, I is essential since inappropriate values could cause simulation errors, system failure or damage.

Table 5-1 Summary of variables for each module.

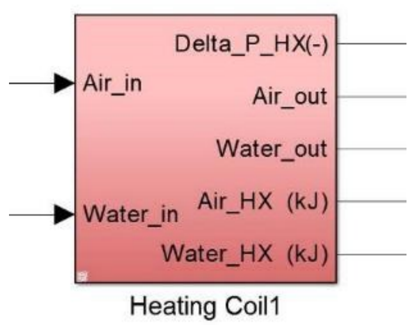
	Variable Category				
	Module input	**Calculated parameter**	**System parameter**	**Control variable**	**Module output**
Fan (or Pump)	T_i, m_i, P_i, W_i	C_h, C_f	D, a_{0-4}, e_{0-4}	N	T_o, m_o, P_o, η_{fan}, E_{fan}, W_o
Flow Merge	T_{i1}, m_{i1}, P_{i1}, T_{i2}, m_{i2}, P_{i2}, W_{ia1}, W_{ia2}	-	K	-	T_o, P_o, m_o, W_{oa}
Flow Split	T_i, m_i, P_i, W_{ia}	P_0	K	-	T_{o1}, P_{o2}, T_{o2} m_{o1}, m_{o2}, W_{oa}
Heating Coil	T_{ia}, m_{ia}, P_{ia}, W_{ia}, T_{iw}, m_{iw}, P_{iw}	NTU, C_{min}, C_{max}, τ, T_{oass}, T_{oass}	U_{coil}, A_{hc}, V_{hc}, τ_c, K_a, K_w	T_{iw}, m_{iw}	T_{oa}, m_{oa}, W_{oa}, T_{ow}, m_{ow}, P_{oa}, P_{ow},
Cooling Coil	T_{ia}, m_{ia}, P_{ia}, W_{ia}, T_{iw}, m_{iw}, P_{iw}	R_a, NTU_w , ε_w, R_w, R_a^*, ε_a^*, η_a^*, NTU_a^*, h_a^*, h_w, h_{ia}, h_{oa}, $h_{s,c}$	A_{fin}, A_{sur}, η_{fin}, η_{fin}^*, r_{fin},	T_{iw}, m_{iw}	T_{oa}, m_{oa}, W_{oa}, T_{ow}, m_{ow}, P_{oa}, P_{ow}
Zone	T_{amb}, $T_{ia}(T_{sa})$, $m_{ia}(m_{sa})$, P_{ia}, W_{ia}, N_p	$q(t)$	C_{w1} , C_{w2} , C_z , C_r , U_{w1}, U_{w2}, U_r, A_{w1}, A_{w2}, A_r, q_p, q_l	-	$T_{oa}(T_{ra}, T_z)$, m_{oa}, W_{oa}, P_{oa}, T_r, T_{w1}, T_{w2}
Conduit (air or water)	T_{ia}, m_{ia}, P_{ia}, W_{ia}	-	L	-	T_{oa}, P_{oa}, m_{oa}, W_{oa}
Damper (or Valve)	P_i, m	K, K_0	A, W_f	D	P_o
PI Controller	R, Y	-	P, I, b	-	X

To summarize, modeling blocks, variables and parameters were categorized into five types based on their function, namely, module inputs, calculated parameters, system parameters, control variables, and module outputs (see Table 5-1). It is clear that the control variables are limited to a number of modules, i.e., fan (or pump) rotational speed, damper (or valve) positions, ON/OFF switches, etc.

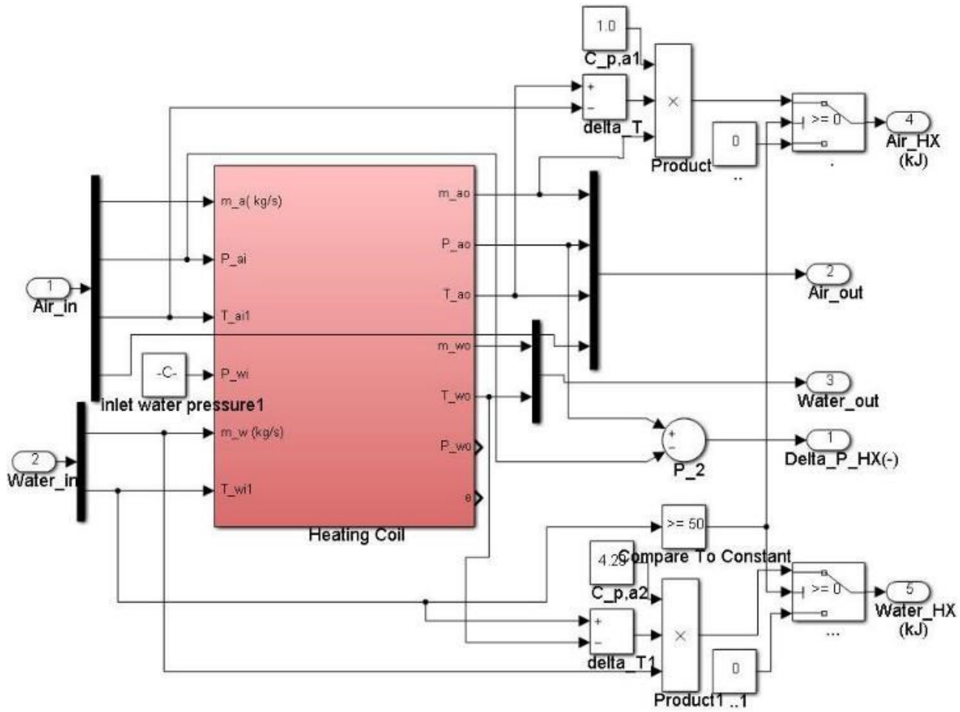
5.3.2 Simulink Programming

In order to initiate the simulation platform development, the customized HVAC blocks are designed by modularizing representative HVAC component in a generalized style. The major dataflow in those blocks is the change of fluid properties. Three major fluids, namely, air, water, and steam, were group separately in each block during the programming process.

For the air group, the air mass flow rate, pressure, temperature, pollutant(s) concentration(s), and humidity ratio are grouped together. For the water group, the water flow rate, pressure, and temperature are grouped together. For the steam group, the steam flow rate, pressure, and temperature are grouped together. For each HVAC component block, there would be one or more groups discussed above. An example of this setting is given in Figure 5-1, a) shows the out layer of heating coil block, while b) shows how the air, water related information (variables) are separately grouped at block inputs/outputs.



(a)



(b)

Figure 5-1 An example of grouping the information based on fluid type in heating coil (a) Out layer of heating coil block. (b) Configuration of air, water related information.

Therefore, the customized HVAC blocks are programmed in this fashion to keep the flow properties grouped together and passed through one block to another based on flow type.

5.4 System Characteristic and Flow Rate Calculation

To determine airflow rate, the system characters (pressure drop) must be calculated and matched with the fan capacity (pressure rise). Air duct system characteristic is a function of airflow rate, and the airflow rate determines the pressure rise through fans. By selecting the ambient pressure as a reference, pressure changes across the whole duct system can be calculated as following (34):

$$\Delta P_{fan} + \sum_{j=1}^{n} \Delta P_j = 0 \tag{34}$$

Where ΔP_{fan} is the pressure rise (+) provided by the fan, $\sum_{j=1}^{n} \Delta P_j = \Delta P_{dps} + \Delta P_{cc} + \Delta P_{hc} + \Delta P_{duc} + \Delta P_{fil} + \Delta P_z$, represents the pressure drop (-) through the ductworks (from the air inlet to exhaust).

Since there is recirculated air feedback to the flow merge box, the flow rate of recirculated air (m_{ra}) can be calculated based on (35). It utilized the pressure difference at the recirculated branch (i.e., between the airflow split block and flow merge block) to determine the airflow rate.

$$\Delta P_{recirculated} = sign(m_{ra}) K m_{ra}^2 \tag{35}$$

If we assume air leakage is negligible, then the total air system mass balance can be described by (36).

$$m_{ea} = m_{fa} = m_{sa} - m_{ra} \tag{36}$$

By taking all the detailed pressure calculations (1), (7), (10), (11), (17), (29), (32) into (34), and solving equations (34), (35), and (36)(36) together, the airflow rates at exhaust, supply and return air ducts (m_{ea}, m_{sa}, m_{ra})can be calculated.

5.5 Demonstration of Application Process

This section aims to demonstrate how to implement the Simulink blocks from customized HVAC library for simulation analysis. A specified case study is employed here to demonstrate the application process (detailed in Figure 5-2), which includes 1) identify system component, 2) configure individual HVAC block from the library, 3) develop HVAC system without control, 4) develop the control system, 5) run simulation, and 6) analyze result. Each step is further explained in the following demonstration case study.

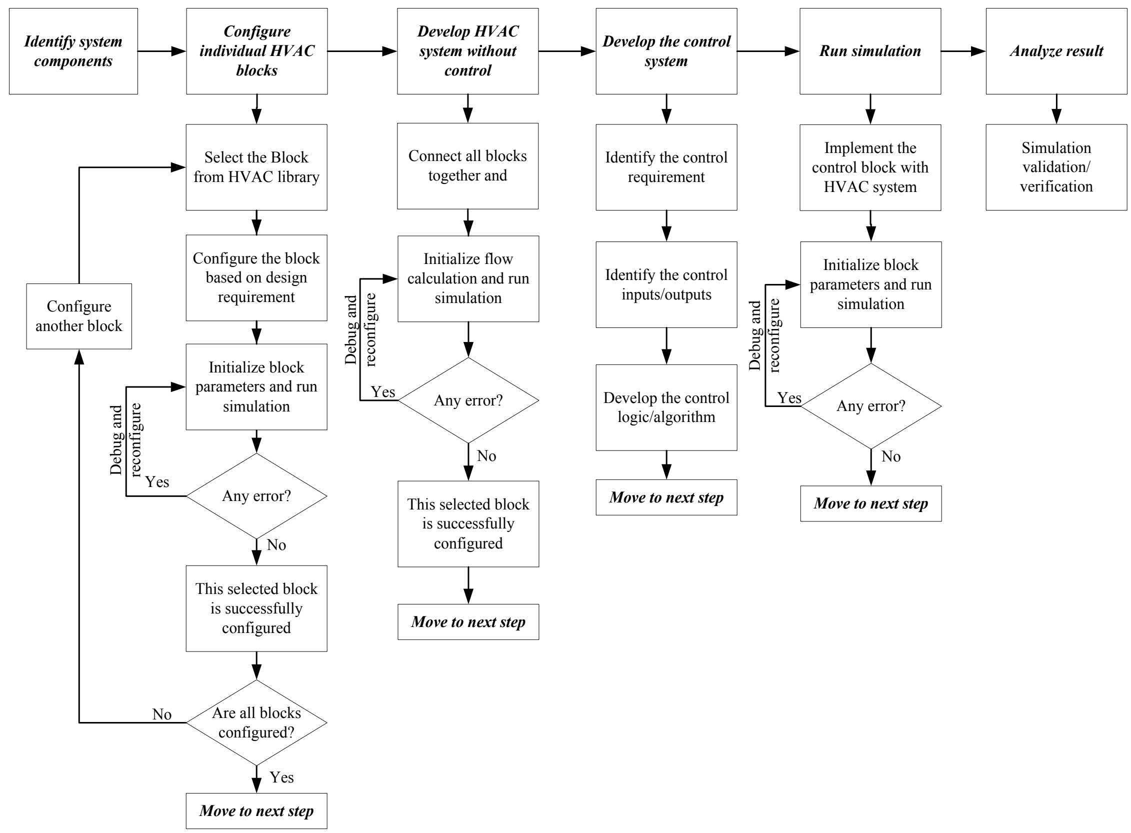


Figure 5-2 Flow chart of developing the Simulink model simulation.

It is assumed that it is a single zone system (without window) includes heating coil, cooling coil, conditioned zone, supply fan, conditioned zone, air duct, flow merge and split. The interior dimension of the zone is 12 m by 6 m with a height of 3.5 m. The interior zone volume is assumed approximate 252 m^3. It is also assumed that there are 10 occupants during the office hour (8:00 -17:00).

5.5.1 Identify System Component

The first step is to identify the basic HVAC modular components based on HVAC system schematic. From Figure 5-3, this study case need Simulink blocks for heating coil, cooling coil, air handling unit, supply fan, conditioned zone, flow merge, and flow split. Before using the available blocks in the HVAC library, it is highly recommended to understand the mathematical model before working on the programming process. For relatively complex blocks, e.g. cooling coil, it is also necessary to do a hand calculation of the mathematic model and compare to the simulation result to see if they match with each other.

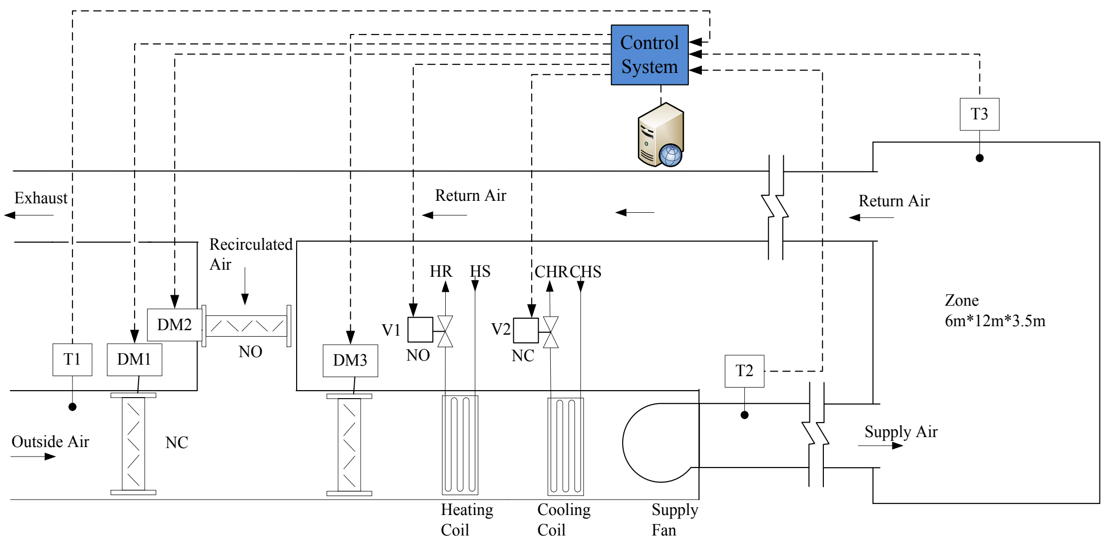


Figure 5-3 Proposed single-zone HVAC system schematic for case study.

5.5.2 Configure Individual HVAC Blocks

The parameters of these blocks need to be adjusted to meet the design requirement. It requires the proper sizing of HVAC components (i.e. supply fan, duct, damper, heating coil, cooling coil, valve.).

5.5.3 Develop HVAC System without Control

After each block has been successfully tested, they need to be connected based on flow path. It is highly recommended to build and test the air-handling unit (AHU) before creating the whole air loop with flow merge and flow split. After the AHU is tested, one can test the flow merge and flow split together with the ductwork but without the AHU to see if the flow calculation works correctly. Then, the AHU can be added into the air loop system for simulation without any control.

Once the HVAC blocks are connected, the open loop system (without the control) should be tested in simulation. If it fails with errors, one should follow the debugging procedure to solve the problem. Based on author's experience, it is usually related to misconnection of wires, incorrect setting or missing of initial condition block. Once the error is clear, the simulation result needs to be analyzed to see if the open loop system is within a reasonable range.

5.5.4 Develop the Control System

To develop the control system, one can either use available Simulink control blocks or create customized control blocks using Simulink or MATLAB functions. This process requires the developer to firstly identify the inputs, outputs, and control loops of the control systems. In this case study, it is proposed to include several control scenarios including PI controllers based closed loop controls to maintain supply air pressure (#1) and zone air temperature (#2, #3), forward controls to modulate OA/RA dampers positions (#4) and to modify zone temperature setpoint (#5). Inputs, outputs, and control loops of the two control modes are summarized in Table 5-2.

Table 5-2 Example of typical local controls (loops) used in case study.

Number	Control name	Control type	Control algorithms	Inputs	Control variable
#1	Supply air pressure maintain	Closed loop	PI control	$P_{sa}, P_{sa_{set}}$,	d_{sa}
#2	Heating coil water flow rate	Closed loop	PI control	$T_z, T_{z_{set}}$	m_{hw}
#3	Cooling coil water flow rate	Closed loop	PI control	$T_z, T_{z_{set}}$	m_{cw}
#4	OA% control	Open loop	Logic forward	$time$	d_{oa}, d_{ra}, d_{ea}
#5	Schedule based reset	Open loop	Logic forward	$time$	$T_{z_{set}}$

The control blocks were developed based on damper position and zone temperature reset based on office schedule, with heating coil water flow rate and cooling coil water flow rate modulation based on zone temperature setting point. Once the control blocks are built, they will be integrated with the HVAC model. Here a system with highlighted control system blocks (at the bottom) is presented in Figure 5-4.

5.5.5 Run Simulation

To run the simulation, the models need to be initialized first. The Initial Condition (IC) blocks are applied at the upstream of dataflow for cooling coil blocks, flow determination process, etc. The setting value of the IC block depends on the reasonable initial calculation point for the related equations, so that the simulation can quickly converge. Here, a sine wave was used to represent the ambient air temperature profile. Meanwhile, inappropriate parameters can slow down the simulation speed, and result in system instability. Therefore, it is necessary to set appropriate proportional and integral gains for those PI controllers. Trial and error method is used to tune the PI controllers.

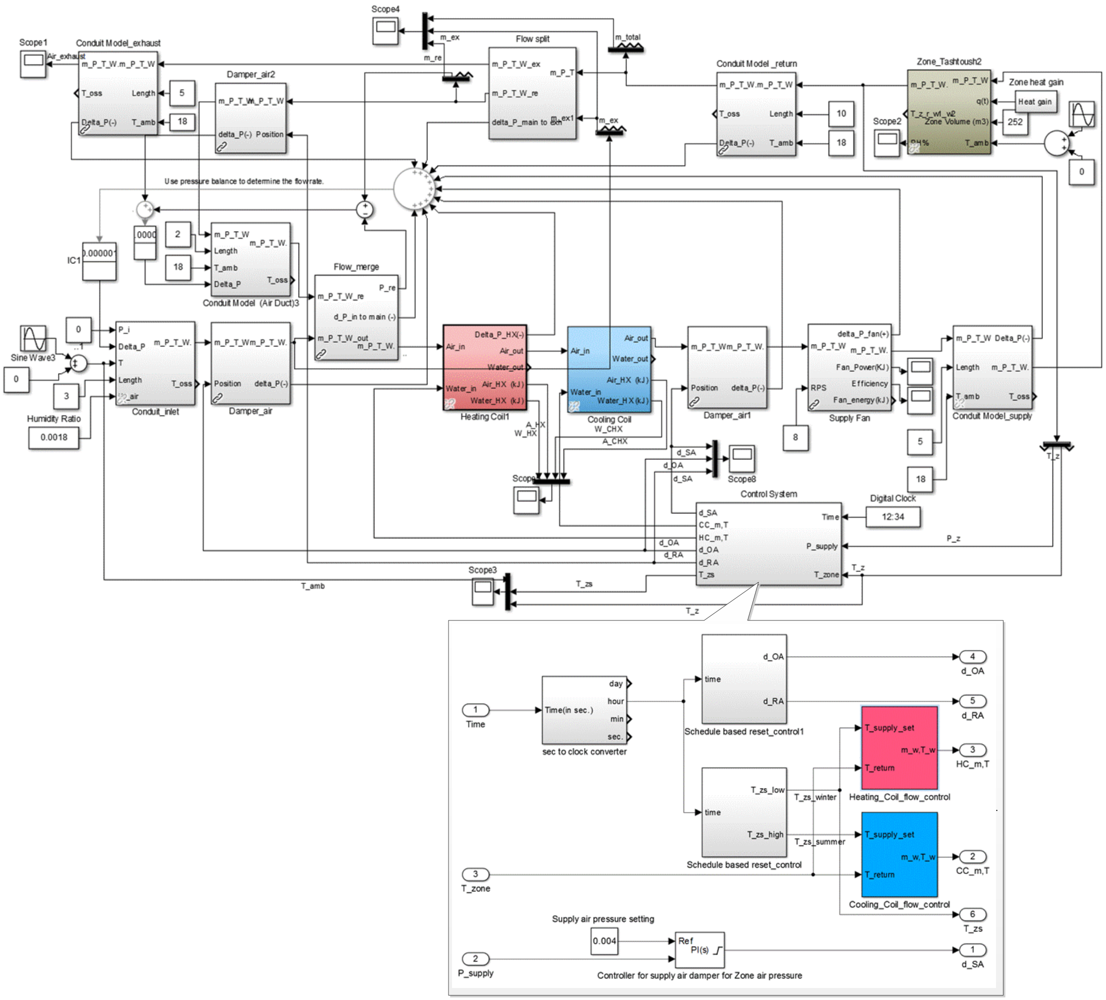


Figure 5-4 Schematic of simulated system in Simulink®.

Secondly, it is important to configure the model configuration parameters, including the simulation time, solver options, etc. For the solver option, it is recommended to select variable-step so that the simulator can adjust the running step size based on converging requirement. Then, one can start with ode45 for simulation; if it is too slow due to the model complexity, the solver can be changed to ode15s (stiff/NDF). The relative tolerance is another parameter to adjust if the simulation fails to converge. It is recommended to use smaller relative tolerance (e.g., 1e-7) to solve this problem.

Then, set the range of time for the simulation and click the run button. Through the simulation, Simulink tools can be used to monitor the values for inputs and outputs at each block by using the data display tool from the simulation tab. By enabling this function, one can click any block and display the simulated real-time values during the simulation process. Here, several control scenarios were demonstrated through simulations (see Table 5-3). The system configuration and programming for this study utilizes MATLAB 2012b. The PC used for the simulation has a processor with Inter® Core™ 2 Quad CPU (Q9400 @2.66 GHz).

Table 5-3 List of simulation scenarios conducted for the case study.

Scenarios	**Control/uncontrolled parameters**	**Environmental conditions**	**Controls[1]**
1, Uncontrolled	$\mathrm{d_{oa}} = 0.5, \mathrm{d_{ra}} = 0.8,\ \mathrm{d_{sa}} = 0.7,$	$\mathrm{T_{amb}} = 6\sin(7.30 * 10^{-5}t - 2.1rad) + 30$ (summer) $\mathrm{T_{amb}} = 6\sin(7.30 * 10^{-5}t - 2.1rad) + 0$ (winter) $\mathrm{W_{amb}} = 0.018\mathrm{kg_w/kg_{da}}$ (summer), $0.0018\mathrm{kg_w/kg_{da}}$ (winter); For uncontrolled condition $\mathrm{T_{sur}} = 30°\mathrm{C}$ (summer), 0°C (winter); For controlled condition: $\mathrm{T_{sur}} = 25°\mathrm{C}$ (summer), 18°C (winter);	None
2, Fixed damper position and zone temperature setting, and supply air pressure control	$\mathrm{p_{sa}} = 0.004\mathrm{kPa}$, $\mathrm{d_{oa}} = 0.5, \mathrm{d_{ra}} = 0.8, \mathrm{T_{zs}} = 24°\mathrm{C}$ (summer), $\mathrm{T_{zs}} = 22°\mathrm{C}$ (winter),		1, 2, 3
3, Zone temperature and damper positions control based on schedule.	$\mathrm{p_{sa}} = 0.004\mathrm{kPa}$, occupied: $\mathrm{T_{zs}} = 24°\mathrm{C}$ (summer), $\mathrm{T_{zs}} = 22°\mathrm{C}$ (winter) $\mathrm{d_{oa}} = 0.4, \mathrm{d_{ra}} = 0.7$; unoccupied: $\mathrm{T_{zs}} = 28°\mathrm{C}$ (summer), $\mathrm{T_{zs}} = 16°\mathrm{C}$ (winter), $\mathrm{d_{oa}} = 0.1, \mathrm{d_{ra}} = 0.6,$		1, 2, 3, 4, 5,

[1]: number listed in this column refers to number in Table 5-2.

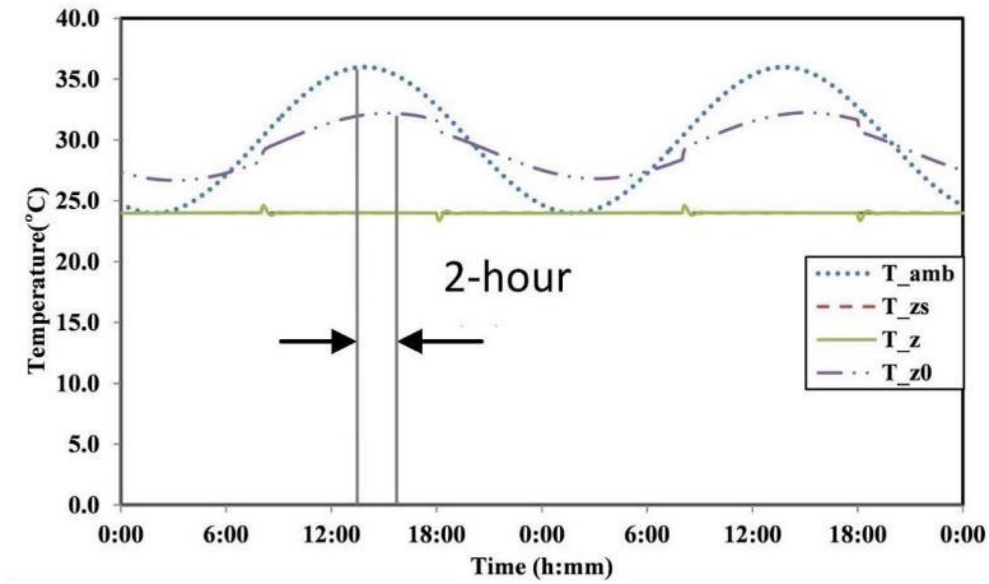
5.5.6 Result and Discussion

Simulation results for the three scenarios are presented in following figures and several key points are noted here. First, the uncontrolled system indicated that the zone thermal capaicitance would have an impact on the delay of the uncontrolled zone temperature (T_{z0}) of about 2 hours for the peak load (marked in Figure 5-5 (a)). During the summer, T_{z0} was almost greater than 26°C and up to 30°C when the ambient air temperature (T_{amb}) ranged from 24 to 36°C. During the winter, T_{z0} was below 15°C, while T_{amb} ranged from -6 to 6°C (Figure 5-5).

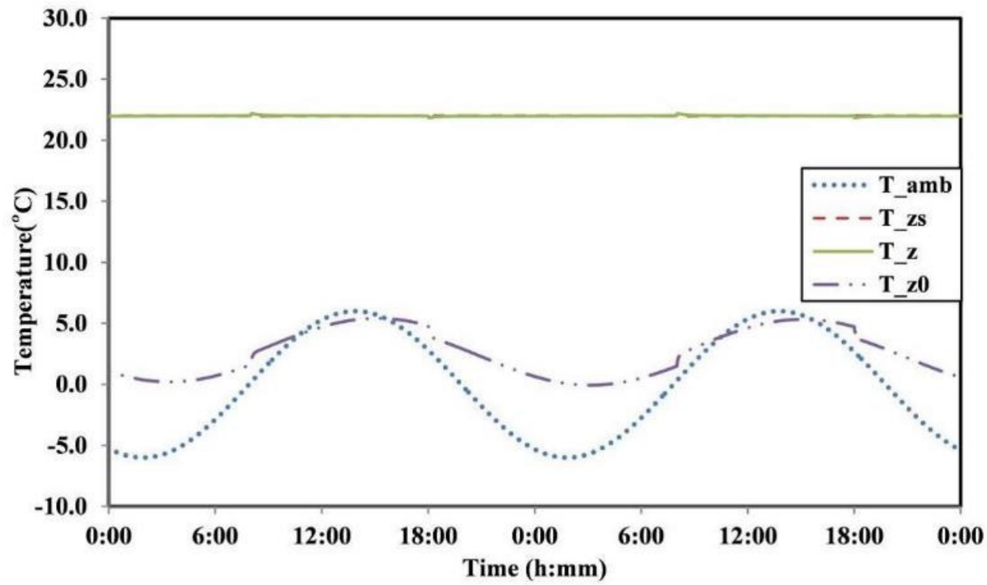
With fixed zone temperature control under fixed damper position, the zone air temperature (T_z) can be well maintained at the setpoint during the summer and winter condition (Figure 5-5). Due to the fixed setting of zone temperature and damper positions, cooling was always required during the summer and heating during the winter (Figure 5-6).

For schedule based reset, T_z can be mostly maintained with settings during the occupied period (Figure 5-7). While the OA/RA dampers positions (d_{OA}, d_{RA}) were reset based on schedule, the SA damper position (d_{SA}) was controlled to maintain supply air pressure requirement (Figure 5-8 (a)). The flow rates thus changed along with the damper modifications (Figure 5-8). For the energy consumption, when the zone is not unoccupied during the summer, the temperature setpoint reset and damper position reset actually shut down the cooling system and save a huge energy, comparing Figure 5-9 (a) with Figure 5-6 (a). Similarly, it would also save energy during the heating season comparing Figure 5-9 (b) with Figure 5-6 (b).

Figure 5-9 also indicates some spikes of heat exchange rate. This was caused by overshoot of coil water flow responding to the changes of temperature and damper position settings. Since the same PI values were used in both control scenarios, it would be the appropriate values for fixed control becomes inappropriate for reset control. Therefore, adaptive PI controller or self-tuning strategies should be considered in the future simulation.

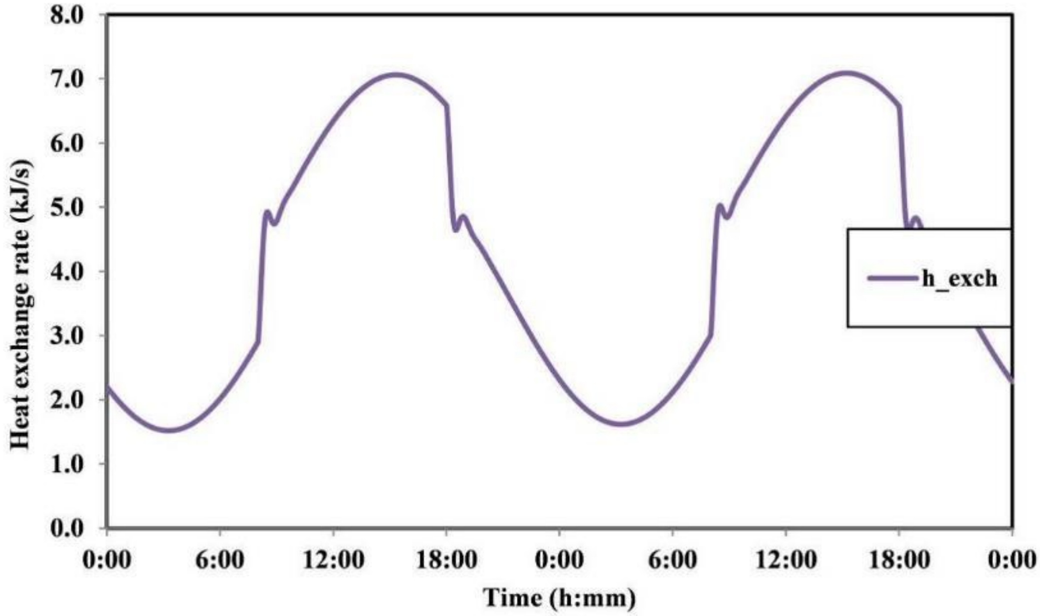


(a)

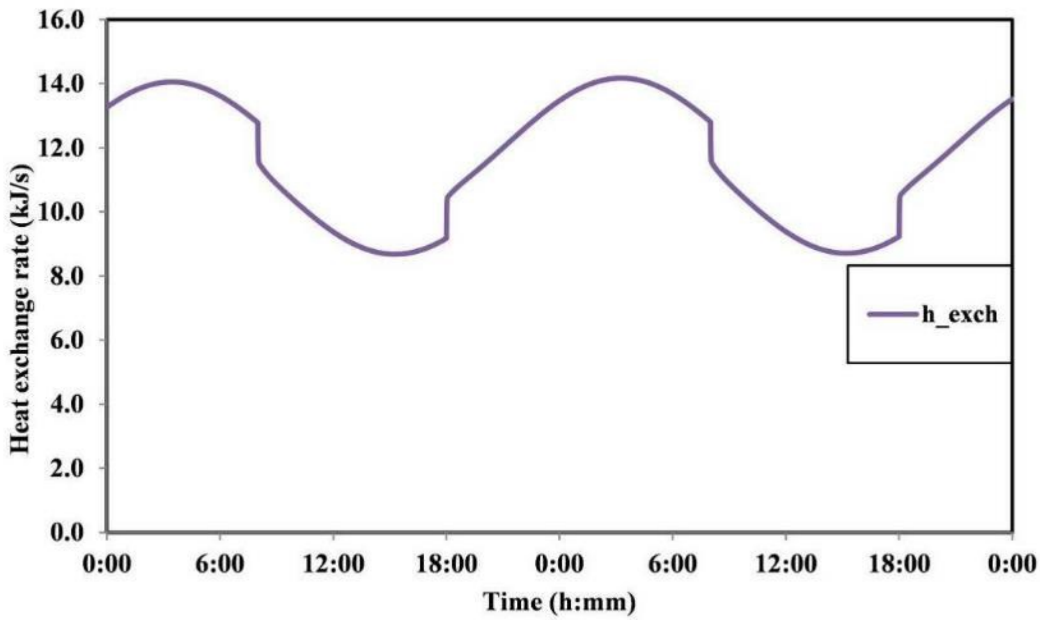


(b)

Figure 5-5 Two-day simulation results of air temperature without control compared with fixed control (a) summer condition, (b) winter condition.

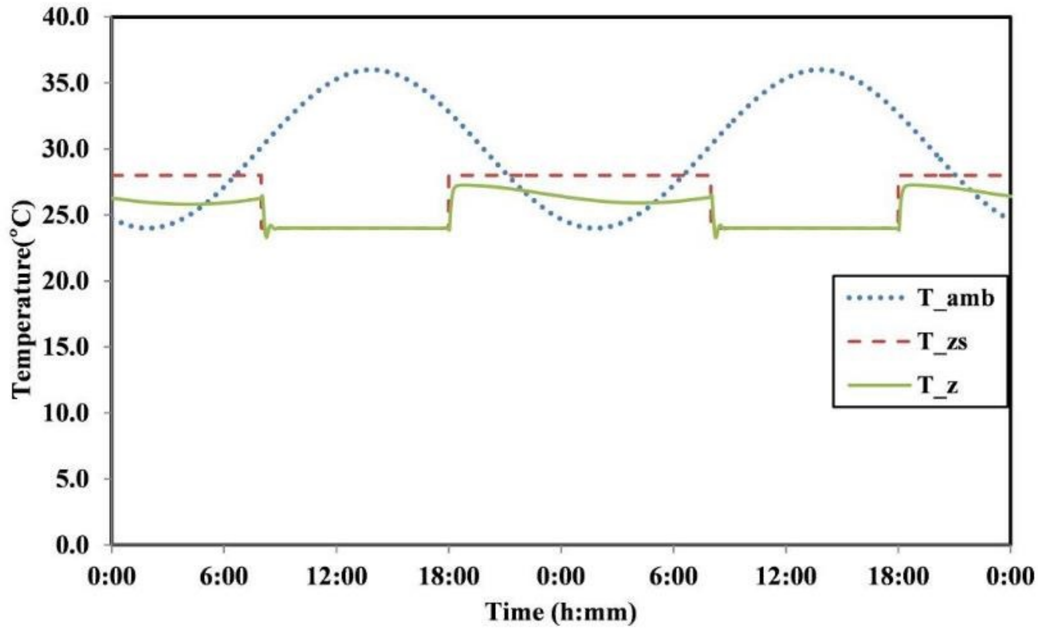


(a)

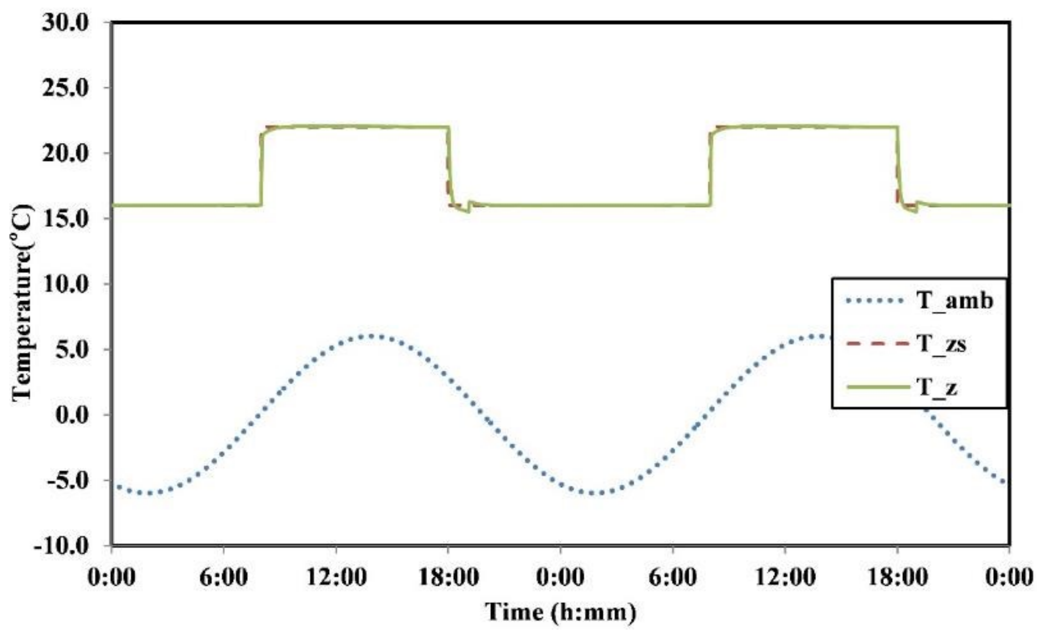


(b)

Figure 5-6 Two-day simulation results of heat exchange rate at the fixed control (a) summer condition for cooling coil, (b) winter condition for heating coil.

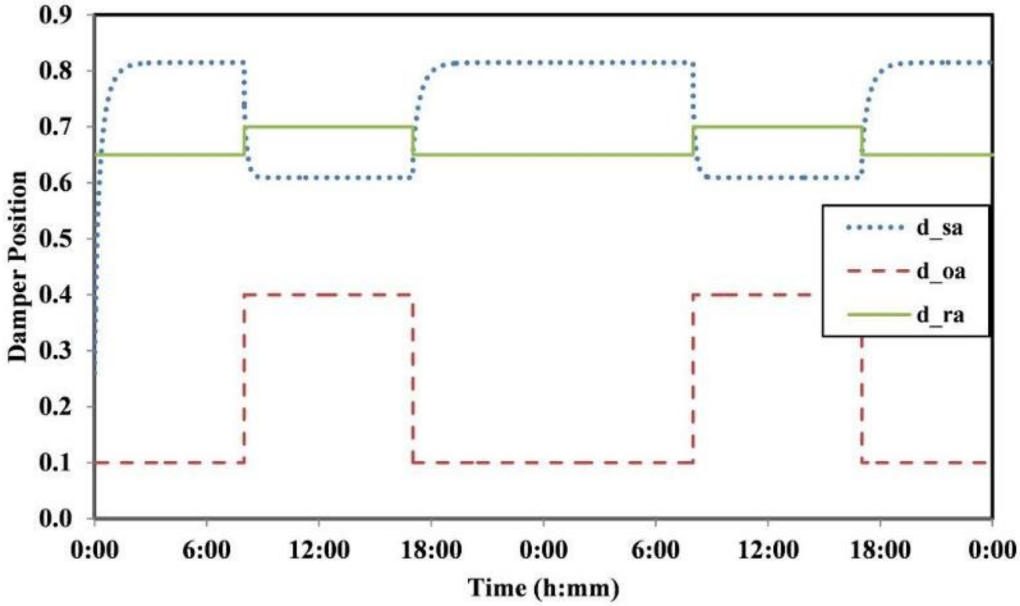


(a)

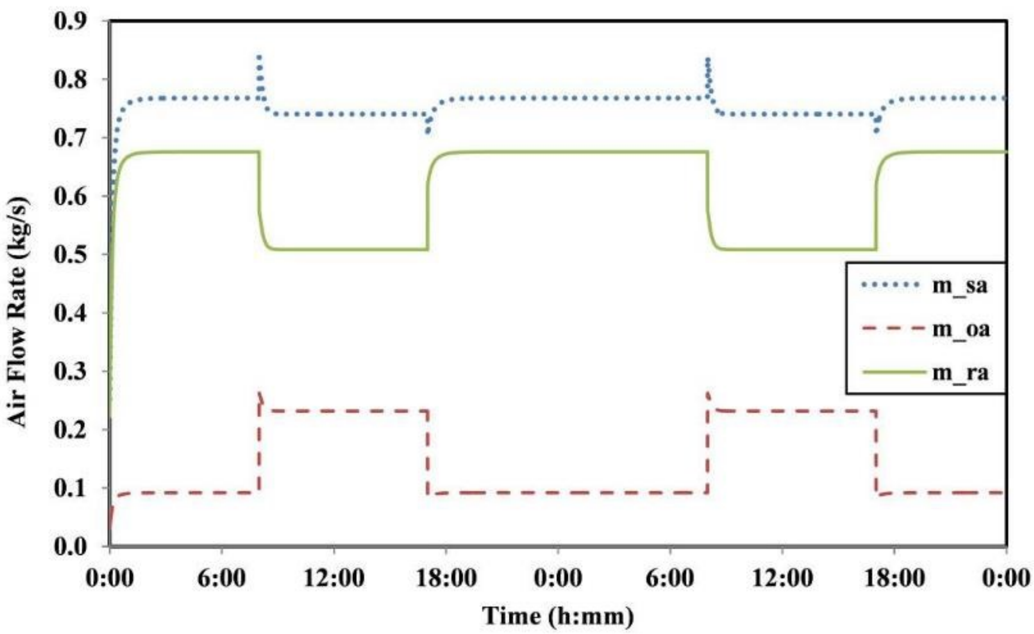


(b)

Figure 5-7 Two-day simulation results of air temperature with schedule based reset (a) summer condition, (b) winter condition.

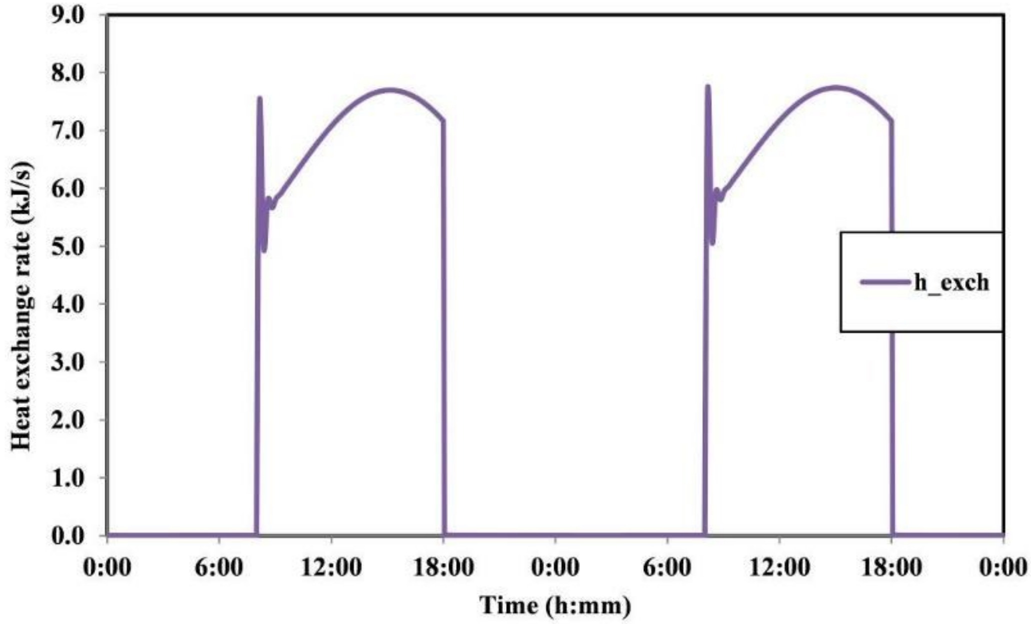


(a)

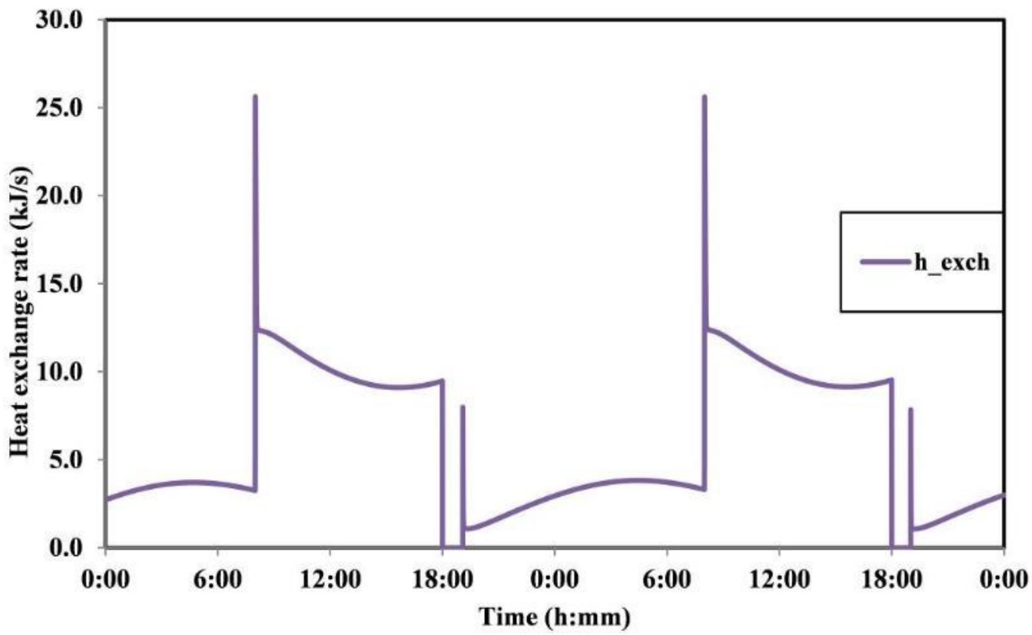


(b)

Figure 5-8 Two-day simulation result of OA/RA/SA dampers positions with schedule based reset (summer and winter conditions) (a) dampers position, (b) airflows rate.



(a)



(b)

Figure 5-9 Two-day simulation results of heat exchange rate at the coil with schedule based reset (a) summer: cooling coil, (b) winter: heating coil.

5.5.7 Limitations and Learning Perspectives

However, through the process of developing the customized library block and simulation cases, the author realized some precautions in the following categories.

First, since the customized library is created and used for local plant programming, there is linkage between the original library file and the local library. The local block linkage needs to be disabled for any modification if needed. After modification, it can be updated back to the customized library and other pre-used files by using the resolve link function. If there are multiple block changes, it is highly recommended to identify which one(s) need to be restored or resolved. Otherwise, this process will likely result in unexpected close of MATLAB.

Second, whether or not to enable zero-crossing detection on state transitions can be a trade-off between accuracy and performance (MathWorks 2013). One might consider disabling the zero-crossing option to speed up the simulation process if the system exhibit frequent fluctuations between two modes of continuous operation (e.g., the determination of whether the cooling coil is wet or dry will result in this type of fluctuation).

5.6 Conclusions

A simulation framework that allows the dynamic simulation of HVAC control system based on Simulink was developed. This platform can be potentially used to support evaluation of different control strategies, analysis of control parameters setting, and development of new control algorithms. In addition, if a simulation model is validated with operation data, then the validated model can be used to compare the simulation data with real-time operation for fault detection and diagnosis.

CHAPTER 6

AN INTEGRATED CASE STUDY

This chapter focuses on demonstrating the control self-configuration framework using a case study project. First, a simulation case is developed using the dynamic HVAC simulation model library. This is a single duct VAV system with multi-zone. Then, two self-configuration scenario cases are demonstrated. The first scenario is the New System Configuration. The *System Profile* of this case system is used to construct the HVAC control, following the self-configuration procedure. The second scenario is the Sensors Failure Reconfiguration. The updated *System Profile* is used in this scenario. Last, the performances of the HVAC system simulation at both scenarios are presented, compared and discussed.

6.1 Building Characteristic and HVAC System Specification

This case study utilizes the developed BCK database from Chapter 3 and the dynamic HVAC system simulation models library developed in Chapter 5 to demonstrate the self-configuration process designed in Chapter 4. Performances of different configuration results are compared in the simulations. The schematic of the HVAC system is given in Figure 6-1.

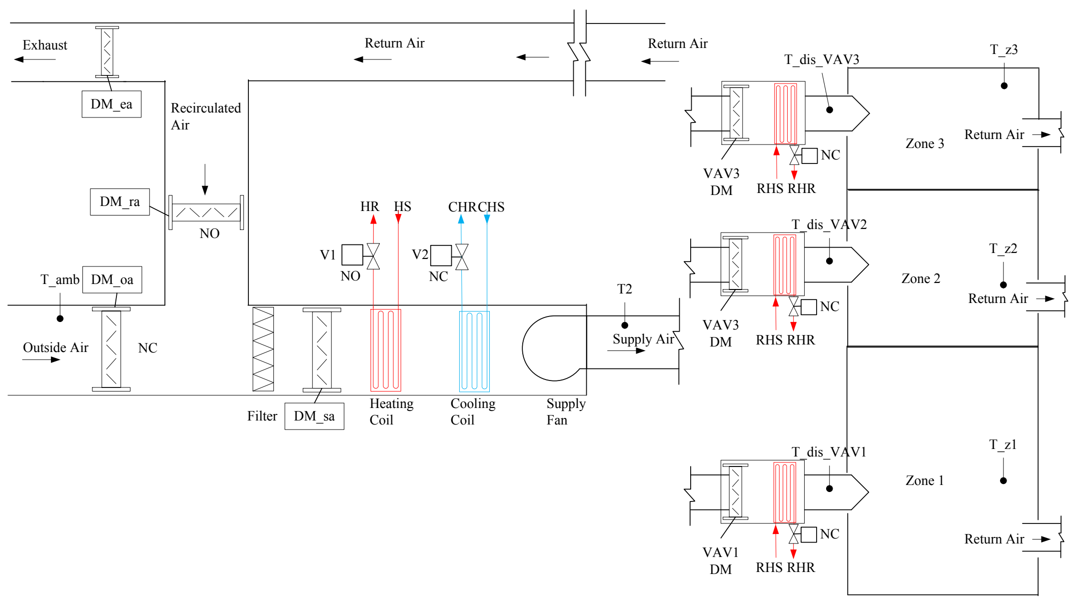


Figure 6-1 Schematic of the HVAC system.

This is a single duct VAV system with three thermal zones. Each zone is conditioned through a VAV box with water based reheat coil. Both the heating and cooling coils at the main air duct are also water based. The details of the building characteristics are given and calculated in Table 6-1. Among those values, the size of the zones, ambient air at design condition (temperature and relative humidity), designed space condition (temperature and relative humidity), wall and roof areas and thermal conductivity, internal heat gains, lowest and highest supply air temperature setpoint were arbitrarily given. These values are highlighted in gray.

Then, the occupancies of each zones were calculated based on the default values given in ASHRAE standard 62.1-2010 (ANSI/ASHRAE 2010). The outside airflow rate requirement for each zone were calculated based on the design condition (occupancy and area using the equation (37) and the

$$V_{bz} = R_p \cdot P_z + R_a \cdot A_z \tag{37}$$

where

A_z = zone floor area: the net occupiable floor area of the ventilation zone (m^2)

P_z = zone population: the number of people in the ventilation zone during typical usage

R_p= outdoor airflow rate required per person

R_a = outdoor airflow rate required per unit area

Then, the cooling load and heating load were calculated using the Psychrometric chart at sea level, by identify the enthalpy values for the design condition, ambient condition, and mixed air condition (see Figure 6-2).

Table 6-1 Building and HVAC system characteristics.

	Details	**Zone 1**	**Zone 2**	**Zone 3**	**Total**
Building Characteristics	Area size: L*W (m)	10*10	6*10	4*5	
	Area (A, m^2)	100	60	20	**180**
	Height (H, m)	2.5	2.5	2.5	
	Volume (Z_volume, m^3)	250	150	50	**450**
	Occupancy (person)	5	3	1	**9**
	Schedule (hh:mm - hh:mm)	9:00-17:00	9:00-17:00	9:00-17:00	
	Office Area Outdoor Air Rate (Ra, L/s.m^2)			0.3[Note 1]	
	Office People Outdoor Air Rate (Rp, L/s. person)			2.5[Note 1]	
	Calculate Occupied Minimum OA Airflow (Vbz, l/s)	42.5	25.5	8.5	
	Occupied Minimum OA Airflow (Kg/s)	0.052	0.031	0.010	**0.094**
	OA rate /SA rate (%)	12%	11%	7%	**11%**
	Exterior Wall 1 U- Value (U_w1, kW/m^2.°C)	0.003	0.003	0.003	
	Exterior Wall 2 U- Value (U_w2, kW/m^2.°C)	0.002	0.002	0.002	
	Exterior roof U- Value (U_r, kW/m^2.°C)	0.002	0.002	0.002	

	Area of Exterior Wall 1 (A_w1, m2)	25	15	10	
	Area of Exterior Wall 2 (A_w2, m^2)	25	25	12.51	
	Area of roof (A_r, m^2)	100	60	20	
	Internal gain (q(t), kW)	1.6	1.05	0.6	**3.25**
Summer Cooling Design	Design Indoor Temperature Setpoint (T_z, ^oC)	21	21	21	
	Design Indoor Relative Humidity (RH, %)	40	40	40	
	Design Condition Ambient Temperature (T_amb, ^oC)	33	33	33	
	Design Condition Ambient Relative Humidity (RH, %)	40	40	40	
	Lowest supply air temperature setpoint (T_sa, ^oC)	13	13	13	
	Calculated Wall temperature (^oC)	27	27	27	
	Calculated cooling load from the zone (kW)	3.55	2.34	1.17	**7.06**
	Calculated airflow (kg/s)	0.444	0.293	0.146	**0.88**
	Calculated airflow (cfm)	767	506	253	**1526**
	Calculated cooling load from outside air (kW)	1.93	1.16	0.39	**3.47**
	Calculated total cooling load (kW)	5.5	3.5	1.6	**10.5**
	Calculated total cooling load using Psychrometric Chart (kW)	5.1	3.4	1.7	**10.1**
	Calculated latent cooling load using Psychrometric Chart (kW)	0.9	0.6	0.3	**1.8**
	Calculated sensible cooling load using Psychrometric Chart (kW)	4.2	2.8	1.4	**8.4**
Winter Heating Design	Design Indoor Temperature Setpoint (T_z, ^oC)	19	19	19	
	Design Condition Ambient Temperature (T_amb, ^oC)	-5	-5	-5	
	Lowest supply air temperature setpoint (T_sa, ^oC)	40	40	40	
	Calculated Wall temperature (T_w, ^oC)	7	7	7	
	Calculated heating load (HX, kW)	3.90	2.58	1.14	**7.62**
	Calculated airflow (kg/s)	0.186	0.123	0.054	**0.36**
	Calculated airflow (cfm)	321	212	94	**627**
HVAC System Specification	Cooling Coil size (heat exchange rate, kJ/s)				11
	Heating coil size (heat exchange rate, kJ/s)				8
	VFD fan operation at design flow rate 1600 cfm, while pressure rise at designed condition is about 0.07 kPa				

Note 1: values obtained from ASHRAE Standard 62.1-2010 Page 13, office spaces.

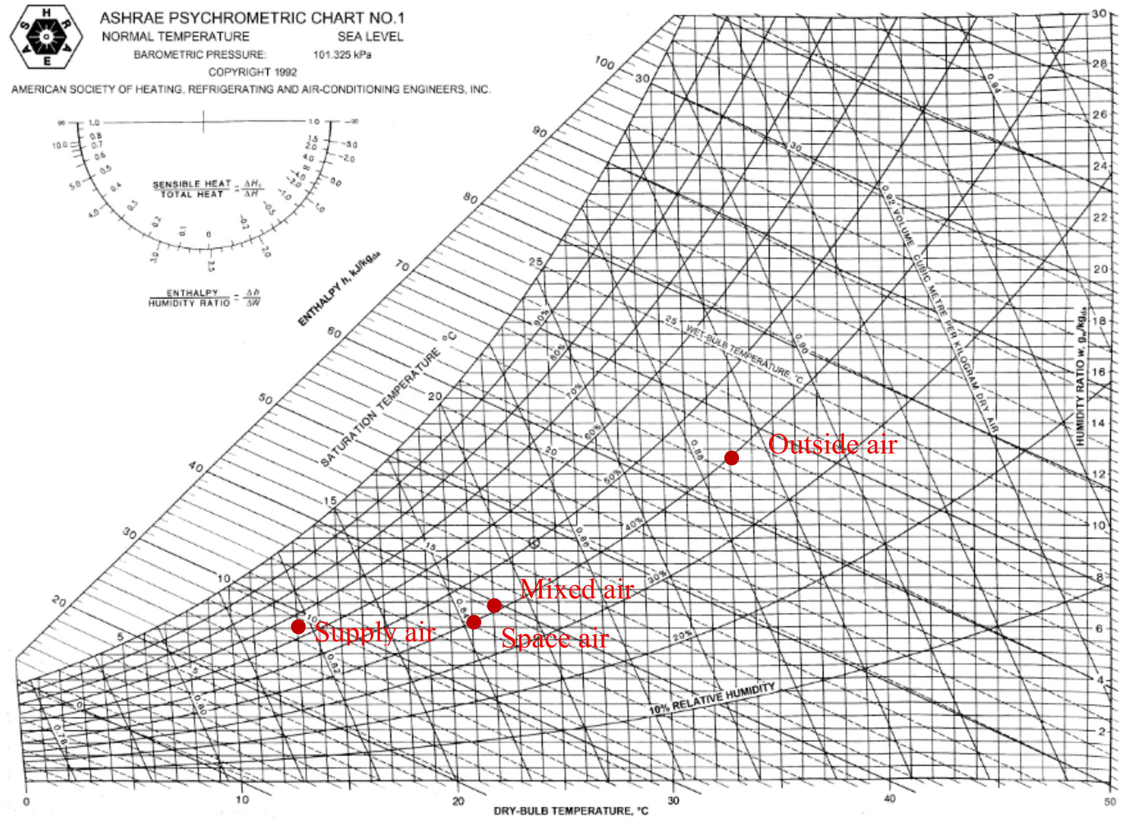


Figure 6-2 Psychrometric chart based cooling load calculation (Psychrometric chart obtained from ASHRAE Handbook of Fundamentals (ASHRAE 2013)).

6.2 Construction of the Dynamic Simulation Platform

Based on the building and HVAC system characteristics given above, a dynamic simulation platform was developed using the MATLAB/Simulink HVAC system models created in Chapter 5. Parameters of each Simulink HVAC components model are reconfigured to match this case study.

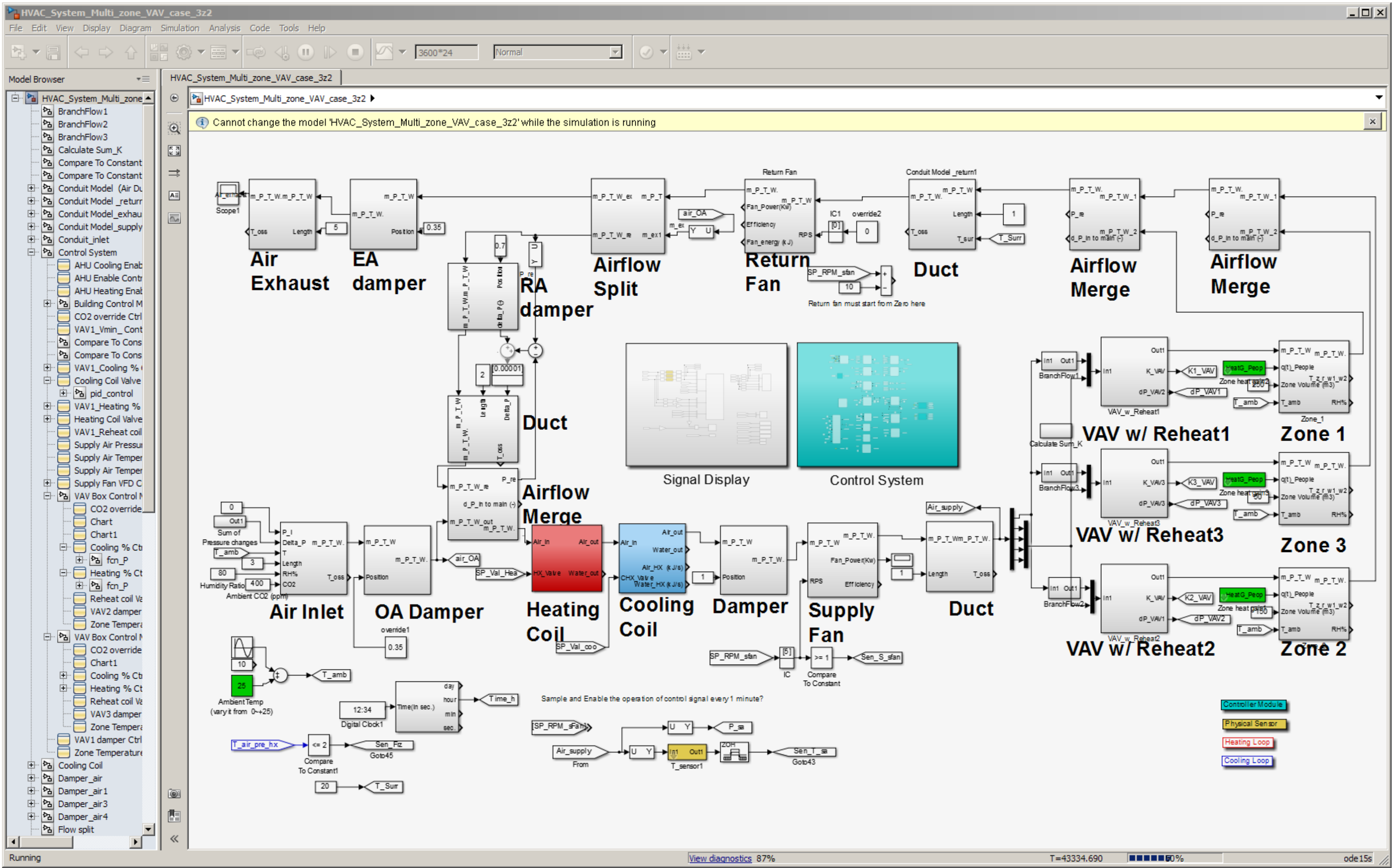


Figure 6-3 Programming of HVAC system – MATLAB/Simulink simulation platform screenshot.

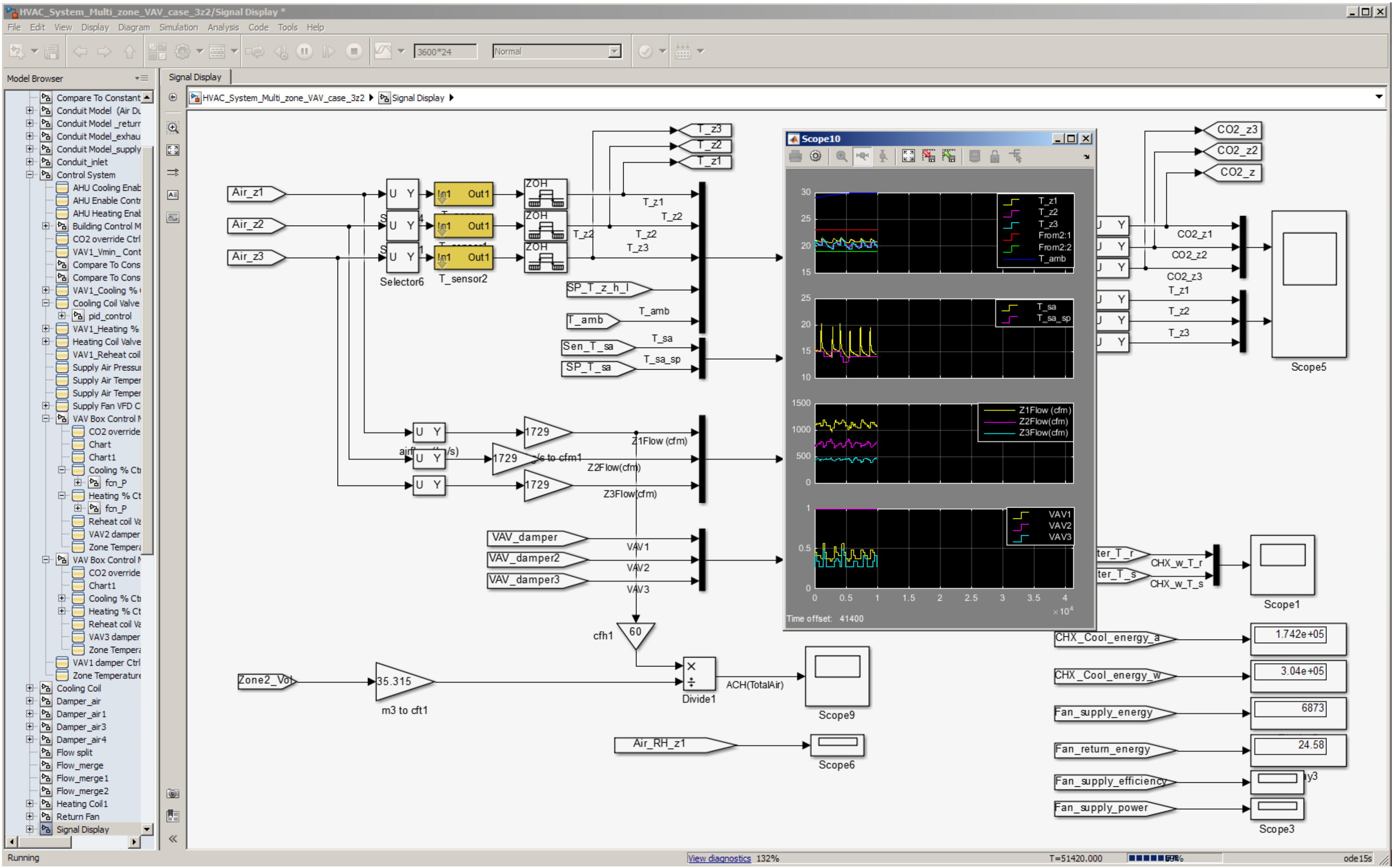


Figure 6-4 Programming of signal display panels – MATLAB/Simulink simulation platform screenshot.

6.3 Demonstration of the Control Self-Configuration

This section will demonstrate the self-configuration process using two self-configuration scenario cases: new system configuration and sensor failure reconfiguration. Each scenario is demonstrated using the prerequisite, the illustration of the self-configuration process, and the simulation result of the HVAC system performance using the self-configured control modules.

6.3.1 Scenario case I

The scenario for case I was set to configure a new building control system. This implementation process followed the procedure described in section 4.2.5.2.

6.3.1.1 Prerequisite

The prerequisite for the self-configuration process is listed in the *System Profile* (Table 6-2) for HVAC system being configured.

Table 6-2 The *System Profile* for Scenario Case I.

Items	Details
1. HVAC system type	Single duct VAV system
2. Sensors (data points) list	Temperature – ambient air
	Temperature – supply air (main)
	Temperature – Zone 1 air
	Temperature – Zone 2 air
	Temperature – Zone 3 air
	CO_2 concentration – Zone 1 air
	CO_2 concentration – Zone 2 air
	CO_2 concentration – Zone 3 air
	Pressure – supply air (main)

	Status of freezing alarm
	Status of supply fan operation
3. Local control variables list	AHU heating coil valve
	AHU cooling coil valve
	Outside air/Recirculated air dampers
	Supply fan variable frequency drive (VFD)
	Zone 1 VAV box damper
	Zone 1 VAV box reheat coil valve
	Zone 2 VAV box damper
	Zone 2 VAV box reheat coil valve
	Zone 3 VAV box damper
	Zone 3 VAV box reheat coil valve
4. Specific parameters list and their values	Zone 1 minimum airflow when occupied
	Zone 2 minimum airflow when occupied
	Zone 3 minimum airflow when occupied
	Zone 1 maximum design airflow
	Zone 2 maximum design airflow
	Zone 3 maximum design airflow

6.3.1.2 BCK Modules Configuration

Due to the lack of the self-configuration programming code, the BCK modules are configured following the self-configuration procedure. The major step-by-step results are illustrated in Figure 6-5. The *System Profile* was firstly used to filter the BCK database to identify the relevant control modules in original database. Then, the local control variables were used to identify the local control modules. Based on the local control modules' input variable list, a number of system control variables were identified and used to identify the system control modules. After that, the configured control modules lists were checked and then exported. The exported control modules (only the programming code) were then tested in the dynamic simulation platform constructed in Section 6.2.

The screenshot of the configured each control module (only the programming code) is shown in Figure 6-6.

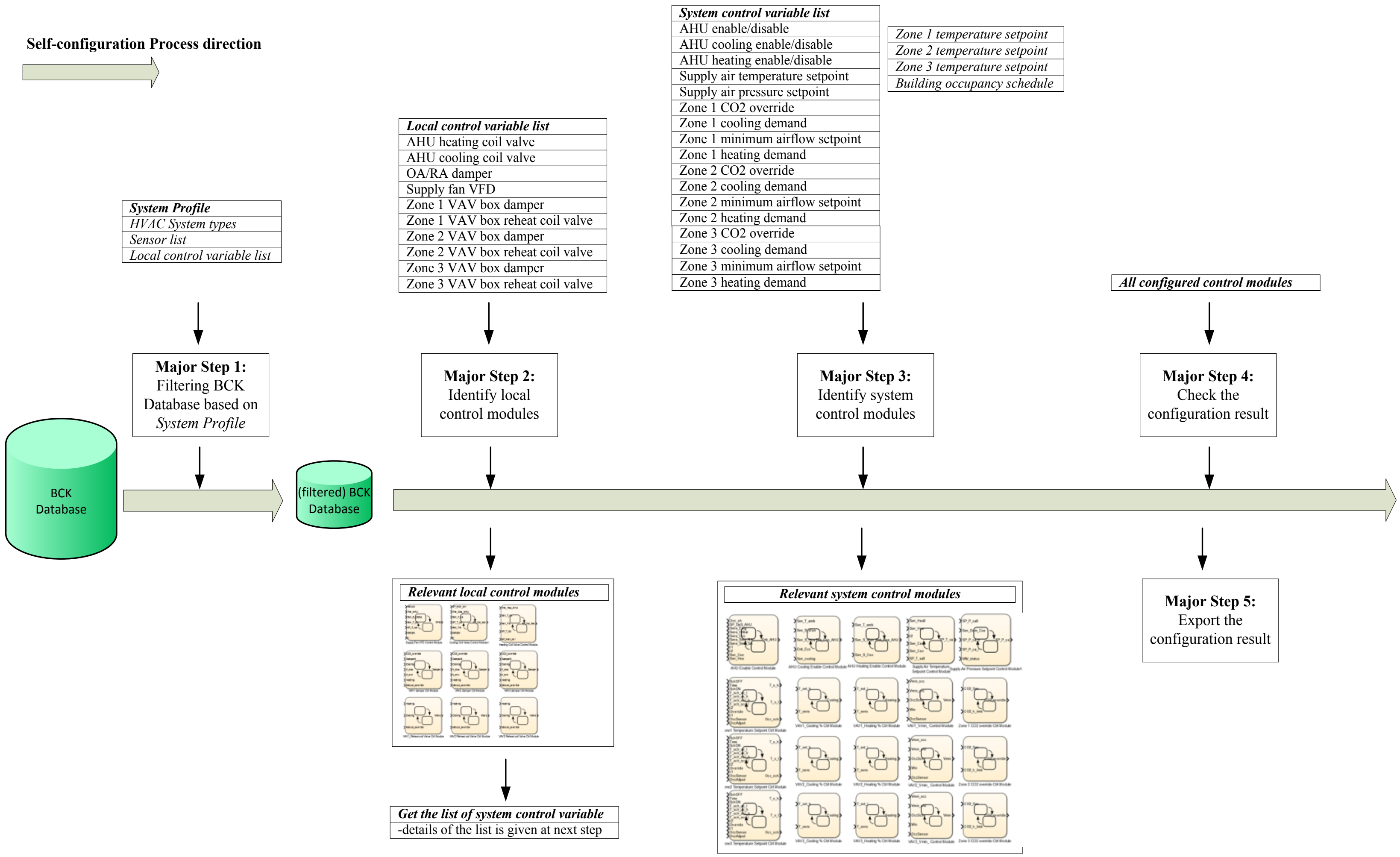


Figure 6-5 Self-configuration results from the major steps (Scenario case I).

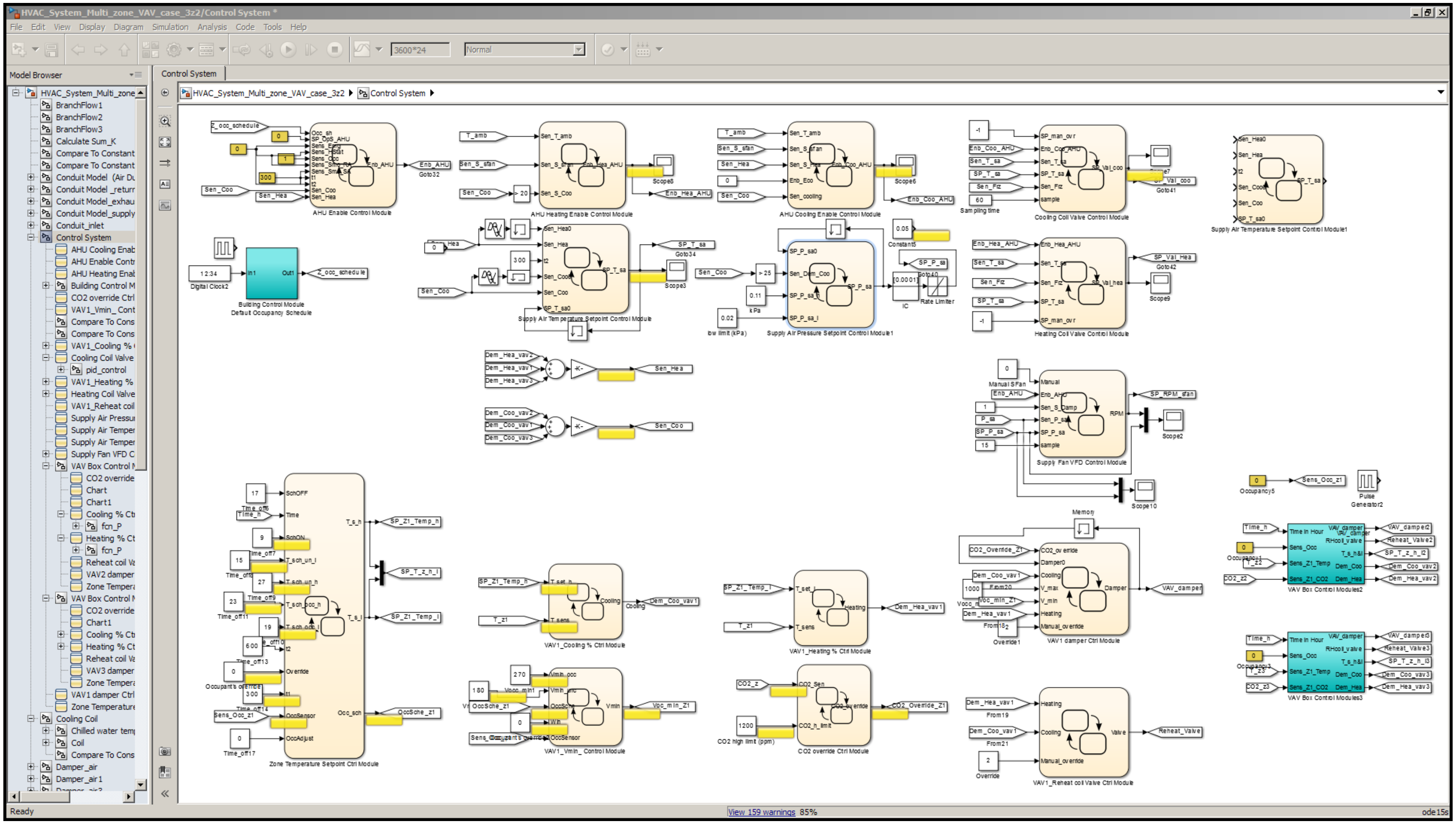
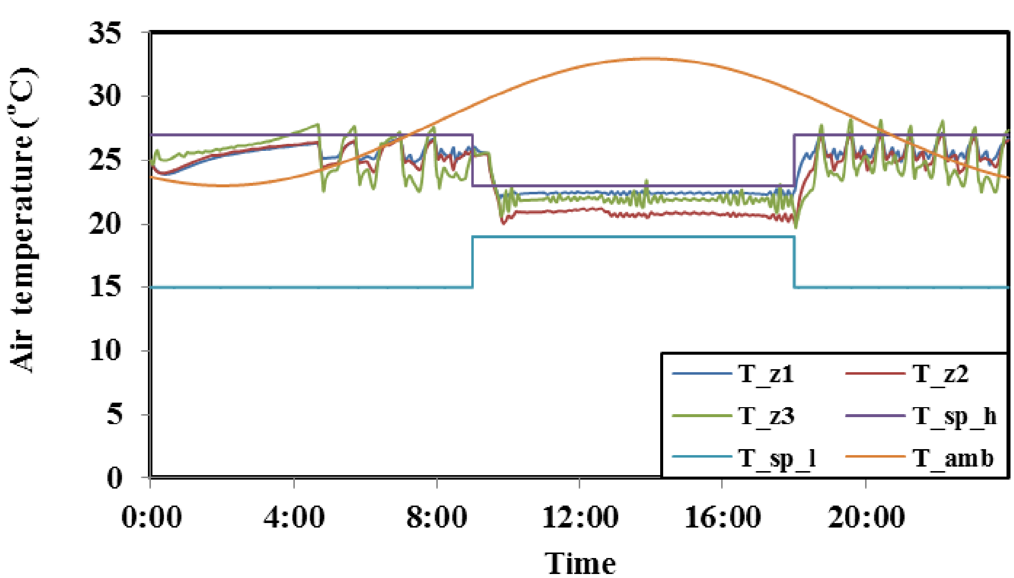


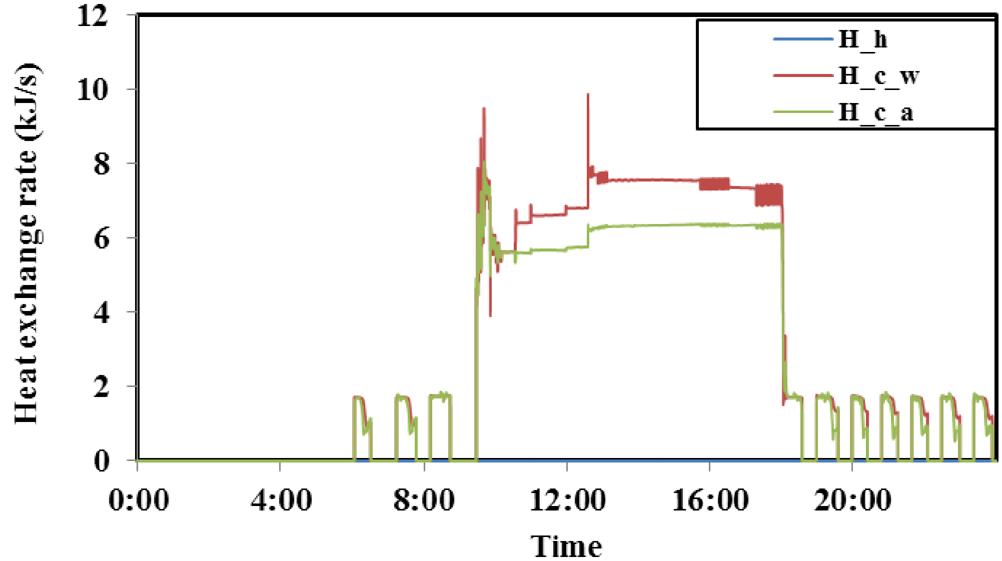
Figure 6-6 Configuration result of the BCK modules (screenshot of control programming codes).

6.3.1.3 Simulation result of the control performance

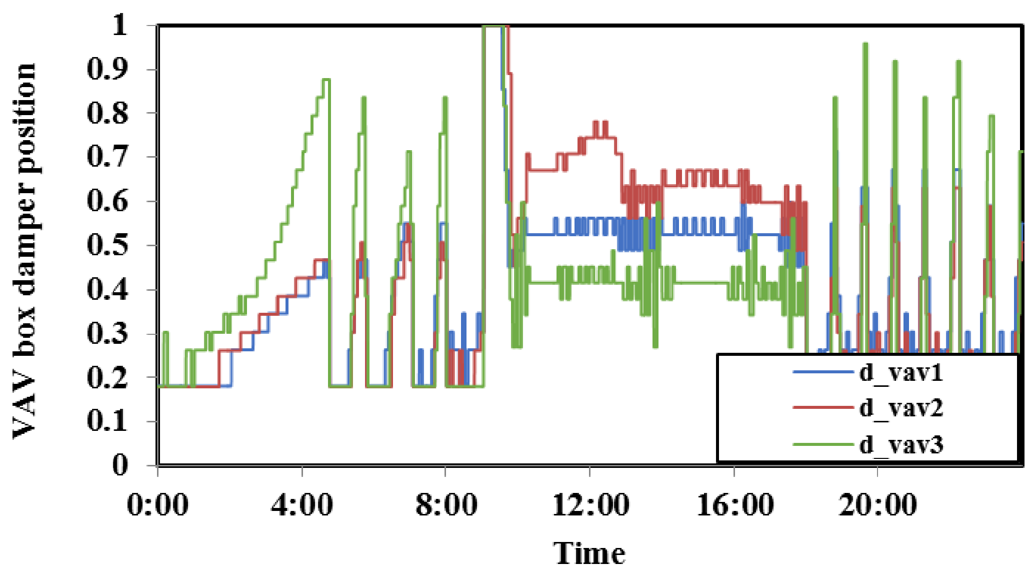
The control performance of the newly configured control is presented in Figure 6-7 (a) to (e). The simulation result showed that the indoor air temperatures were well maintained within setpoint during the whole period. The CO_2 concentrations were maintained under 1200 ppm. There were some fluctuations of the temperature at the unoccupied period. This was caused by the ON OFF of fan operation, which caused the VAVs to modulate the damper positions. When the fan just started, its rotational speed quickly went up to maximum and then came down. This problem might be solved by revising the control programming code to disable the modification of VAVs during the unoccupied period. In general, the simulation result indicated that the configured control modules fulfill the functional requirement of air conditioning in this HVAC system.



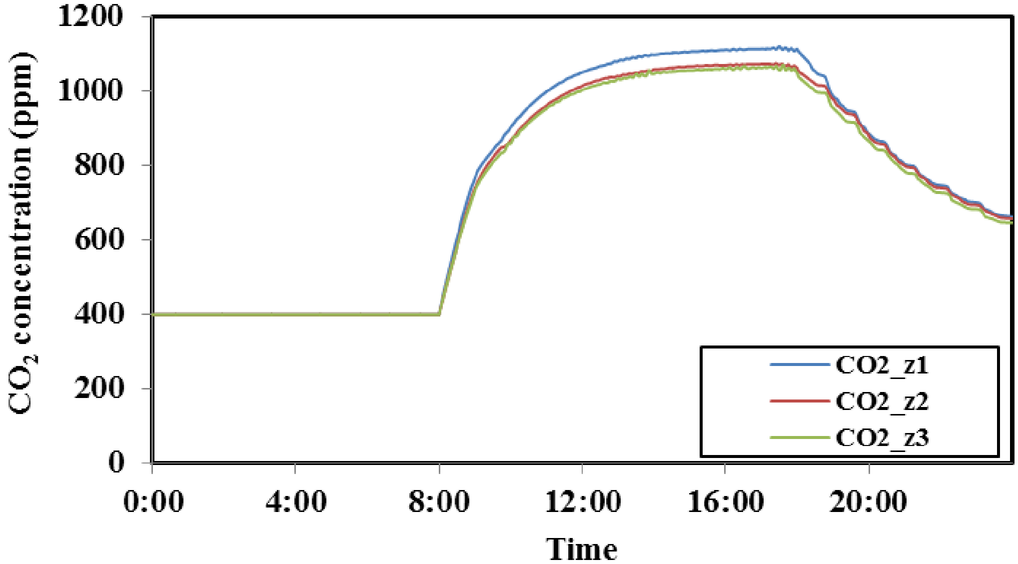
(a)


H_h
H_c_w
H_c_a
Heat exchange rate (kJ/s)
Time
0
2
4
6
8
10
12
0:00
4:00
8:00
12:00
16:00
20:00

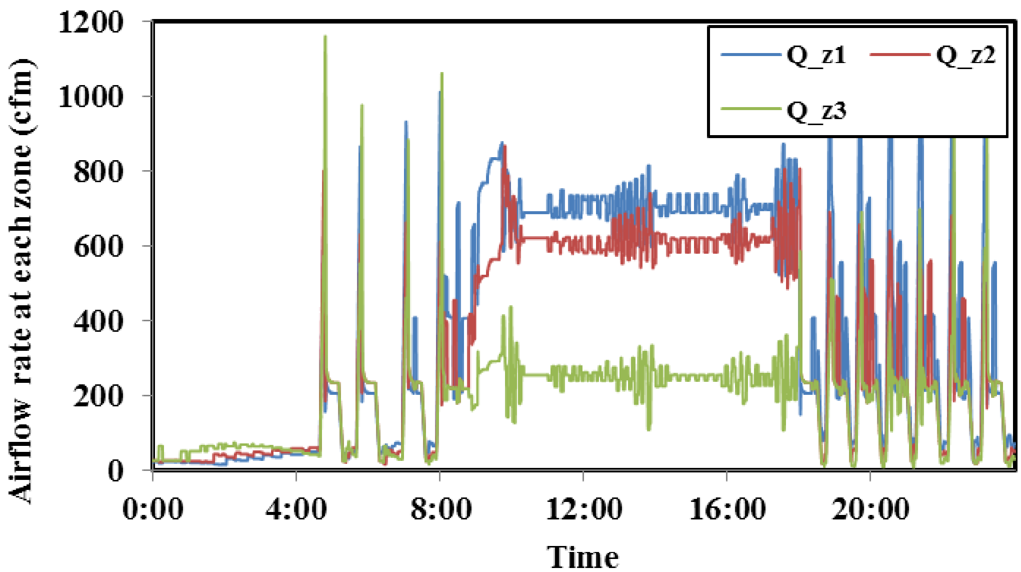
(b)


d_vav1
d_vav2
d_vav3
VAV box damper position
Time
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0:00
4:00
8:00
12:00
16:00
20:00

(c)


1200
1000
800
600
400
200
0
CO2 concentration (ppm)
0:00
4:00
8:00
12:00
16:00
20:00
Time
CO2_z1
CO2_z2
CO2_z3

(d)


1200
1000
800
600
400
200
0
Airflow rate at each zone (cfm)
0:00
4:00
8:00
12:00
16:00
20:00
Time
Q_z1
Q_z2
Q_z3

(e)

Figure 6-7 Scenario case I HVAC system operation at the cooling design day (a) air temperatures, (b) heat exchange rates, (c) VAV boxes damper and reheat coil positions, (d) CO_2 concentration at each zone, (e) airflow rate at each zone.

6.3.2 Scenario case II

The scenario case II simulated the situation when a sensor/equipment failure occurs. In this case, the situation was set as: during the operation, a FDD tool detected the CO_2 sensor in Zone 3 was broken. Based on this result, the *System Profile* was updated by removing the CO_2 sensor in Zone 3 from the Sensors list. Then, with the updated *System Profile*, the self-configuration program reconfigured the control system and replaced the control modules that were affected by the broken sensor removal.

6.3.2.1 Prerequisites

Table 6-3 The *System Profile* for Scenario Case II.

Items	Details
1. HVAC system type	Single duct VAV system
2. Sensors (data points) list	Temperature – ambient air
	Temperature – supply air (main)
	Temperature – Zone 1 air
	Temperature – Zone 2 air
	Temperature – Zone 3 air
	CO_2 concentration – Zone 1 air
	CO_2 concentration – Zone 2 air
	~~CO_2 concentration – Zone 3 air~~
	Pressure – supply air (main)
	Status of freezing alarm
	Status of supply fan operation
3. Local control variables list	AHU heating coil valve
	AHU cooling coil valve

	Outside air/Recirculated air dampers Supply fan variable frequency drive (VFD) Zone 1 VAV box damper Zone 1 VAV box reheat coil valve Zone 2 VAV box damper Zone 2 VAV box reheat coil valve Zone 3 VAV box damper Zone 3 VAV box reheat coil valve
4. Specific parameters list and their values	Zone 1 minimum airflow when occupied Zone 2 minimum airflow when occupied Zone 3 minimum airflow when occupied Zone 1 maximum design airflow Zone 2 maximum design airflow Zone 3 maximum design airflow

6.3.2.2 BCK Modules Reconfiguration

The reconfiguration process used the updated *System Profile* and reconfigured the relevant control modules (shown in Figure 6-8). Due to the lack of CO_2 sensor for Zone 3, the VAV damper control module with CO_2 override were excluded at the first step. Then, during the identification process for the local control modules, only the VAV damper control module without CO_2 override was selected. Then, the system control module - CO_2 override setpoint was only selected for Zone 1, Zone 2, but not for Zone 3. In this way, the reconfigured control modules shall satisfy the updated *System Profile*.

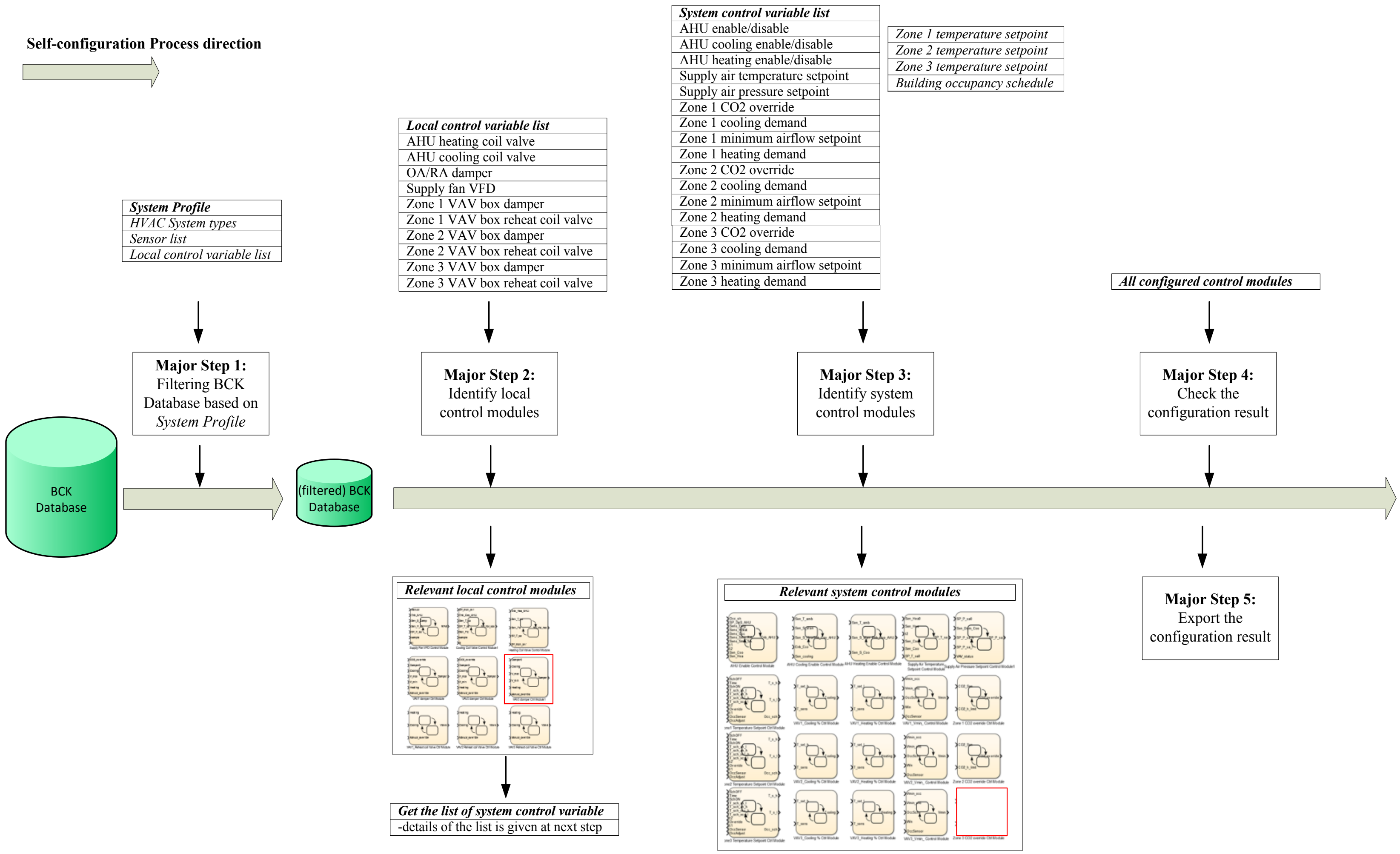
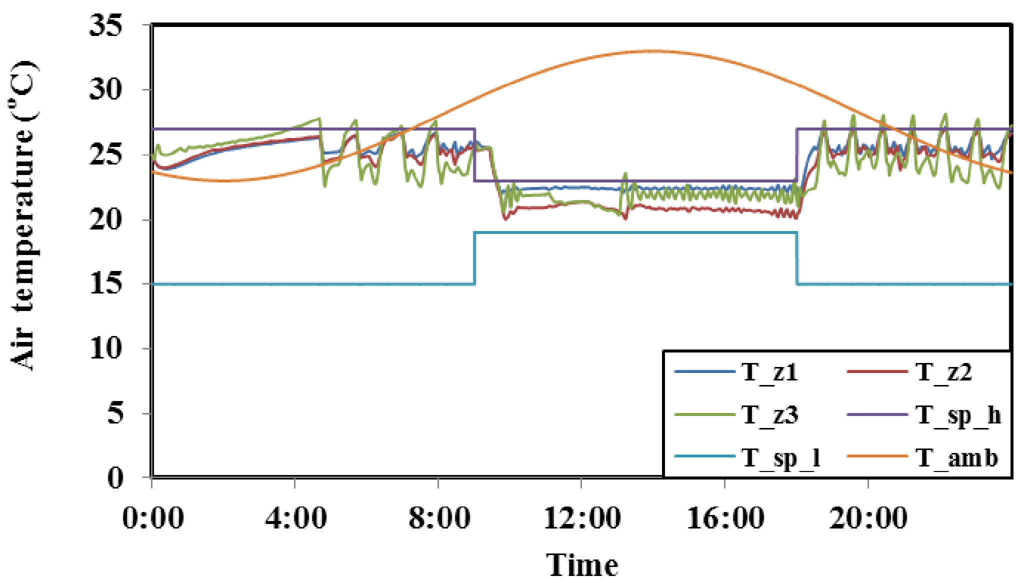


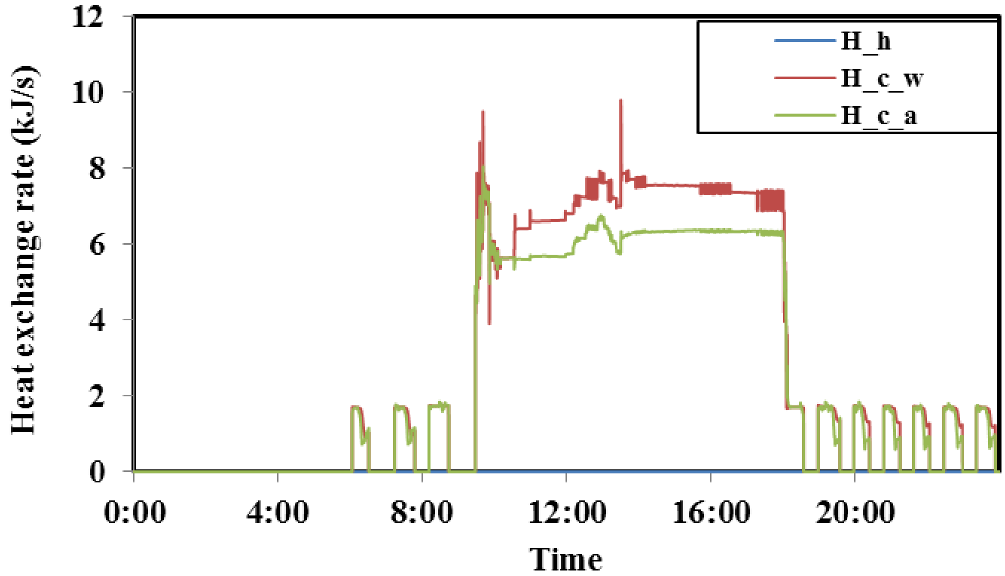
Figure 6-8 Self-configuration results from the major steps (Scenario case II).

6.3.2.3 Simulation result of the control performance

The control performance of the reconfigured control modules was simulated as: the failure of CO_2 sensor for Zone 3 started from 11:00, and the failure situation was sensor drift with white noise; then, at 13:00, the sensor failure was diagnosed, and the self-configuration program updated the whole control modules. The simulated HVAC system performances are shown in Figure 6-9 (a) to (e). In Figure 6-9 (a), the temperature of each zone was generally well maintained within the setpoint band. In Figure 6-9 (b), it can be seen that the cooling coil heat exchange rate increased slightly right after the sensor broken (if comparing with the previous case Figure 6-7 (b)). This is due to the fully open of VAV3 damper caused by the false CO_2 signal (see Figure 6-9 (c), and the damper returned to normal after reconfiguration process happened at 13:00. With self-configuration process, the system was able to switch to another set of control algorithms in a timely manner, without losing any functional requirement during this process.

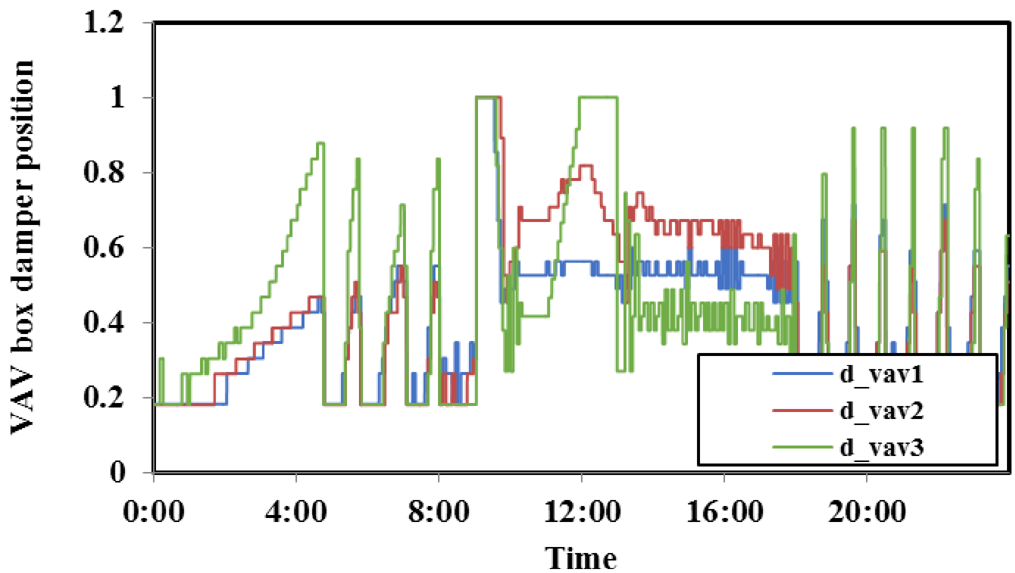


(a)



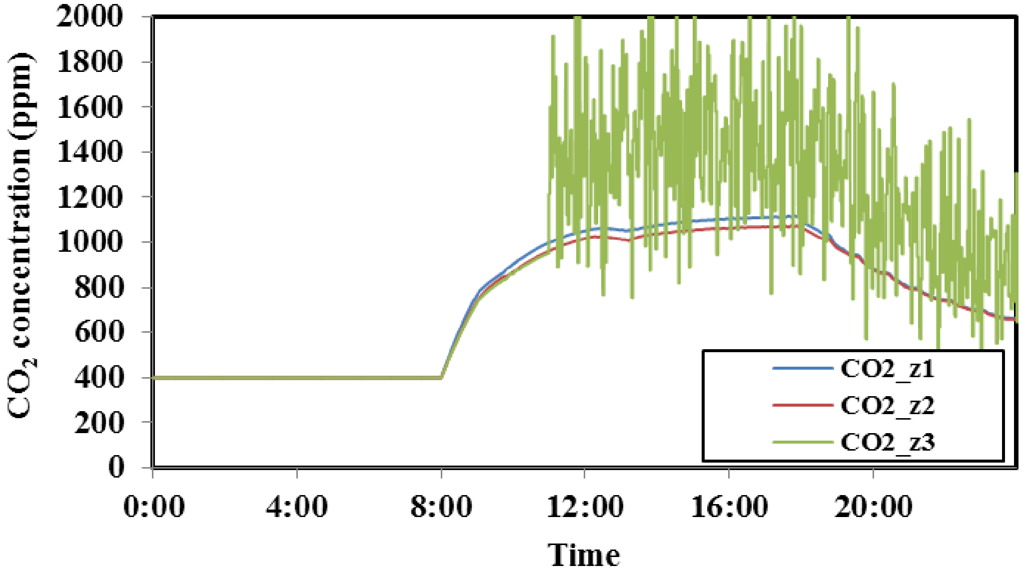
12
10
8
6
4
2
0
Heat exchange rate (kJ/s)
H_h
H_c_w
H_c_a
0:00
4:00
8:00
12:00
16:00
20:00
Time

(b)

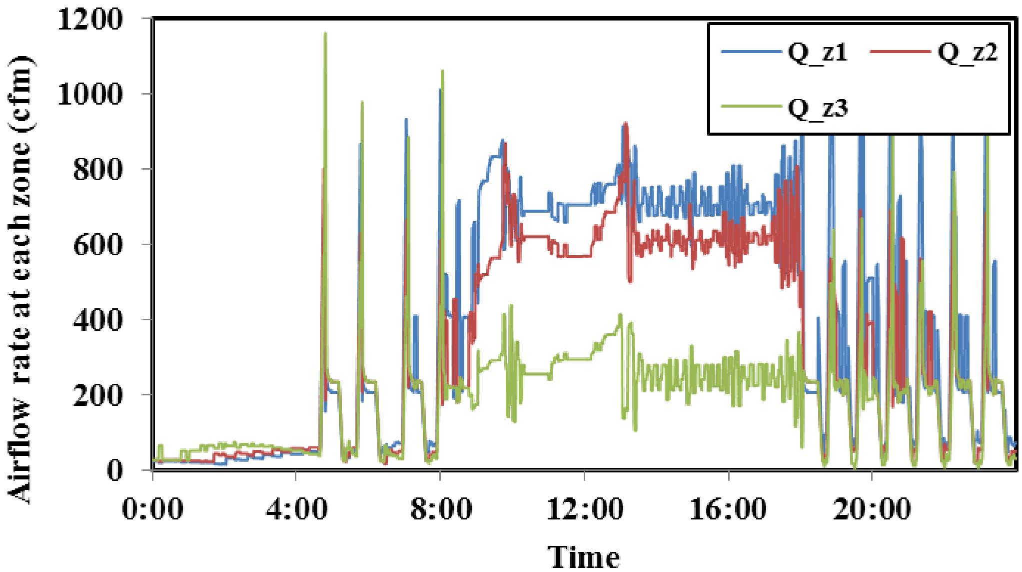


1.2
1
0.8
0.6
0.4
0.2
0
VAV box damper position
d_vav1
d_vav2
d_vav3
0:00
4:00
8:00
12:00
16:00
20:00
Time

(c)



(d)



(e)

Figure 6-9 Scenario case II HVAC system operation at the cooling design day (a) air temperatures, (b) heat exchange rates, (c) VAV boxes damper and reheat coil positions, (d) CO_2 concentration at each zone, (e) airflow rate at each zone.

6.4 Summary and Discussion

In summary, this chapter presented a case study that utilized the control self-configuration process to generate the control modules for two different scenarios. The configured control system was then tested through the dynamic simulation platform. The simulation results indicate the control performances met the functional requirement. It also shows that the immediate response of the self-configuration during the sensor failure case might provide solutions to better utilizing FDD tools and generate alternative control algorithms for the systems with failure sensors or equipment. In this way, the HVAC system would be able to keep maintaining the functional requirement with the best control configuration at the given *System Profile*. Meanwhile, the other self-configuration scenarios (retrofit and BCK database updates) were not demonstrated here. However, it can be seen that this would enable the deployment of new control algorithms through this BCK database.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

This is not the end. It is not even the beginning of the end. But, it is perhaps the end of the beginning.

— Winston Churchill

This chapter concludes this research by summarizing the findings and limitations. Particularly, conclusions from this study are discussed from the BCK representation framework development, BCK management, building control self-configuration, and dynamic simulation of HVAC systems for control analysis. Then, a list of future directions is given in the following context.

7.1 Conclusion

This study has investigated the research topics of BCK information modeling and the self-configuration of building control. Through this study, the research objective of developing BCK representation and management framework and then utilizing it to self-configure building control has been achieved. Specifically, this is realized through four parts of interrelated work. First, it developed a universal/standard data schema for the BCK based on analysis of existing representation formats (in Chapter 2). Second, based on the BCK data schema, a prototype BCK database was created (in Chapter 3). Then, a self-configuration framework was developed based on interacting with the BCK database (in Chapter 4). Last, a dynamic HVAC simulation platform was developed in MATLAB/Simulink to test the control configuration result (in Chapter 5). The integrated case study in Chapter 6 provided the demonstration of the self-configuration process.

This study contributes to information modeling for building control system (especially focus on the representation of BCK), the dynamic HVAC system modeling for control analysis, and the control self-configuration to support system reconfiguration and adaptation. The methods developed in this work are innovative and explorative. It is the first time that "building control knowledge" is defined and used as a new concept, by incorporating the critical elements to represent the BCK. The data schema for the BCK developed in this study provides a potential solution to standardize the BCK representation to eliminate the ambiguity issues in control knowledge documentation.

The developed BCK database can potentially be used as a platform for managing BCK, allowing users to obtain relevant BCK modules and contribute to new modules development. This could potentially help the deployment of advanced control algorithms as new control modules.

Different from the existing self-configuration process (with pre-installed fixed control algorithms), self-configuration of control can be achieved through integrating and interacting with BCK database. BCK database is serving as the source center for the control knowledge management. This self-configuration process would be more generic and potentially have a wider applicable range. The challenge parts would be the maintenance of BCK knowledge database, and method development for selecting and configuring appropriate control modules for specific case.

The MATLAB/Simulink based simulation platform can provide solutions for complicated calculation of dynamic airflow rates, air quality, and other environmental parameters for control configuration analysis. MATLAB/Simulink is typically used for advanced control algorithm development. With the dynamic simulation platform also developed in MATLAB/Simulnk, it is convenient for developers to diagnose any programming errors in the same simulation environment without the requirement of using co-simulation and debugging in different simulation environments at the same time.

7.2 Future Directions

Based on the findings and limitations of this work, a number of research directions are recommended for further work:

- **Development of BCK database**

  The database content can be further enhanced with the disclose of project reports, e.g., ASHRAE *RP 1455: Best of Class Control Sequences for Air Systems* (Hydeman and Eubanks 2015). Because of the extensibility of the database structure, additional classes can be added to in the future. For example, hardware (equipment/system) information class can be added in the next step. In this class, each equipment/system will be an object, including basic information about the ID, manufacture, model #, S/N #, functionality, etc. Meanwhile, the future developer may analyze the IFC framework and incorporate the control knowledge representation as part of the model view structure.

- **Integration of the dynamic simulation platform for evaluating the self-configuration result**

  Using the dynamic simulation platform to assess the performance of self-configured building control can be developed based on control performance evaluation criteria (e.g., using control quality factor that are current being developed in *ASHRAE RP 1587 Control Loop Performance Assessment*). In this case, after testing and evaluation of different control module configurations, the one with the best control performance may be identified. In this way, the dynamic simulation platform can be incorporated and used as part of the self-configuration process. This would also need to change the self-configuration process to allow different versions of control system to be configured.

- **Development of programming code for the self-configuration process**

  Due to the time limitations, this work was not able to develop the programming code for the self-configuration process. This should be developed in the future to realize the whole process through programming code. The anticipated programming language for developing this process is Python.

- **Test of the self-configuration process with more FDD cases**

The self-configuration process, especially the reconfiguration scenario for the sensor and equipment failure case, might be potentially beneficial to the FDD process. It may be developed as a supplementary process that can support the FDD. As demonstrated in the integrated case study, if the self-configuration process reconfigured the control with an alternative version, this might provide the immediate response to maintain the appropriate operation of a HVAC system during an abnormal situation. This demands more tests and evaluation with the conventional FDD tools and algorithms.

- **Development of standard classification of HVAC systems**

  HVAC system type is one of the criteria used in the primary querying process of the BCK database. Currently, there are different ways of classifying HVAC systems, but no standard method. A standard classification shall be developed based on surveying of the engineers and the analysis of the systems and components types.

- **Validation of the HVAC simulation models**

  The MATLAB/Simulink dynamic HVAC system models are developed based on the mathematic equations mostly from HVACSIM+ and other peer-reviewed published journal papers. Although part of these models have been validated and tested in field or lab experiment, those models have not been tested and validated as a whole altogether. Therefore, there is a need to validate the complete simulation platform using field / experimental data.

- **Analysis of artificial intelligence based advanced control algorithms**

  The idea of self-configuration is to make the configuration process more efficient and effective. However, it meanwhile puts the system more autonomous and smarter, especially with the implementation of artificial intelligent algorithms. The self-configuration itself might potentially relocate some job positions to the management of the BCK database, but the direction of advanced control algorithms development and deployment in the HVAC system shall be discussed to avoid unwanted smartness in the systems operation.

REFERENCES

Afram, A. and F. Janabi-Sharifi (2015). "Gray-box modeling and validation of residential HVAC system for control system design." Applied Energy 137(0): 134-150.

Akinci, B., J. H. Garrett, et al. (2011). Identification of Functional Requirements and Possible Approaches for Self-Configuring Intelligent Building Systems. Final Report. Gaithersburg, MD, National Institute for Standards and Technology. NIST Grant 60NANB8D8140.

Angelov, P. P. and R. A. Buswell (2003). "Automatic generation of fuzzy rule-based models from data by genetic algorithms." Information Sciences 150(1–2): 17-31.

ANSI/ASHRAE (2010). ASHRAE Standard 62.1-2010 Ventilation for Acceptable Indoor Air Quality. Atlanta, GA, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.

ASHRAE (2005). ANSI/ASHRAE Standard 134-2005 Graphic Symbols for Heating, Ventilating, Air-Conditioning, and Refrigerating Systems. Atlanta, GA 30329, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.: 79.

ASHRAE (2005). Sequences of Operation for Common HVAC Systems. Multimedia CD Atlanta, GA, American Society of Heating Refrigerating and Air-Conditioning Engineers

ASHRAE (2007). ASHRAE Guideline 13-2007: Specifying Direct Digital Control Systems, American Society of heating Refrigerating and Air-Conditioning Engineers.

ASHRAE (2009). 2009 ASHRAE Handbook: Fundamentals : Si Edition, American Society of Heating, Refrigerating & Air Conditioning Engineers, Incorporated.

ASHRAE (2010). BACnet - A Data Communication Protocol for Building Automation and Control Networks. Atlanta, GA 30329, American Society of Heating, Refrigerating and Air-Conditioning Engineers. ANSI/ASHRAE Standard 135-2010.

ASHRAE (2011). 2011 ASHRAE Handbook - Heating, Ventilating, and Air-Conditioning Applications (I-P Edition), American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.

ASHRAE (2013). 2013 ASHRAE Handbook: Fundamentals (IP & SI), American Society of Heating, Refrigerating and Air Conditioning Engineers.

AutomatedLogic. (2012). "CtrlSpecBuilder." Retrieved Dec. 31, 2012, from https://www.ctrlspecbuilder.com.

AutomatedLogic (2014). EquipmentBuilder for Educator 5.5. http://www.automatedlogic.com/eikon-for-educators/%20/, AutomatedLogic. 5.5.

Avci, M., M. Erkoc, et al. (2013). "Model predictive HVAC load control in buildings using real-time electricity pricing." Energy and Buildings 60(0): 199-209.

Bai, J., S. Wang, et al. (2008). "Development of an adaptive Smith predictor-based self-tuning PI controller for an HVAC system in a test room." Energy and Buildings 40(12): 2244-2252.

Baldwin, J. M., R. A. Bishop, et al. (1994). Self configuring controller. U.S. Patent Office. United States, American Standard Inc. 5,276,630.

Bi, Q., W.-J. Cai, et al. (2000). "Advanced controller auto-tuning and its application in HVAC systems." Control Engineering Practice 8(6): 633-644.

Bobál, V., J. Böhm, et al. (2005). Digital Self-tuning Controllers: Algorithms, Implementation and Applications. London, UK, Springer Science & Business Media.

Bourdouxhe, J. P., M. Grodent, et al. (1998). Reference Guide for Dynamic Models of HVAC Equipment, American Society of Heating, Refrigerating & Air Conditioning Engineers, Incorporated.

Brambley, M. R., P. Haves, et al. (2005). Advanced sensors and controls for building applications: Market assessment and potential R & D pathways. Technical Report Richland, WA, Pacific Northwest National Laboratory.

Braun, J. E. and N. Chaturvedi (2002). "An inverse gray-box model for transient building load prediction." HVAC&R Research, 8(1): 73-99.

buildingSMART International. (2013). "Building Automation Modeling." Model View Definition Industry Foundation Classes (IFC 4) Retrieved Oct. 6, 2014, from http://docs.buildingsmartalliance.org/MVD_BAMIE/link/building-service-element.htm.

Calabrese, S. (2010). Sequences and Control Schemes for Air and Water Systems, ASHRAE Illinois Chapter.

Cao, X. H., J. M. Chen, et al. (2010). "Building-Environment Control With Wireless Sensor and Actuator Networks: Centralized Versus Distributed." IEEE Transactions on Industrial Electronics, 57(11): 3596-3605.

Chen, P. (1976). "The Entity-Relationship Model: Toward a Unified View of Data." ACM Transactions on Database Systems 1: 9-36.

Chen, Y. and S. Treado (2014). Self-Configuration of Building Control System Using Knowledgebase. American Society of Heating, Refrigerating and Air Conditioning (ASHRAE) Annual Conference, Seattle, WA

Chesley, G. D. (1977). Self-configurable circuit structure for achieving wafer scale integration. U.S. Patent Office. 4,038,648.

Chesley, G. D. (1980). Self-configurable computer and memory system. U.S. Patent Office. 4,191,996.

Clark, D. R. (1985). HVACSIM$^+$ Building Systems and Equipment Simulation Program Reference Manual. National Bureau of Standards. Gaithersburg, MD 20899, U.S. Department of Commerce,.

Congradac, V. and F. Kulic (2009). "HVAC system optimization with CO2 concentration control using genetic algorithms." Energy and Buildings 41(5): 571-577.

COOLTOOLS™ (2009). Design Guidelines: Chilled Water Plant Design Guide.

Dempsey, M. (2003). Automatic translation of Simulink models into Modelica using Simelica and the AdvancedBlocks library. Proceedings of the 3rd International Modelica Conference,, Linköping,Sweden.

Dong, B. (2010). Integrated Building Heating, Cooling and Ventilation Control. Doctor of Philosophy Dissertations, Carnegie Mellon University.

Dong, B., Z. O'Neill, et al. (2014). "A BIM-enabled information infrastructure for building energy Fault Detection and Diagnostics." Automation in Construction 44: 197-211.

Dounis, A. I., P. Tiropanis, et al. (2011). "Intelligent control system for reconciliation of the energy savings with comfort in buildings using soft computing techniques." Energy and Buildings 43(1): 66-74.

Dunlop, J., K. Burnham, et al. (1995). Application of fuzzy logic based self-tuning control. Industrial Automation and Control, 1995 (IA & C'95), IEEE/IAS International Conference on (Cat. No. 95TH8005), IEEE.

East, E. W. (2014, August 4, 2014). "Construction Operations Building Information Exchange (COBie)." Retrieved Feburary 13, 2015, from http://www.wbdg.org/resources/cobie.php.

Federspiel, C. C., W. J. Fisk, et al. (2004). "Worker performance and ventilation in a call center: analyses of work performance data for registered nurses." Indoor Air 14: 41-50.

Harrod, G., B. Rigg, et al. (2011). HVAC System Controller Configuration. U. S. Patent, Johnson Controls Technology Company. US 8,078,326 B2.

Huang, W. and H. N. Lam (1997). "Using genetic algorithms to optimize controller parameters for HVAC systems." Energy and Buildings 26(3): 277-282.

Husaunndee, A., R. Lahrech, et al. (1997). SIMBAD: A simulation toolbox for the design and test of HVAC control systems. Proceedings of the 5th international IBPSA conference.

Hydeman, M. and B. Eubanks (2015). ASHRAE Research Project Report 1455-RP: Advanced Control Sequences for HVAC Systems – Phase I Air Distribution and Terminal Systems. Atlanta, GA, American Society of Heating, Refrigerating and Air-Conditioning Engineers: 106.

Hydeman, M., S. Taylor, et al. (2003). Advanced Variable Air Volume System Design Guide. Energy Commission. State of California,.

International Energy Agency (IEA) (2014). Transition to Sustainable Buildings: Strategies and Opportunities to 2050, International Energy Agency.

International Organization for Standardization (2013). ISO 16739:2013 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries

Johnson, W. S. (1883). Electric tele-thermoscope. U.S. Patent Office. United States.

Karmacharya, S., G. Putrus, et al. (2012). Thermal modelling of the building and its HVAC system using Matlab/Simulink. Environment Friendly Energies and Applications (EFEA), 2012 2nd International Symposium on.

Kephart, J. O. and D. M. Chess (2003). "The vision of autonomic computing." Computer 36(1): 41-50.

Khan, M. W., M. A. Choudhry, et al. (2015). "Adaptive fuzzy multivariable controller design based on genetic algorithm for an air handling unit." Energy 81(0): 477-488.

Kim, J. B., W. Jeong, et al. (2015). "Developing a physical BIM library for building thermal energy simulation." Automation in Construction 50(0): 16-28.

Klepeis, N. E. (2001). "The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants." Journal of Exposure Analysis and Environmental Epidemiology 11(3): 231-252.

Kusiak, A., M. Li, et al. (2010). "Modeling and optimization of HVAC energy consumption." Applied Energy 87(10): 3092-3102.

Kusiak, A. and G. Xu (2012). "Modeling and optimization of HVAC systems using a dynamic neural network." Energy 42(1): 241-250.

Kusiak, A., G. Xu, et al. (2011). "Optimization of an HVAC system with a strength multi-objective particle-swarm algorithm." Energy 36(10): 5935-5943.

Liu, X., B. Akinci, et al. (2013). "Extending the information delivery manual approach to identify information requirements for performance analysis of HVAC systems." Advanced Engineering Informatics 27(4): 496-505.

Lu, L., W. Cai, et al. (2005). "HVAC system optimization—in-building section." Energy and Buildings 37(1): 11-22.

Lucas, J., T. Bulbul, et al. (2013). "An object-oriented model to support healthcare facility information management." Automation in Construction 31: 281-291.

Ma, Y., G. Anderson, et al. (2011). A distributed predictive control approach to building temperature regulation. American Control Conference (ACC). San Francisco, CA IEEE: 2089-2094.

Ma, Z. (2005). The Fuzzy Object-Oriented Databases. Fuzzy Database Modeling with XML, Springer US. 29: 159-186.

Maeda, M. and S. Murakami (1992). "A Self-Tuning Fuzzy Controller." Fuzzy Sets and Systems 51(1): 29-40.

Martin, A. J. and C. P. Banyard (1998). Library of system control strategies. Bracknell, UK, Building Services Research and Information Association (BSRIA).

MathWorks (2012). PID Controller (2 DOF): Compensator formula (2012b).

MathWorks. (2012). "Simulink: Simulation and Model-Based Design." Retrieved Nov. 29, 2012, from http://www.mathworks.com/products/simulink/.

MathWorks. (2013). "When to Enable Zero-Crossing Detection." Retrieved April 21, 2013, from http://www.mathworks.com/help/stateflow/ug/when-to-enable-zero-crossing-detection.html.

May-Ostendorp, P., G. P. Henze, et al. (2011). "Model-predictive control of mixed-mode buildings with rule extraction." Building and Environment 46(2): 428-437.

Microsoft. (2011). "Field2.SaveToFile Method (DAO)." Retrieved Feb 1, 2015, from https://msdn.microsoft.com/en-us/library/office/ff191852%28v=office.15%29.aspx.

Montgomery, R. and R. McDowall (2008). Chapter 9 - Control Diagrams and Sequences. Fundamentals of HVAC Control Systems. Oxford, Elsevier: 216-249.

Moroşan, P.-D., R. Bourdais, et al. (2010). "Building temperature regulation using a distributed model predictive control." Energy and Buildings 42(9): 1445-1452.

Mossolly, M., K. Ghali, et al. (2009). "Optimal control strategy for a multi-zone air conditioning system using a genetic algorithm." Energy 34(1): 58-66.

Naidu, D. S. and C. G. Rieger (2011). "Advanced control strategies for heating, ventilation, air-conditioning, and refrigeration systems-An overview: Part I: Hard control." HVAC&R Research, 17(1): 2-21.

Nesler, C. G. (1986). "Adaptive control of thermal processes in buildings." Control Systems Magazine, IEEE 6(4): 9-13.

Procyk, T. J. and E. H. Mamdani (1979). "A linguistic self-organizing process controller." Automatica 15(1): 15-30.

Ramesh, T., R. Prakash, et al. (2010). "Life cycle energy analysis of buildings: An overview." Energy and Buildings 42(10): 1592-1600.

Riederer, P. (2005). Matlab/Simulink for building and HVAC simulation-State of the art. Ninth International IBPSA Conference.

Roth, K. W., D. Westphalen, et al. (2005). Energy Impact of Commercial Building Controls and Performance Diagnostics: Market Characterization, Energy Impact of Building Faults and Energy Savings Potential. J. R. (DOE). Cambridge, MA, TIAX LLC.

Ryan, J. D. and R. K. Shah (2007). Self-configuring controls for heating, ventilating and air conditioning systems. U.S. Patent Office. United States, Carrier Corporation. US 7,243,004 B2.

Scattolini, R. (2009). "Architectures for distributed and hierarchical Model Predictive Control – A review." Journal of Process Control 19(5): 723-731.

Schein, J. (2007). "An information model for building automation systems." Automation in Construction 16(2): 125-139.

Scheuer, C., G. A. Keoleian, et al. (2003). "Life cycle energy and environmental performance of a new university building: modeling challenges and design implications." Energy and Buildings 35(10): 1049-1064.

Seem, J. E. (2013). Process control systems and methods having learning features, Google Patents.

Shao, S. (1988). "Fuzzy self-organizing controller and its application for dynamic processes." Fuzzy Sets and Systems 26(2): 151-164.

Soyguder, S. and H. Alli (2009). "An expert system for the humidity and temperature control in HVAC systems using ANFIS and optimization with Fuzzy Modeling Approach." Energy and Buildings 41(8): 814-822.

Soyguder, S., M. Karakose, et al. (2009). "Design and simulation of self-tuning PID-type fuzzy adaptive control for an expert HVAC system." Expert Systems with Applications 36(3, Part 1): 4566-4573.

Stephens, R. (2010). Chapter 2: Database Types. Beginning Database Design Solutions. R. Elliott, S. Jones, S. Hoberman, A. Smith and K. Cofer. Indianapolis, IN Wiley.

Sundell, J. (2004). "On the history of indoor air quality and health." Indoor Air 14: 51-58.

Sundell, J., H. Levin, et al. (2011). "Ventilation rates and health: multidisciplinary review of the scientific literature." Indoor Air 21(3): 191-204.

Tashtoush, B., M. Molhim, et al. (2005). "Dynamic model of an HVAC system for control analysis." Energy 30(10): 1729-1745.

The White House (2011). We Can't Wait: President Obama Announces Nearly $4 Billion Investment in Energy Upgrades to Public and Private Buildings. WASHINGTON, DC, Office of the Press Secretary.

Trčka, M. and J. L. M. Hensen (2010). "Overview of HVAC system simulation." Automation in Construction 19(2): 93-99.

U.S. DOE. (2003). "Commercial Sector Characteristics, Building Energy Data Book." Retrieved Oct. 4, 2014, from http://buildingsdatabook.eren.doe.gov/TableView.aspx?table=3.2.5.

U.S. DOE. (2010). "3.1.4 2010 Commercial Energy End-Use Splits, by Fuel Type (Quadrillion Btu) " Building Energy Data Book Retrieved Oct. 28, 2013, from http://buildingsdatabook.eren.doe.gov/TableView.aspx?table=3.1.4.

U.S. DOE. (2010). "Buildings Share of U.S. Primary Energy Consumption (Percent)." Buildings Sector Energy Consumption, from http://buildingsdatabook.eren.doe.gov/TableView.aspx?table=1.1.3.

U.S. DOE. (2014). "Building Energy Data Exchange Specification (BEDES)." BEDES Dictionary 1.0 Retrieved Dec. 27,, 2014, from http://energy.gov/eere/buildings/building-energy-data-exchange-specification-bedes.

Underwood, C. P. (1999). HVAC Control System: Modelling, Analysis and Design. New York, , E & FN Spon.

Utterson, E. (2006). Temperature Control Systems – Why Don't They Work? Solutions: The Role of the Consulting Engineer.

Vieira, G. E. and C. P. Veiga (2009). Hierarchical and centralized architectures for distributed production planning, scheduling and control activities. Service Operations, Logistics and Informatics, 2009. SOLI '09. IEEE/INFORMS International Conference on.

Wetter, M. (2009). "Modelica-based modelling and simulation to support research and development in building energy and control systems." Journal of Building Performance Simulation 2(2): 143-161.

Wright, J. A., H. A. Loosemore, et al. (2002). "Optimization of building thermal design and control by multi-criterion genetic algorithm." Energy and Buildings 34(9): 959-972.

Wruck, R. A. (2008). Indoor air quality and economizer control methods and controllers. U.S. Patent Office. United States, Honeywell International Inc. US 7,434,413 B2.

Wu, Z., R. V. N. Melnik, et al. (2007). "Model-based analysis and simulation of airflow control systems of ventilation units in building environments." Building and Environment 42(1): 203-217.

Xu, X., S. Wang, et al. (2004). "A robust sequencing control strategy for air-handling units." Building Services Engineering Research and Technology 25(2): 141-158.

Zeldman, J. (2001). Taking your talent to the Web: a guide for the transitioning designer, New Riders.

Zhou, X. and J. E. Braun (2007). "A simplified dynamic model for chilled-water cooling and dehumidifying coils—Part 1: Development (RP-1194)." HVAC&R Research, 13(5): 785-804.

Zhou, X. and J. E. Braun (2007). "A simplified dynamic model for chilled-water cooling and dehumidifying coils - Part 2: Experimental validation (RP-1194)." HVAC&R Research, 13(5): 805-817.

Zobel, J. (2004). Writing for Computer Science. London, UK, Springer

Zupančič, B. and A. Sodja (2011). Advanced Multi-Domain Modelling: Advantages and Dangers. 2011 UkSim 13th International Conference on Computer Modelling and Simulation (UKSim), .

Zupančič, B. and A. Sodja (2013). "Computer-aided physical multi-domain modelling: Some experiences from education and industrial applications." Simulation Modelling Practice and Theory 33(0): 45-67.

APPENDIX A: ELEMENTS FORMATTING FOR A TYPICAL CONTROL MODULE

Elements	**Attributes**	**Data Type**	**Format (explanation)**		
			Local module	**System module**	**Building module**
Control module name	Symbol	text	L-#(component)_#(variable)_###(control algorithm)	S-##(system)-_#(variable)_###(control algorithm)	B-_#(variable)_###(control algorithm)
	Narratives	text	Local control module-(component name) (variable) using (control algorithm)	System control module-(system name) (variable) using (control algorithm)	Building control module-(variable) using (control algorithm)
Control objective	Narratives	text	To modulate the controlled variable with the purpose doing something		
Operation mode	Symbol	text	English words		
	Narratives	text	Provide (functionality)		
System schematic	Symbol	text	Module ID - sch		
	Diagram	file	Attached diagram		
Control flow diagram	Symbol	text	Module ID - cfd		
	Diagram	file	Attached diagram (see Appendix B: Example-A VAV Box with Reheat Control Module (control flow diagram section))		
Data point (Inputs /Outputs /Parameter)	Symbol	text	###(variable characteristic 1)-### (variable location)-## (type)		
	Narratives	text	(Detailed description of the variable, including the variable location, measured media)		
	Unit	text	(SI or PI units or Unitless)		
	Source	text	Sensor/Virtual/Manual		
Alarms	Symbol	text	AL - ##(characteristic1) - ##(characteristic2) - ###(location)		
	Narratives	text	Describe the alarm		
	Logic	text	If Xi is in Yi, then triggered the alarm		
Control sequences	Symbol	text	### (using three letters to represent operation mode)		
	Mode	text	Use the mode symbol in the Operation mode section		
	Narratives	text	see Appendix B: Example-A VAV Box with Reheat Control Module (control sequence section)		
Functions / algorithms	Symbol	text	Fcn - ###(algorithm)		
	Narratives	text	Name of the algorithm and functional description of this algorithm		
	F-inputs	text	Algorithm inputs		
	F-outputs	text	Algorithm outputs		
	F-Parameters	text	Algorithm parameters		
	Formula	text	Mathematical description of the algorithm		
	Reference	text	Author-Year (of the related reference for this algorithm)		
Programming code	Symbol	text	Module ID - C - Version #		
	Attachment	file	Attachment (see Table 2-4 for different types comparison)		

APPENDIX B: EXAMPLE-A VAV BOX WITH REHEAT CONTROL MODULE

- **Control module name**

Symbol	Narratives
L-Z1-VAV_D	Local control module - VAV damper position for Zone #1

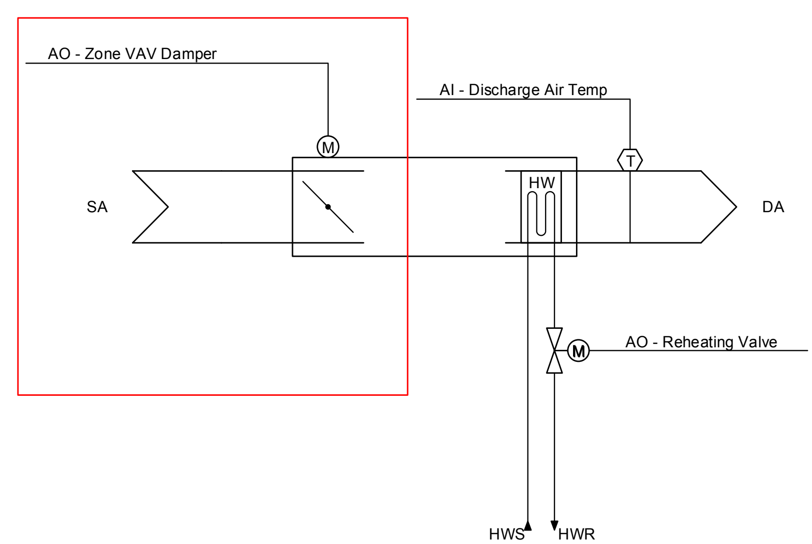
- **Control objective**

Narratives
To modulate the VAV damper position with the purpose of realizing different operation scenarios.

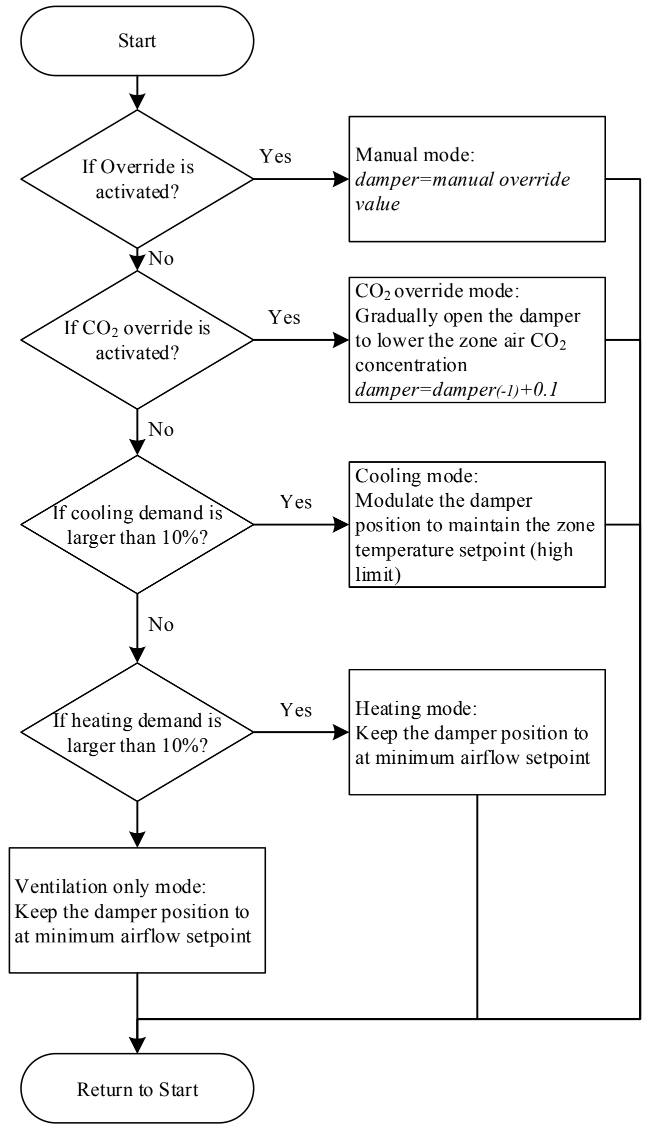
- **Operation mode**

Symbol	Narratives
Manual override	Provide manual operation
Ventilation only	Provide ventilation (only) required by code
Cooling	Provide appropriate airflow rate during cooling mode
Heating	Provide appropriate airflow rate during heating mode
CO2 override	Provide CO_2 override control to keep indoor CO_2 level satisfy the requirement

- **System schematic**

Symbol	Diagram
L-Z1-VAV_D-sch	

- **Control flow diagram**

Symbol	Diagram
L-Z1-VAV_D-cfd	

- **Data point**

Symbol	Narratives	Unit	Source
C-Z1O-S	CO_2 override signal	unitless	virtual
Coo-Z1-V	Zone #1 cooling demand percentage	%	virtual
Hea-Z1-V	Zone #1 heating demand percentage	%	virtual
Z-Z1D-S	Zone #1 VAV damper position	unitless	virtual
Z-Z1D-O	Zone #1 VAV damper position manual override	unitless	manual

- **Alarm**

Symbol	**Narratives**	**Logic**
AL-H-T-ZA	High temperature alarm	Triggered: if the zone temperature is higher than high threshold value for more than 3 minutes
AL-L-T-ZA	Low temperature alarm	Triggered: if the zone temperature is lower than low threshold value for more than 3 minutes
AL-H-h-ZA	High humidity alarm	Triggered: if the zone humidity is higher than high threshold value for more than 3 minutes
AL-L-h-ZA	Low humidity alarm	Triggered: if the zone humidity is higher low threshold value for more than 3 minutes

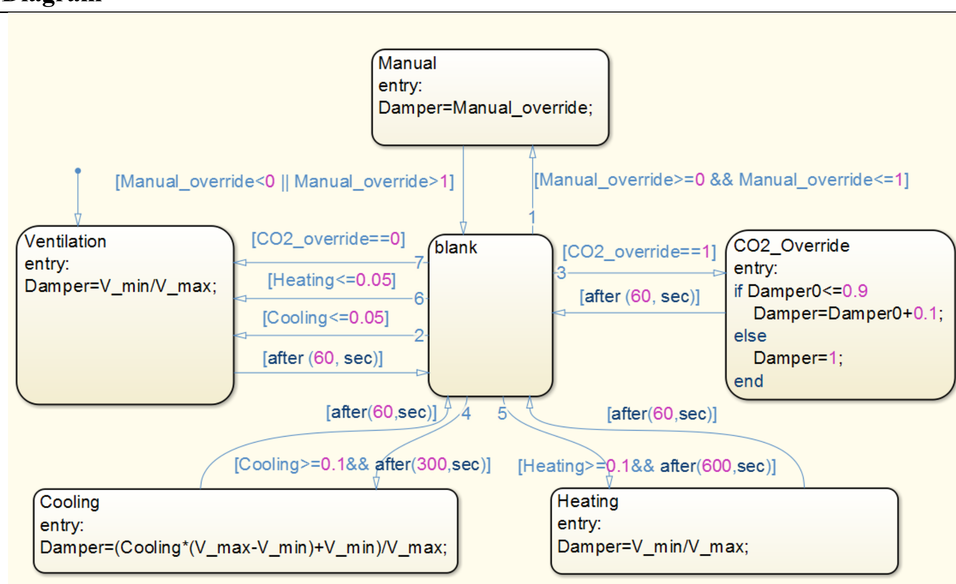
- **Control sequence**

Symbol	**Mode**	**Narratives**
L-Z1-VAV_D_seq-1	Manual	IF the manual override is activated, THEN the damper is under the Manual operation mode: i.e., the damper position is determined by the manual override signal.
L-Z1-VAV_D_seq-2	Ventilation	IF the manual override is NOT activated AND the CO_2 override is NOT activated, AND cooling demand is smaller than 5%, AND heating demand is smaller than 5%, THEN the damper is under the Ventilation only operation mode: i.e., the damper is controlled to maintain the minimum airflow setpoint.
L-Z1-VAV_D_seq-3	Heating	IF the manual override is NOT activated AND the CO_2 override is NOT activated, AND cooling demand is smaller than 5%, AND heating demand is larger than 10%, THEN the damper is under the Heating operation mode: i.e., the damper is controlled to maintain the minimum airflow setpoint (see reheat coil valve for heating temperature control).
L-Z1-VAV_D_seq-4	Cooling	IF the manual override is NOT activated AND the CO_2 override is NOT activated, AND cooling demand is larger than 10%, AND heating demand is smaller than 5%, THEN the damper is under the Cooling operation mode: i.e., the damper is controlled to maintain the lower zone temperature setpoint by adjusting the damper to modify the cooling airflow rate.
L-Z1-VAV_D_seq-5	CO_2 override	IF the manual override is NOT activated AND the CO_2 override is activated, THEN the damper is under the CO_2 override operation mode: i.e., the damper is gradually increased to lower the indoor CO_2 concentration to the setpoint level (default 1000 ppm).

- **Functions/algorithms**

Symbol	Narratives	Formula	Inputs	Outputs	Parameters	Reference
Fcn-Z-VAV-Coo	Damper opening calculation for cooling	$D = \frac{(cooling*(V_{max}-V_{min})+V_{min})}{V_{max}}$	Cooling demand $(cooling)$; Minimum airflow rate (V_{min})	Damper position (D)	Maximum airflow rate design (V_{max})	Engineering practices
Fcn-Z-VAV-Ven	Damper opening calculation for ventilation and heating	$D = \frac{V_{\min}}{V_{max}}$	Minimum airflow rate (V_{min})	Damper position (D)	Maximum airflow rate design (V_{max})	Engineering practices

- **Programming code**

Symbol	Diagram
L-Z1-VAV-C-0	

VITA - Yan Chen

EDUCATION

The Pennsylvania State University	University Park, PA
Ph.D. Architectural Engineering (mechanical option)	2015
Purdue University	West Lafayette, IN
M.S.E. Agricultural & Biological Engineering (focuses on air quality)	2010
Tianjin University	Tianjin, China
B.E. Electrical Engineering (focuses on automation)	2008

WORKING EXPERIENCE

Graduate Research Assistant – The Pennsylvania State University	2011-2015
Commissioning Intern – Sebesta Inc., Arlington, VA	Summer 2014
HVAC Control Lab Assistant – The Pennsylvania State University	Spring 2014, 2015
Research Specialist – University of Missouri, Columbia, MO	2010-2011
Graduate Research Assistant – Purdue University	2008-2010

SELECTED PEER-REVIEWED JOURNAL ARTICLES

Chen, Y. and S. Treado, Development of a simulation platform based on dynamic models for HVAC control analysis*, Energy and Buildings*, 2014, 68, Part A, 376-386.

Treado, S. and **Y. Chen**, Saving building energy through advanced control strategies. *Energies* special issue: *Energy Efficient Building Design,* 2013, 6, 4769-4785.

Chen, Y., J. Q. Ni, C. A. Diehl, A. J. Heber, and , L.-L., Chai. Large scale application of vibration sensors for fan monitoring at commercial layer hen houses. *Sensors* 2010, 12:11590-11604.

Chen, Y., S. Treado, and J. Messner. Building HVAC control knowledge data schema-Towards a unified representation of control system knowledge. *Automation in Construction*, Under Review.

Chen, Y., S. Treado, and J. Messner.. Development of an integrated database for building control knowledge management, *Advanced Engineering Informatics,* Submitted.

SELECTED CONFERENCE PAPERS & PRESENTATIONS & POSTERS

Chen, Y. and S. Treado. 2015. Poster: An Integrated method for building control self-configuration, *U.S. DOE Building Technologies Office Peer Review*, April 2015, Vienna, VA

Chen, Y. and S. Treado. 2014. Self-Configuration of Building Control System Using Knowledgebase. *ASHRAE Annual Conference*, Seattle, WA

HONORS & AWARDS

The International Society of Automation (ISA) Educational Foundation Scholarship	2014
Graduate Student Travel Grant, the Pennsylvania State University	2014
Laraine and Jack Beiter Excellence Endowment Scholarship in Architectural Engineering, the Pennsylvania State University	2013

PROFESSIONAL ACTIVITIES

Journal Reviewer, *Energy and Buildings, HVAC&R Research Journal, Sensors*

ASHRAE Student member, technical committee (TC) Corresponding member (TC 1.4, 7.5)