CITY UNIVERSITY OF HONG KONG

香港城市大學

Event-Driven Optimization for Complex HVAC Systems

基於事件驅動的複雜空調系統優化策略研究

Submitted to
Department of Architecture and Civil Engineering
建築學與土木工程學系
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
哲學博士學位

By

WANG Junqi
王俊淇

August 2017
二零一七年八月

ABSTRACT

Improving the operating efficiency of heating, ventilating and air-conditioning (HVAC) systems is important since a small improvement can lead to substantial energy savings, especially for large-scale complex HVAC systems. Real-time optimization (RTO) is an efficient tool to improve the operating efficiency. Almost all traditional RTO methods utilize the time-driven optimization (TDO) mechanism, in which optimization actions are triggered by time in a periodic manner. However, with the increasing complexity of HVAC systems, the traditional TDO mechanism encounters challenges. Since the state variables in complex HVAC systems (e.g. weather and occupancy conditions) are highly stochastic, the TDO mechanism can easily lead to unfavorable optimization actions, e.g. delayed or unnecessary actions. This is because the TDO with a fixed optimization frequency cannot capture the stochastic changes of the state variables promptly. Increasing the optimization frequency is a simple way to improve the performance of TDO, but it will increase the online computational load. As a result, the TDO can hardly achieve a good balance between the optimization performance (e.g. energy efficiency) and the online computational load, which restricts its practical applications.

The limitations associated with the TDO mechanism call for a reformulation of real-time optimization strategies, which motivate us to develop a new optimization mechanism. The mechanism of event-driven optimization (EDO) is therefore proposed and investigated in this study. The key idea of the event-driven optimization is to use “event” rather than “time” to trigger optimization actions. Because it can realize the concept of “taking optimization actions only when necessary”, the EDO mechanism has potential capability for applications in complex HVAC systems to reduce the computational resource utilization while ensure the system performance. To investigate the potential capability systematically, this thesis develops a comprehensive EDO framework and a design procedure for complex HVAC systems.

Firstly, the EDO framework is established, which contains the EDO strategy, optimal control diagram, and fundamental terms associated with the event. An event is formally defined as a set of state transitions. The core of the EDO framework is the {event, policy, action} structure, which works based on the principle that when event occurs, an action is taken based on the policy (which links events with actions). To facilitate the implementations of the EDO, event attributes, types and mathematical representations are also synthesized.

Secondly, to guarantee and improve the optimization performance of the EDO, a design approach is developed according to the EDO framework of {event, policy, action}. Because state transitions (i.e. events) are numerous in a complex HVAC system, a methodology is developed to identify and establish the event space, which includes three steps to address the problems of identifying critical state transitions, event definition, and event space optimization. Both direct and indirect methods are developed for event space establishment. The direct method is constructed based on the system COP (coefficient of performance) deviation. Two indirect methods are developed based on prior-knowledge and data mining, and are thus termed the knowledge-based and data-based methods.

The effectiveness and performances of these methods are demonstrated through the cases studies performed on the simulation platform. Results suggested that when the system dynamics are stochastic and difficult to predict, the EDO strategy is able to adapt to the changing environment because it has a quicker response to environmental changes. Results also show that the EDO strategy can effectively reduce the computational load while not sacrifice the energy performance because unnecessary optimization actions can be avoided.

Keywords: real-time optimization, HVAC, event-driven optimization, complex system

CITY UNIVERSITY OF HONG KONG
Qualifying Panel and Examination Panel

Surname:	WANG
First Name:	Junqi
Degree:	PhD
College/Department:	Department of Architecture and Civil Engineering

The Qualifying Panel of the above student is composed of:

Supervisor(s)

Dr. HUANG Gongsheng	Department of Architecture and Civil Engineering City University of Hong Kong

Qualifying Panel Member(s)

Dr. CHEN Jiayu	Department of Architecture and Civil Engineering City University of Hong Kong
Dr. LAM Joseph Choi	Department of Architecture and Civil Engineering City University of Hong Kong

This thesis has been examined and approved by the following examiners:

Dr. LEE Wai Ming	Department of Architecture and Civil Engineering City University of Hong Kong
Prof. LAI Chi Keung Alvin	Department of Architecture and Civil Engineering City University of Hong Kong
Dr. HUANG Gongsheng	Department of Architecture and Civil Engineering City University of Hong Kong
Dr. LU Lin	Department of Building Services Engineering The Hong Kong Polytechnic University

PUBLICATIONS ARISING FROM THIS THESIS

Journal papers published:

1. **J.Q. Wang**, G.S. Huang, Y.J. Sun, X.P. Liu, Event-driven optimization of complex HVAC systems, Energy and Buildings 133 (2016) 79-87 (**IF: 4.067**, JCR Q1).

Journal papers submitted:

1. **J.Q. Wang**, Q.S. Jia, G.S. Huang, Event space design approach for event-driven optimization in complex building air conditioning systems, submitted to Applied Energy, 2017.
2. P. Zhou, **J.Q. Wang**, G.S. Huang, J.J. Gao, A simple temperature modelling method for air-conditioned large-scale open (indoor) spaces, submitted to Building and Environment, 2017.
3. W. Wang, **J.Q. Wang**, G.S. Huang, J. Chen, Occupancy-based outdoor air control in multi-zone HVAC systems through Wi-Fi probe enabled occupancy prediction approach, submitted to Energy and Buildings, 2017.

Conference papers:

1. **J.Q. Wang***, SW Lou, P. Zhou, G.S. Huang, A design approach for event-driven

optimization in building complex air conditioning systems, IEEE-CASE, Xi'an, China, 2017 (**accepted**).

2. **J.Q. Wang***, P. Zhou, and G.S. Huang, A data mining approach to discover critical events for event-driven optimization in building air conditioning systems, WES-CUE, Singapore, 2017 (**accepted**).
3. **J.Q. Wang***, G.S. Huang, P. Zhou, Event-driven optimal control of complex HVAC systems based on COP·mins, ICAE, Beijing, 2016. (**EI**)
4. **J.Q. Wang**, G.S. Huang, Y.J. Sun. Optimal control of complex HVAC systems: event-driven or time-driven optimization? CLIMA, Aalborg, Denmark, 2016.
5. **王俊淇***,黄公胜,孙勇军,"基于事件驱动的空调系统实时优化控制",第二十届全国暖通空调制冷学术年会,海口,2016.
6. P. Zhou, **J.Q. Wang**, G.S. Huang, An evaluation of heat transfer coefficient in an independent zonal temperature controls with CFD, ICAE, Beijing, 2016. (**EI**)
7. W. Wang, X. Song, **J.Q. Wang**, J. Chen, Supervisory demand-based multi-room ventilation strategy: outdoor air flow rate optimization based on real time location and multi objective evolutionary algorithm, the 10th ISHVAC, Jinan, China, 2017.

ACKNOWLEDGEMENTS

First and foremost I want to thank my supervisor Dr. HUANG Gongsheng, Associate Professor, for his continuous support and patient guidance. Together with Dr. HUANG, it has been an interesting journey in my Ph.D. life to learn, discuss and develop ways to solve various problems arising in my research. His passion and rigorous attitude to research make the excellent example for me. Besides a teacher, he is also a nice guy to get along with in various ways, such as badminton, hiking and social gathering. Besides, I would like to express my heartfelt appreciation to Dr. SUN Yongjun, for his constructive suggestions and help. My special thanks also go to my qualifying panel members Dr. LAM Joseph Choi and Dr. CHEN Jiayu for the comments and advices on my research. Moreover, I really appreciate Dr. JIA (Samuel) Qing-Shan for hosting me in Tsinghua University and proving an excellent learning and research environment.

Secondly, I am truly grateful to my colleagues and friends, namely, Zhou Pei, Huang Pei, Liao Yundan, Li Yantong, Zhang Linfeng, Zhang Sheng, Lou Siwei, Wang Wei, Gao Jiajia, Wang Jiaqiang, Li Jun, Chen Xingbin, Wu Shanshan, Chen Chun, Chen Zhuang and guys in ACE Dept.. The long road is memorable because of their help and support. Thanks for my best friend in Suzhou, Song Qiang, who never hang up my calls.

Thirdly, I am deeply grateful to my parents Mr. Wang Linfu and Mrs. Ji Hong for their unconditional love, trust and support. Finally, my sincere gratitude to Shi (Jane) Xian for her accompany, care and love. I cannot complete this thesis without her.

The financial supports of studentship and conference grants from Hong Kong government and City University of Hong Kong are gratefully acknowledged.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

ST	sampling time
X	system state (vector)
x	state variable
τ_i	time index at step i
e	event
PLR	part-load ratio
SCOP	system coefficient of performance
CH	chiller
E_{space}	event space
A_{space}	action space
σ	threshold
U	uncontrolled variables
RH	relative humidity (%)
P	power (kW)
T	temperature (ºC)
δ	perturbation
T_{apr}	difference between the leaving water temperature and incoming air wet-bulb temperature of cooling tower (ºC)
Δh	enthalpy difference between the saturated (at cooling tower inlet water temperature) and incoming air of cooling tower (kJ/kg)
$\Delta T_{step\ change}$	temperature difference for each step change (ºC)
$Power$	power requirement
EC	energy consumption
CC	computation consumption

$ES\%$	energy saving percentage
$CS\%$	computation saving percentage
PS	performance score
$Freq$	frequency
M	mass flow rate
Q	capacity

Subscripts

chw	chilled water
scw	supply cooling water
schw	supply chilled water
sa	supply air
prm	primary
sec	primary
wb	wet-bulb
db	dry-bulb
w	water
sys	system
tot	total
ct	cooling tower
rej	rejection
ch	chiller
ev	evaporator
cd	condenser

pump	pump
fan	fan
lower	lower bound
upper	upper bound
min	minimal
max	maximal
BC	base case or benchmark case
trans.	transient
accum.	accumulated

CHAPTER 1. INTRODUCTION

1.1 Background

The buildings sector accounts for 20.1% of the total (delivered) energy consumed worldwide (U.S. Department of Energy 2016), where heating, ventilating and air conditioning (HVAC) systems contribute the major proportion (around 50%) in developed countries (Pérez-Lombard, Ortiz & Pout 2008). With the growing global population, enhancement of building services and comfort levels, together with the rise in time spent indoors, building energy consumption is expected to continuously increase in the near future (Pérez-Lombard, Ortiz & Pout 2008). Improving the energy efficiency of HVAC systems is thus important, and real-time optimization (RTO) is a powerful tool to improve the operating efficiency.

RTO is a model-based, closed-loop process control strategy, which aims to make the ‘‘best’’ decisions to optimize, or to improve, the operating performance of a system based on the information obtained by observing and analyzing the system’s behavior (Darby et al. 2011). RTO is also known as optimal control (Evans 2005) or supervisory control (Wang, Ma 2008), but the difference is that RTO focuses on small-time scale (e.g. hour) (Darby et al. 2011); while the other two are more general and do not have such constraint. Most of the current RTO methods use the periodic optimization strategy, which adopts the so-called “time-driven optimization” (TDO) mechanism. In the TDO mechanism, control actions are triggered by “time” in a periodic way. For instance, Kusiak, Li and Tang (2010) demonstrated that 7.66% of the total energy could be saved by hourly updating of the supply air pressure and temperature.

In a relatively simple and small-scale system, the TDO mechanism works well as demonstrated by plenty of studies (Kusiak, Li & Tang 2010). However, modern engineering systems become increasingly complex as the technology continually develops, such as the transportation, energy and building systems, so does the HVAC system (Cassandras, Lafortune 2009). One major problem is that state transitions in complex systems, such as changes in occupancy and weather conditions, are numerous, highly stochastic and difficult to predict. The traditional TDO is no longer efficient when reacting to stochastic (aperiodic) changes, which can be explained through a simple illustration in Figure 1. 1.

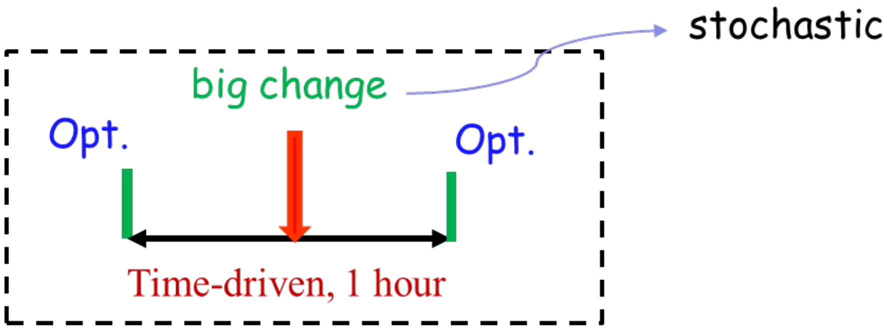

Figure 1. 1 Illustration of TDO method (Opt. = Optimization)

For example, “one optimization per hour” is used as the optimization frequency in the TDO strategy, i.e. system control settings will be updated every one hour. When a big change occurs at the middle (e.g. 30 minutes after the current optimization action), this TDO mechanism cannot respond to this change timely. In other words, the optimization action will be delayed by 30 minutes, which is detrimental to the system performance. Certainly, a higher optimization frequency could be used to reduce the delay of the optimization action. However, the computational load will increase correspondingly. Meanwhile, unnecessary optimization actions may be conducted when system is operating stably, which is a wastage of resources like the computational load and communication bus load (Sandee, Heemels & Van Den Bosch 2007).

Computational complexity will be encountered in complex HVAC systems due to the complexity of optimization problems, which is explained as follows. Firstly, complex HVAC systems always contain a large number of components or subsystems (A˚ström, Kumar 2014, Windham, Treado 2016). Secondly, multiple constraints are imposed on the operation of the systems, such as operation protections of different components (Cai 2015, Windham, Treado 2016). Moreover, coupling effects between equipment and interactions with other building systems (like lighting systems) also cannot be neglected for the optimal control (Guan et al. 2016, A˚ström, Kumar 2014, Windham, Treado 2016). These emergent features associated with the system complexity make the analysis and solution of the optimization problem difficult. The direct consequence is that the computation time becomes longer. Thus, the optimization action would be further delayed, which would be adverse to the optimization performance (e.g. energy efficiency).

In summary, the conventional TDO strategy is a periodic optimization scheme and may lead to delayed or unnecessary optimization actions when reacting to stochastic changes, which would deteriorate the optimization performance and waste resources (such as energy, computation and communication). In addition, TDO is not efficient in balancing the optimization performance and computational load, which may become a major obstacle in real implementations (Wang, Ma 2008). Considering these limitations and difficulties associated with TDO, the development of a more suitable and efficient RTO strategy is of critical importance, which is the main problem this thesis tries to tackle.

1.2 Aims and objectives

The overall aim of this study is to develop a new optimization framework, i.e. the event-driven optimization (EDO), for complex HVAC systems with a better balance between the optimization performance and computational load. The *specific objectives are listed as below:*

(1) Fundamentals: to investigate the operational characteristics of complex HVAC systems and reveal the basic operational patterns and mechanisms. Important factors affecting the operation efficiency of HVAC systems will be identified, including variables coming from outdoor weather, indoor climate, system and equipment, which would serve as a foundation of this research.

(2) EDO framework: to establish the EDO framework based on the investigation of basic operational patterns of complex HVAC systems. To facilitate this new

optimization strategy, essential components (such as event identification) and terms (such as policy, event attributes, event types and mathematical event representations) will be defined.

(3) EDO design: to a design approach for the established EDO strategy so that the desirable system performance can be achieved. Events will be selected from important state variables affecting the optimization objectives. Guidelines and algorithms will be developed to define, identify and optimize the event.

(4) Validation: to validate the feasibility and evaluate the reliability. Comprehensive studies will be conducted to validate the feasibility and evaluate the optimization performance and reliability of the proposed EDO in various operation scenarios.

1.3 Organization of the thesis

The whole thesis is divided into 8 chapters. The main content of each chapter is presented as follows.

Chapter 1 presents the background of this research, where the real-time optimization in HVAC is discussed. Challenges are briefly pointed out, and the research objectives are presented accordingly.

Chapter 2 reviews the literature in HVAC optimal control area, in which the traditional TDO strategy is discussed. The event-driven strategy, including its basic idea, applications and current practices of design are presented. The research gaps are identified and the key issues of applying event-driven strategy in HVAC are revealed.

Chapter 3 introduces the EDO framework and the simulation platform used in the thesis. The EDO framework contains the scheme, the strategy, and the optimal control diagram of the EDO. The fundamental terms of events are introduced, including the formal definition, event attributes, event types, mathematical representations and event identification. To evaluate the optimization performance, the co-simulation platform, the formulation of the optimization problem and performance indices are also presented.

Chapter 4 develops a design approach for the EDO. A five-step design procedure is introduced for building the {event, policy, action} structure. Since "event" determines the optimization performance of EDO, the general decision making flow chart, the overall methodology and the basic event selection criteria of the event space establishment are introduced. Both direct and indirect methods are suggested to identify important state transitions, which are discussed and demonstrated by the case studies in Chapter 5, 6 and 7.

Chapter 5 presents the knowledge-based method for the event space establishment. Then, an algorithm for event threshold selection is proposed for continues state variables. Using the knowledge-based method, five critical events are identified. For each event, seven different thresholds are tested based on three typical load and weather profiles. The optimization performances of events are analyzed in a comprehensive way regarding the energy saving, computation saving and performance score. Finally, suitable events are selected to form the policy, and the performance is validated.

Chapter 6 presents the data-based method for the event space establishment. This chapter firstly introduces the algorithm used to identify state transitions, the optimization reward estimator and the random forest algorithm for variable importance evaluation. Then, a simple method to compute the suitable event threshold based on the Euclidean distance between decision variable vectors is proposed. After that, the case study to demonstrate and validate the effectiveness of the proposed data-based method is presented. Finally, the results and findings are summarized.

Chapter 7 presents the direct method for event space establishment. The SCOP-deviation-based method is used to directly emulate the optimization objective, where the transient and accumulated SCOP deviations are defined. Then, the COP·mins is proposed for the calculations of the SCOP deviations. To find the reference SCOP, the artificial neural network is utilized. At last, the case study is presented in to illustrate the performance of the SCOP-deviation-based method.

Chapter 8 summarizes the work conducted in this thesis, lists out the main contributions and points out the limitations and possible future work.

CHAPTER 2. LITERATURE REVIEW

Since this thesis targets on the optimal control problems in complex HVAC systems, Section 2.1 discusses the general background in HVAC optimal control field, reviews the traditional TDO strategy and points out the challenges. Then, Section 2.2 introduces the basic idea and applications of the event-driven strategy since it could be a good alternative to the time-driven strategy. Section 2.3 discusses the current practices on the design of the event-driven strategy. Section 2.4 presets the prior knowledge in HVAC optimal control and data mining techniques, which can be used to find important state transitions affecting the optimization performance. Section 2.5 presents the discussions based on the literature review. Finally, the chapter summary is given in Section 2.6.

2.1 HVAC Optimal Control and Challenges

As the RTO is a powerful tool for the improvement of HVAC operating efficiency, a number of work and research have been done since 1980’s. Some good review works in HVAC optimal control fields are listed as follows for reference. In (Wang, Ma 2008), the developments of HVAC optimal control were extensively reviewed, which contains a clear classification of optimal control methods and lots of optimization algorithms; ASHRAE handbook surveyed the publications since 1980s (ASHRAE

2015), focusing more on engineering practices; recent work discussing more advanced techniques can be found in (Ahmad et al. 2016, Aste, Manfren & Marenzi 2016, Okochi, Yao 2016, Suganthi, Iniyan & Samuel 2015, Shaikh et al. 2014, Windham, Treado 2016).

2.1.1 Development procedure and fundamental questions of HVAC optimal control

In the past few decades, the basic procedure for developing HVAC optimal control algorithms has been established, which is shown in Figure 2. 1.

- Step 1: Specify optimization objective(s)

The first step is to specify the optimization objective. Common control objectives are energy consumption, cost, thermal comfort and IAQ (Ahmad et al. 2016). In the majority of the cases, a cost function is formulated as the minimization of energy consumption or operating cost when satisfying thermal comfort and IAQ constraints.

- Step 2: Identify decision variables

In principle, any variables affecting the optimal control objectives can be used as the decision variables. For the traditional electric HVAC systems, common decision variables are: air pressure or temperature, chilled water supply temperature and flow rate, cooling water supply temperature and flow rate, operating number of chillers, chilled-water and cooling-water pumps, cooling towers, fans and other equipment (Braun et al. 1989). These variables often have strong coupling effects (i.e. mutual influences), and rational optimal control of these decision variables can significantly reduce the operating cost (Braun et al. 1989).

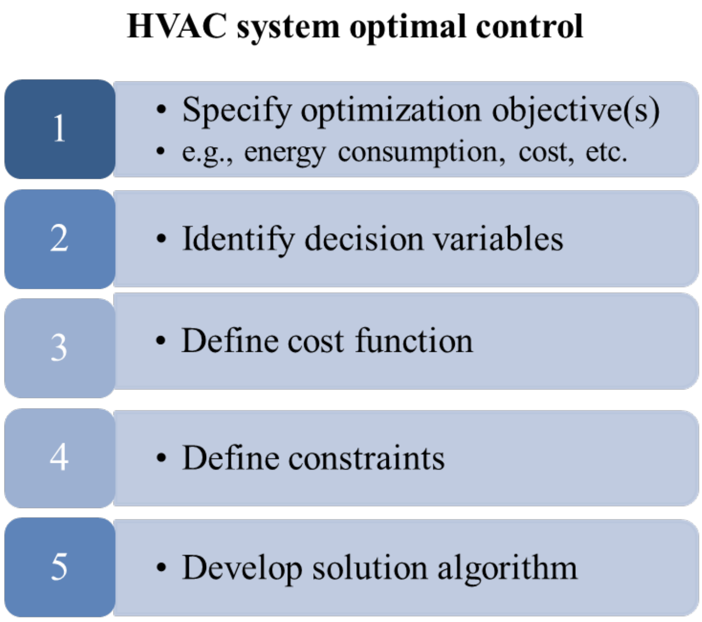

Figure 2. 1 Basic procedure for developing HVAC optimal control algorithm

- Step 3: Define cost function

The cost function represents the objective that is going to optimize. For the model-based optimal control methods (Wang, Ma 2008), the numerical models used for establishing cost function (also building performance evaluation) can be subdivided into three categories based on the models used, namely white-box, grey-box and black-box models (Aste, Manfren & Marenzi 2016, Afram, Janabi-Sharifi 2014).

- Step 4: Define constraints

The optimization problem is subject to equality or inequality constraints which need to be defined before solving the problem. An example of the equality constraint is that the sum of the capacities of operating chillers must equal to the load. An inequality constraint could be the bound of a decision variable.

- Step 5: Choose (or develop) solution algorithm

After establishing the cost function, a solution algorithm is needed to find the optimal values for decision variables. Till now, a number of optimization algorithms has been developed and verified, including classic linear/quadratic programming, gradient-based iterative methods (Braun, Diderrich 1990, Cumali 1994), evolutionary algorithms (Nassif, Kajl & Sabourin 2005), branch and bound (Fisk 2013), simulated annealing (Chang et al. 2006), particle swarm optimization, mixed integer programming and the generally used genetic algorithm (Wang, Ma 2008, Shaikh et al. 2014). As each algorithm is superior to all the others in certain aspects, suitable algorithms should be chosen, and appropriate adjustments should be made to suit the special requirements of the problem.

The HVAC system often utilizes a two-level control hierarchy for optimal control (Braun 2014), which consists of a supervisory level and a local-control level. At the supervisory level, there are *three basic questions* (summarized in Figure 2. 2) that must be answered when solving an HVAC optimal control problem. These fundamental questions are similar to the general “plant decision hierarchy” presented in (Darby et al. 2011).

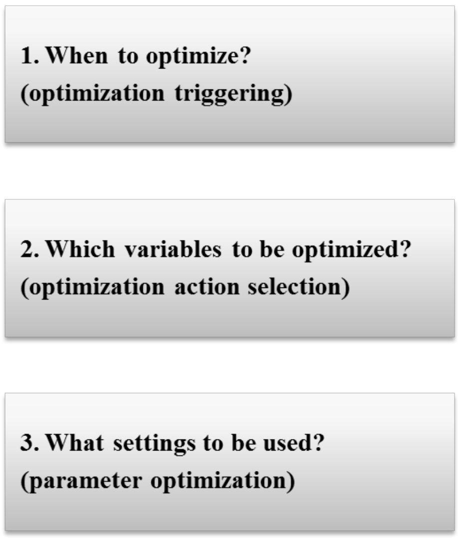

Figure 2. 2 Three basic questions in HVAC optimal control problems

The first question concerns the timing of the optimization triggering, that is, when to optimize the system. The second concerns the selection of the optimization action, that is, which decision variable(s) to be optimized. The optimization action should specify the subsystem, the equipment and the decision variable that need to be optimized. The third is the traditional parameter optimization problem, that is, what settings to be used.

Most of the literature on HVAC optimal control focuses on defining the cost functions and developing the solution algorithms; hence, the five-step procedure presented above (see Figure 2. 1) is only sufficient to answer the third question (i.e. "what settings to be used?"). The traditional strategies to handle the first and second questions are relatively simple and encounters several challenges (also briefly summarized in Section 1.1), especially as the complexity of HVAC systems increases. Details concerning these challenges will be given in the subsequent sections.

2.1.2 Traditional time-driven strategy

In the RTO problem of HVAC systems, "when to trigger the optimization" is the first basic question to be answered. The traditional TDO strategy adopts the periodic scheme to optimize the system operation with fixed frequencies. This scheme is widely used due to its simplicity and effectiveness. Note that the optimization frequency considered in the HVAC RTO problem is typically at an hourly base (Darby et al. 2011), whereas the general optimal control problem can use a much lower frequency, such as monthly resetting (Fong, Hanby & Chow 2006, 2009).

Several examples of RTO in HVAC are presented as follows. Zaheer-uddin and Zheng (2000) divided the 24-hour operation into three stages (night set-back, start-up and normal modes) to schedule the system operation. Actions were only performed when the operation stage changes. Mossolly, Ghali and Ghaddar (2009) conducted an hourly optimization for fresh air volume and supply air temperatures. In a four-month simulation of the system operation, they found that the energy consumption was reduced by 30.4%. Similarly, by hourly resetting of the supply air pressure and temperature, Kusiak, Li and Tang (2010) demonstrated that 7.66% of the total energy could be saved. They also showed that the total energy savings of four different optimization frequencies decreased monotonically as the optimization frequency decreased, i.e. 1 hour, 2 hours (saving 5.95%), 5 hours (saving 3.12%) and 50 hours (saving 0.83%) per optimization. Ma and Wang (2011) developed an online adaptive model to provide reliable and accurate estimates under dynamic operational conditions. Their results showed that about 0.73 to 2.55% of daily energy could be saved (compared with the base strategy) by periodically updating the three temperature set-points. More recently, Huang, Zuo and Sohn (2017) developed an operational support system to optimize the condenser water set point in an existing chiller plant using three frequencies, i.e. hourly, daily and weekly. Up to 9.67% of energy consumption was saved (on an annual basis) for chillers and cooling towers.

These studies reaffirm that a suitable optimization frequency is important for the performance of TDO, which is a viewpoint made nearly thirty years ago (Braun et al. 1989). However, it is difficult to select a suitable optimization frequency for practical applications due to the lack of studies. Instead, the rule of thumb is always used in practice. The choice of the optimization frequency is often based on the worst-case scenario (meaning a high optimization frequency is used) in order to guarantee acceptable performance. However, this can easily lead to a high computational load and inefficient manipulations. Firstly, when the system is running stably, the computation wastage would happen as unnecessary optimizations were performed (Liu et al. 2014, Sandee, Heemels & Van Den Bosch 2007). Secondly, the high computational load will be adverse to the system operation, which can be explained in two aspects. On the one hand, the computation required by one control updating is costly due to the large problem scale, the nonlinearity, the number of decision variables and coupling effects. Dozens of minutes are required to complete one search for the optimal settings, which would deteriorate the energy performance due to the delay of response. On the other hand, the overall computational load is huge because of a high optimization frequency (e.g. one optimization per 30 minutes). Given that the RTO is usually realized by the Building Automation System (BAS), occupying too much computation resource is detrimental to the efficiencies of other building systems (like electrical, fire, security and lighting systems).

Lowering down the optimization frequency allows the computational load to be reduced, but the optimization performance cannot be guaranteed due to the possibility of delayed actions (Huang, Zuo 2014). Therefore, a compromise between the computational load and energy performance must be considered. However, this is always challenging in practical applications since we lack quantitative studies on the relationship between the optimization frequency and energy or computational performance (Huang, Zuo 2014). In fact, it is inherently difficult to obtain this quantitative relationship since many optimal control actions are not time-driven.

A recent investigation of legacy chiller plants (Huang, Zuo & Sohn 2017) found that the optimization frequency had no significant impact on the energy saving when the wet bulb temperature (daily and weekly) did not vary significantly. In other words, the optimization frequency could be safely reduced in suitable occasions, which both ensures the system performance and saves the computation. In summary, the conventional TDO method is difficult to achieve a good balance between energy and computational performances due to the inefficient manipulations (i.e. delayed or unnecessary actions). From the literature, it is also evident that a better optimal control strategy can be formulated by incorporating the changes of the operational condition into the optimal control strategy. Therefore, it is of great interest to develop a new optimization strategy that is driven by operational condition changes for the more efficient operation of HVAC systems.

2.2 Event-driven Strategy: Introduction and Applications

Fortunately, with the advance in control theory, the idea of event-driven strategy may be promising to solve the aforementioned difficulties associated with the time-driven strategy. Thus, the so-called event-driven strategy will be introduced in this section. In the automatic control area, the significance of the event-driven (also known as "event-based" or "event-triggered") strategy was noted in the 1980s as it could be a more efficient alternative to the time-driven strategy in many applications (Cassandras 2014). Some early attempts of implementations can be tracked back to the 1990s

(Hendricks et al. 1994, Arzén 1999, Heemels et al. 1999, Bernhardsson, Aström 1999). Basically, the research of the event-driven strategy is principally driven by three communities (i.e. control, computing and communication communities) (Arzén 1999). This section mainly discusses the research and development in the control area as it is the most relevant research area to the topic of this thesis.

2.2.1 Introduction of event-driven strategy

As we known, the majority of the research and practical work in control considers time-driven sampling (or periodic sampling), mainly due to the well-established system theory and time-driven control systems (Arzén 1999). However, there are cases where it is interesting to consider the control systems based on the event-driven sampling (so-called event-driven control) in which actions are event-driven rather than time-driven. In fact, many processes in modern complex systems are driven by instantaneous "events" (Cassandras, Lafortune 2009), such as automated manufacturing systems, intelligent transportation systems, and advanced BASs. Substantial activities in these systems are managed by man-made control rules that have the event-driven properties (Arzén 1999). For instance, when the current full capacity is not enough, it is desirable to turn on an additional equipment in the plant.

Clearly, this activity is not suitable to be characterized by time-driven behaviors, and it will be inherently inefficient to use the time-driven strategy to solve the problem.

The event-driven control has received much attention in recent years, and the potential reasons are listed here.

- As stated by Arzén (1999), the event-driven control is closer in nature to the way a human behaves. Many systems are stochastic, meaning that the control (or optimization) actions should react to random events rather than being updated periodically. Indeed, when humans perform manual control, their behaviors are event-driven rather than time-driven. For example, in a stock market, a broker buys or sells stocks only at the "right time" (something just happened); in building energy control, actions are taken only when the networked sensors detect some "meaningful" changes in the environment, such as the temperature or humidity passing a certain level.

- Another reason is the resource utilization. Nowadays, many systems are subject to constraints of resources (e.g. computation, communication and energy). For instance, in a real-time operating system equipped with an embedded controller, the available CPU (central processing unit) time is shared among tasks, such that it appears as if each task is running independently. Occupying the CPU resources to perform control calculations when nothing significant has happened is clearly a waste of computation resources (this is also true for communication resources). This is especially crucial for battery-powered devices, such as wireless sensors (El Gamal et al. 2002, Stark et al. 2002).

- Additionally, the unnecessary changes or updates in systems can cause disturbances (Asad, Yuen & Huang 2017) which, in turn, would affect the system performance. This will inevitably cause frequent changes of the actuator state, leading to unnecessary energy consumption and actuator attrition (Liu et al. 2014).

The aim of the event-driven strategy is to create a better balance between the performance of the controlled processes and other aspects of the system (such as computational load, energy consumption, communication and disturbance, etc.). The key idea of event-driven (or event-based) control is that the control action (or signal transmission) only takes place when the tracking error exceeds a bound or the performance index deviates away sufficiently enough from the desired set point

(Aström 2008). An attractive feature of this idea is that it enables the control actions to follow the stochastic behaviors of system operations, and thus increasing the flexibility of control, communication and optimization (Cassandras 2014). For instance, events can be defined such that no action will be taken when no disturbances are posed on the system or the operation performance is desired, which can avoid unnecessary control actions. Similarly, events can also be defined such that action will be taken promptly when critical changes occur, which can reduce the possible action delay. It is in this way that the event-driven strategy could reduce the resource utilizations (such as energy consumption, communication and computational loads) while still respecting the control performance. Indeed, it has been demonstrated that in some applications, the control performance can be even better with a much lower sampling rate comparing with the time-driven strategy (Astrom, Bernhardsson 2002).

2.2.2 Applications of event-driven strategy

Due to its potential advantages, the applications of the event-driven strategy have been investigated in various aspects of the automatic control area, such as classical feedback control systems (Arzén 1999, Lunze, Lehmann 2010, Anta, Tabuada 2010), distributed systems (Zhong, Cassandras 2010), multi-agent systems (Xu et al. 2017, Li et al. 2013), and wireless network (Araújo et al. 2014).

For example, Heemels, Sandee and Van Den Bosch (2008) developed an event-driven proportional-integral-derivative (PID) controller for a motor. The control values were updated only when a specific error measure exceeds the pre-defined threshold. Results show that 69-76% of computational reduction was achieved, without compromising the control performance. In a motor control problem (Sandee 2006), around 42% of computation time was reduced by adopting the event-driven controller, while still respecting the control performance. In (Henningsson, Johannesson & Cervin 2008), using the proposed sporadic event-based control, better performance was achieved in first-order linear stochastic systems compared with the periodic control regarding both reduced process state variance and reduced control action frequency (40% reduction).

Particularly, for building energy systems, Wu, Jia and Guan (2014) used a finite-horizon discrete-time Markov decision process (MDP) model to control the fan coil unit based on the temperature of the room and the thermal comfort of occupants. Given the difficulty of solving traditional MDP models with the large state space, they proposed an event-based optimization (EBO) approach and applied derivative-based local search to approximately solve the problem. Results showed that the proposed method could converge to a local optimum policy. In (Sun et al. 2013a), the Markov chain of occupant number was developed to optimize the daily HVAC energy cost while satisfying HVAC capacity, comfort and system dynamics. An event-based approach was developed under the Lagrangian relaxation framework so that the decisions were only calculated and executed on an “as needed” basis. The simulation result showed that, the event-based approach is more efficient than time-based stochastic dynamic programming in saving computational load and energy costs, and responded more quickly to changes of occupancy, comfort range, etc. Wu, Jia & Guan (2015) formulated the multi-room HVAC optimal control as an EBO problem, in which decisions were made only when pre-defined events occurred. Results showed that the local-event-based EBO approach performed better than the traditional control methods in terms of energy consumption and thermal comfort. Sun et al. (2015) developed an event-based approach within the Lagrangian relaxation framework to optimize the operation of fan-coil units and fresh-air units in rooms. Seven types of events were defined based the augmented state. Results showed that the event-based approach had a similar energy cost compared with the time-based approaches, while the response is much faster and saving significant computational time.

It is noted that these studies demonstrated the successful application of the event-driven strategy in control and HVAC problems, showing its potential advantages and broad applicability. However, these studies only investigated the air-side systems containing several fan coil units and rooms. The air-side system is only a sub-system of a complete HVAC system. Thus, the feasibility of applying the EDO in a comprehensive HVAC system should be assessed by evaluating the corresponding energy performance, robustness and computational load requirements.

2.3 Design of Event-driven Strategy

The design of an event-driven strategy refers to the specification of the event-triggering condition, also known as the event-triggering mechanism (Heemels, Sandee & Van Den Bosch 2008). The design of the event-triggering condition is critical because it determines when the control settings are updated (Heemels, Sandee & Van Den Bosch 2008). Compared with the time-driven strategy, designing the event-driven strategy often requires more sophisticated techniques (You, Xie 2013). The current practices are summarized as follows.

As a typical example, an event-driven PID controller was designed for a DC-motor in (Heemels, Sandee & Van Den Bosch 2008). The control values were updated only when the tracking error exceeded a threshold. Results showed a satisfactory performance was achieved by the event-driven PID controller together with a great computation reduction, comparing with the time-driven PID controller. However, the thresholds were selected as 0.0005 rad/s and 0.007 rad/s arbitrarily. No explanations were given. Li and Shi (2014) studied the event-triggered model predictive control (MPC) for continuous-time nonlinear systems. The event is triggered as long as the error between the system state and its optimal prediction exceeds a triggering level, which was defined as a function of several parameters. In (Zhang, Feng 2014), an observer-based event-driven controller was developed for continuous-time linear systems. The event was triggered when error singles exceeded the threshold. The numerical example showed that the performance of the event-driven controller is nearly same with the continuous controller. However, the threshold was specified by authors without further explanation. In (Wu, Jia & Guan 2016), the multi-room HVAC optimal control was formulated as an EBO problem, in which decisions (i.e. ON/OFF of the fan coil units) are made only when pre-defined events occur. As the number of global events grows exponentially with the system scale, the local-event-based approach was adopted to simplify the problem. In two numerical examples, the results showed that the local-event-based EBO approach could obtain a near-optimal solution or a better performance, comparing with the threshold-based control, hysteresis control and predictive control methods. It was found that the definitions of the events determined the performance of the EBO approach. However, the event was only defined on the room temperature.

It is worth noting that the designs of the event-driven strategies in the above studies are relatively simple. Besides, the event selections are based on domain experience only. Many of the case systems uses simplified systems in which only a single event was considered. Additionally, the event thresholds were specified in simple ways, sometimes arbitrarily. The selections of the event and event threshold do not involve any optimization processes. Therefore, a systematic design approach for event-driven optimization is required.

2.4 Identify Critical Events in HVAC Systems

To identify the important state transitions corresponding to the optimization performance, prior knowledge on optimal control or data mining techniques are useful resources, which are summarized in the next two sections.

2.4.1 Identify critical events using prior-knowledge

In HVAC applications, the knowledge-based techniques for optimal control have been developed under different names, like model-free supervisory control method (Wang, Ma 2008), heuristic-based control method (Cai 2015) or near-optimal control (ASHRAE 2015). The knowledge-based optimal control method does not involve numerical models (either white-box, black-box or grey-box models). Instead, expertise, engineering analyses or learning techniques are utilized to derive the optimal control rules.

For instance, the expert system (Hordeski 2001) can mimic the human decision-making process for a given working condition, where the decision rules are derived from the knowledge base formulated by the domain experts. Regarding the heuristic-based control method, a near-optimal control method for cooling towers was developed by (Braun, Diderrich 1990), in which and a simple linear control law was formulated since they found that the balancing point for the cooling tower airflow has a linear relationship with part-load-ratio (PLR). In a variable-air-volume (VAV) system, to optimize the static pressure, a heuristic strategy has been proposed (Wang, Burnett 1998), which tries to minimize the resistance inside the duct system for reducing the supply fan energy consumption. The heuristic strategy controls the static pressure set point so that at least one damper is nearly 100% opening at all the time. Besides, in Chapter 42 of the ASHRAE handbook-HVAC applications (ASHRAE 2015), dozens of near-optimal control strategies are presented, including strategies for condenser water flow distribution, pump sequencing, chilled-water reset, chiller sequencing, cooling tower airflow reset and tower fan sequencing. Applications of knowledge-based optimal control strategies can also be found in other building energy systems, such as ice storage (Drees, Braun 1996), shading system (van Moeseke, Bruyère & De Herde 2007), direct-expansion air-conditioning systems (Cai, Braun 2015).

The prior knowledge of optimal control rules in HVAC systems can be possibly utilized for finding critical state transitions, and events can be abstracted accordingly. While many knowledge-based optimal control rules already exist for specific types of components (Davidsson, Boman 2005), the investigation of an entire HVAC system has rarely been discussed. Thus, this thesis will explore the feasibility of applying the knowledge-based method for the EDO in an entire HVAC system.

2.4.2 Identify critical events using data mining of operational data

The building operational data, including temperature, humidity, flow rate, pressure, valve/damper opening and equipment operating status, contains valuable information about the actual building operational performance and patterns. In modern BASs, operational data are collected at a very short time interval (e.g. minutes or seconds). It enables a large amount of building operational data to be available for analyses. Data mining is a powerful technique that can explore and discover meaning knowledge and patterns inside the data set. For building-related data, many successful application have been done using the data mining techniques, such as finding patterns, associations, or relationships (Miller, Nagy & Schlueter 2015), building prediction models (Tang, Kusiak & Wei 2014, Yu et al. 2010), diagnostics (Xiao, Fan 2014) and tuning controllers (Hussain et al. 2014).

For example, both infrequent and frequent diurnal patterns were found using the *DayFilter* proposed by Miller, Nagy and Schlueter (2015). The infrequent operational pattern was found to be consistent with the cooling system faults. The frequent operational patterns can benefit the efficient scheduling of occupancy, lighting and plug load. In (Fan et al. 2015), daily power consumption patterns, sub-system operational patterns and temporal associations between sub-systems were identified using the proposed data mining method. However, the authors did not explore the use of the discovered knowledge in for optimal control. Li et al. (2017) used data mining techniques to analyze the energy consumptions of variable refrigerant flow systems. Three distinct power consumption patterns, undercharge fault, low and higher part load ratio conditions, were identified by clustering analysis. Besides, energy consumption rules were extracted by association rule mining. It was found that the energy saving potentials could be estimated by making comparisons between energy patterns and rules in a top-down way.

It is observed that most previous attempts have been limited to identification of critical factors for energy performance, system faults or operational patterns. For building optimal control, a gap remains between the knowledge base and the real-world applications of the this knowledge. Therefore, this thesis will develop a method to utilize the knowledge acquired from data mining for HVAC optimal control.

2.5 Discussions

Based on the extensive literature reviews given above, findings are summarized as follows.

(1) HVAC optimal control

HVAC systems are major energy consumers in buildings, and RTO is a powerful tool for improving the energy efficiency of HVAC systems. However, with the increasing complexity of HVAC systems, the traditional TDO strategy of RTO is no longer efficient in handling the optimal control problems. Firstly, the TDO can easily postpone the optimization or perform unnecessary optimizations, which is adverse to energy or computational efficiency of system operations. Secondly, it is difficult to select a suitable optimization frequency in practices since the relationship between the optimization frequency and actual optimization performance is unclear (due to the lack of studies).

(2) Event-driven strategy

The event-driven strategy has the potential capability to reduce the resource utilization (e.g. communication, computation and energy consumption) while still ensuring the performance of control or optimization. It has been successfully applied in different areas, including classical feedback control systems, distributed systems, multi-agent systems, wireless network and HVAC systems as well, which shows its broad potential applicability. However, in HVAC systems, only air-side systems have been investigated, and only simple events have been considered. There are no comprehensive studies of the entire HVAC systems. Furthermore, there is no systematic framework to facilitate the application of EDO in HVAC systems. Therefore, an EDO framework will be developed herein.

(3) Design of event-driven strategy

Compared with time-driven strategies, the design of event-driven strategies is much more sophisticated. Current practices only study some relatively simple systems; while it is definitely fine as the main purpose is to demonstrate the event-driven strategy. Moreover, most of these studies has considered only a single event based on the domain experience, which may not be sufficient when a complex system is considered. In addition, the event threshold is often specified arbitrarily, based either on expertise or inference using the system model. However, when no expertise or explicit system model is available, as is likely in real systems, there is no effective solution. There is no optimization process for the selections of events or event thresholds. Thus, a method should be developed for the design of event-driven strategy.

(4) Find critical events

In HVAC, prior-knowledge or operational data can be used to identify the critical state transitions relating to the optimization performance. For knowledge-based methods, a large number of optimal control rules have been developed. However, these rules are mostly specific to certain equipment or sub-systems. The effectiveness of these optimal control rules at the entire-system level remains to be validated. As for the data-based method, the continually increasing collection of BAS data has enabled data mining techniques to be successfully used to find patterns, associations, or relationships, build prediction models, perform diagnostics and tune controllers. However, there is no systematic way to utilize data mining for HVAC optimal control. The EDO provides a good platform for such applications. Thus, it is worthwhile to develop a way to facilitate the use of data mining in the EDO.

2.6 Summary

This chapter synthesizes the literature on HVAC optimal control, event-driven strategy in control and optimization, design of the event-driven strategy and two possible ways to identify important events in HVAC systems. The literature provides the necessary information for the presentation of the thesis and lays the foundation for the development of the methods.

Firstly, the traditional TDO of HVAC optimal control is reviewed. As the operation of HVAC systems usually confronts many stochastic state transitions, such as weather condition changes, load variations and occupancy changes, “time” may not an efficient driver for the HVAC optimal control. Specifically, the current TDO strategy in handling the optimization triggering is still inefficient since it may lead to unfavorable optimization actions (e.g. delayed or unnecessary actions). Besides, the selection of the optimization frequency in TDO is always difficult in practices due to the lack of studies. The limitations associated with the TDO call for the reformulation of the optimal control strategy, which leads us to the event-driven strategy. Then, the basic idea and applications of event-driven strategy are introduced. A large number of previous attempts have demonstrated the potential superiorities of the event-driven strategy by successful applications in different problems. Thirdly, the current practices of the design of event-driven strategy are discussed. Designing the EDO is more complicated because there are several asynchronous event-triggering conditions to be specified. It is found that the current practices of the event-driven strategy design are simple, based on domain knowledge and have no optimization process. Finally, the use of prior knowledge and data mining techniques to identify important state transitions for optimal control in HVAC systems is presented.

CHAPTER 3. EVENT-DRIVEN OPTIMIZATION FRAMEWORK AND SIMULATION PLATFORM

In this chapter, the EDO framework is established, and the simulation platform used to evaluate the EDO is introduced. First, the basic difference between the EDO mechanism and the TDO mechanism is presented in Section 3.1. The basic EDO strategy is also illustrated by the {event, policy, action} structure. Next, the optimal control diagram of EDO is illustrated in Section 3.2. Then, Section 3.3 presents the fundamental terms associated with the "event", including event attributes, event types, mathematical representations and event identification. Section 3.4 introduces the co-simulation platform for optimization performance evaluation. Finally, the formulation of the optimization problem and performance indices are presented in Section 3.5 and 3.6 respectively.

3.1 Event-driven Optimization Scheme and Strategy

The difference between the EDO and the TDO is illustrated in Figure 3. 1, where five decision epochs (DEs) are assumed to be allocated for a certain period and "*t*" is the time interval between two adjacent DEs. Each DE consumes the same amount of computational resources to complete a search for the control optimization. Compared with the TDO, which is a deterministic allocation scheme (the DE location is fixed), the EDO has the adaptability to the changing environment as the DE location can be altered accordingly. Therefore, the EDO can be regarded as a smarter allocation scheme. In the case study part, we will demonstrate that this EDO could achieve an equivalent performance with the same or even less amount of the computing budget.

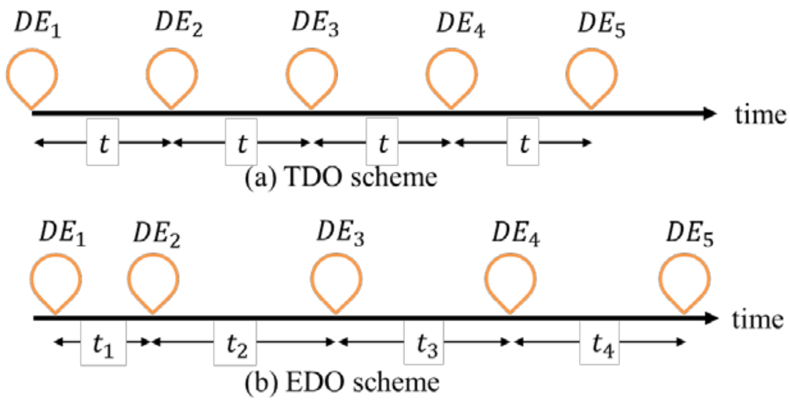

Figure 3. 1 Comparison between the TDO and the EDO schemes (DE=decision epoch)

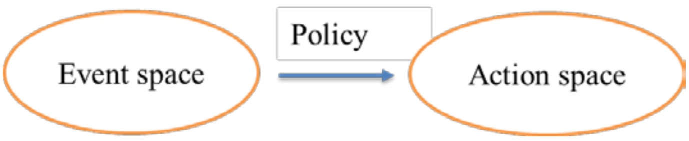

Figure 3. 2 The {event, policy, action} structure

The underlying optimal control strategy is realized by a so-called {event, policy, action} structure (as shown in Figure 3. 2). This {event, policy, action} structure is the core of the proposed EDO scheme, and the basic event-driven strategy is listed as follows:

- if no event occurs, no action will be taken;
- if certain events occur, actions will be taken accordingly based on the policy for achieving certain objectives (e.g. energy efficiency).

Thus, the event determines when to trigger the control optimization. After the optimization is triggered, the policy determines which action to be taken based on the observed event, which is a mapping from the event space (the set of events) to the action space (the set of actions). As the optimization triggering and action selection all depend on the event, it is called the "event-driven optimization".

3.2 Optimal Control Diagram

Figure 3. 3 shows the detailed control diagram of the EDO. When real-time operational data is fed to the control system, events will be identified from a pre-defined event space. When certain events occurred, actions will be taken from the pre-defined action space based on the policy. Therefore, the proposed EDO takes a continuous feedback and implements the optimal control in an event-driven manner.

The optimization algorithms that are developed for time-driven optimization methods can also be used to optimize decision variables in this EDO framework.

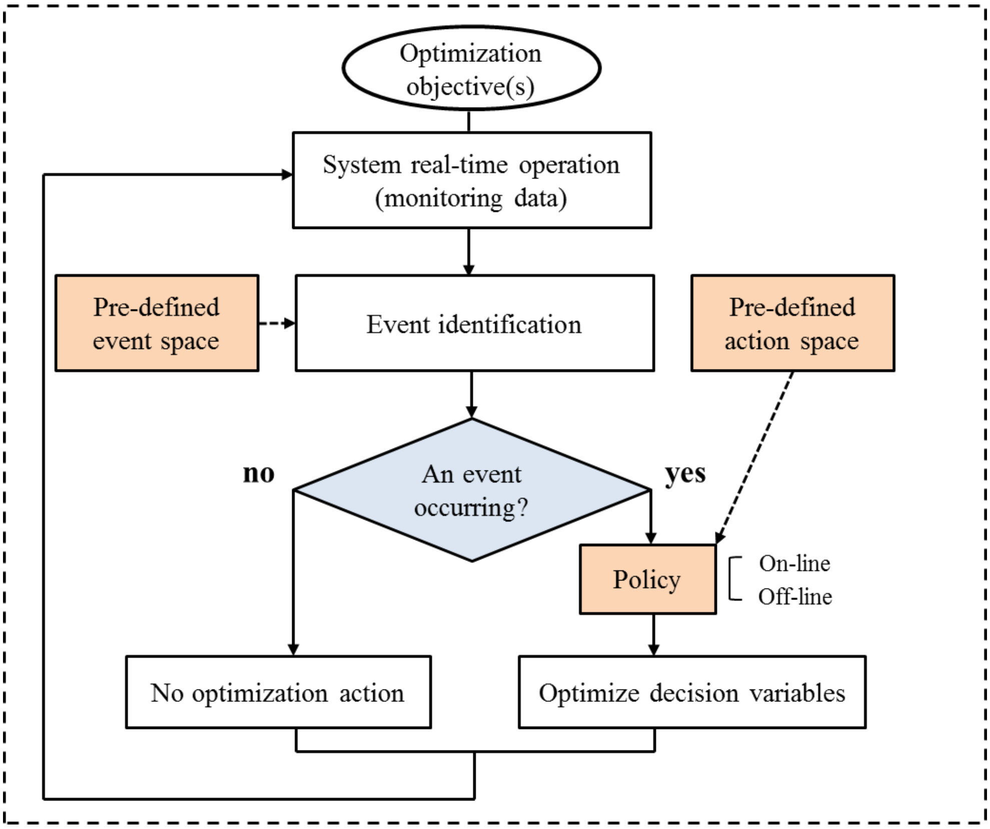

Figure 3. 3 The optimal control diagram of EDO

Please note that there are two different methods to update the policy: on-line version and off-line version. “Off-line” means that the policy is pre-defined and fixed for the next period (e.g. 24 hours) of operation until necessary changes are made; while the on-line version changes the policy according to the real-time condition during the period of operation. The on-line policy updating can be achieved by employing suitable optimization algorithms (e.g. event-based optimization (Jia 2014, Xia, Jia & Cao 2014, Cao et al. 2013)) to search the optimal policy based on the available data.

3.3 Event and Event Identification

3.3.1 What is an event?

An event describes a set of state transitions that physically happen in a system (Cassandras, Lafortune 2009, Xia, Jia & Cao 2014). In daily operation of a complex air-conditioning system, events may come from environment (such as weather changes and solar radiation changes), system itself (such as equipment on/off, equipment faults and operation mode changes) and occupants (such as occupancy changes and occupants' adjustment of thermal comfort related variables). For example, a chiller being switched on can be considered as an "event"; and the cooling load increased by 10% can be considered as an event as well. Hence, "event" can be defined from a discrete state (e.g. the operating status of chillers) or continuous state (e.g. the cooling load), which will be discussed in the Section 3.2.2.

3.3.2 Event attributes and types

Basic event attributes and types are presented here to lay the foundation for event definitions. As shown in Figure 3. 4, a typical event can be represented abstractly by three basic attributes, namely timestamp, descriptive state variable and threshold, each of which has two forms. The timestamp can be a time instant which could represent a transient state transition or a time duration that reflects a continuous state change. For example, a transient state transition could be the indoor temperature increased a certain level (e.g. 25 ºC); a continuous state change could be the relative humidity of indoor air increased by 20% in the past 30 minutes, which represents an accumulated effect. The descriptive state variables can be continuous or discrete, which is determined by the nature of state transitions. For instance, temperature, flow rate and pressure are typical continuous variables in a HVAC system; while the number of devices in operating is a discrete state variable.

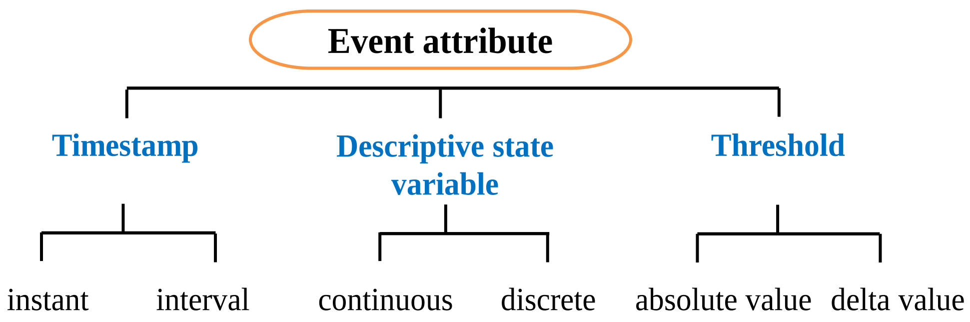

Figure 3. 4 Typical event attributes

Another important attribute is the threshold, which determines the form and quantity of a state transition. Basically, the threshold quantitatively describes “*certain properties*” that the transition depicts, which is a user-defined parameter. It can be an "absolute" or "delta" value (Heemels, Johansson & Tabuada 2015, Hinze, Sachs & Buchmann 2009). The "absolute" mean the descriptive state variable passes a certain level or predefined value, such as the temperature exceeding a set point; while the "delta" refers to the variations of the difference between the current value and its reference or previous value, such as the PLR varying more than 10% compared with the last control updating. Based on different types of timestamp, descriptive state variable and threshold, events can be roughly divided into eight basic event types as shown in Table 3. 1. Different types of events would have slightly different mathematical representations, which are discussed in Section 3.3.3.

Table 3. 1 Basic event types

Attribute / Event type	**Timestamp**	**Descriptive state variable**	**Threshold**
Event type 1	instant	continuous	absolute
Event type 2	instant	continuous	delta
Event type 3	instant	discrete	absolute
Event type 4	instant	discrete	delta
Event type 5	interval	continuous	absolute
Event type 6	interval	continuous	delta
Event type 7	interval	discrete	absolute
Event type 8	interval	discrete	delta

3.3.3 Mathematical representation

Formally, an event is a set of state transitions that happen instantly or continuously in a period of time. The terminology and basic notions that are used in this thesis mainly follow the work in (Cassandras, Lafortune 2009). Let X_{τ_k} be the system state at time τ_k, which is a vector containing a set of state variables that can be used to reflect the system behaviors. When X_{τ_k} contains n state variables, it is denoted as $X_{\tau_k} = \{x_{\tau_k}^1, x_{\tau_k}^2, \dots, x_{\tau_k}^n\}^T$, where x^n is the component of X and is called "state variable", like temperature, humidity or water flow. For x^n, the transition of this state variable is represented as $\{< x_{\tau_i}^n, x_{\tau_j}^n >\}$, which means the transition of x^n from time τ_i to τ_j. Please note that τ_i is the last decision time and τ_j is the current decision time. The decision time interval is normally larger than the sampling time interval.

Suppose $X = \{T,\ RH,\ PLR\}^T$ and define $e_1 := \{< X_{\tau_i}, X_{\tau_j} > \mid\ T_{\tau_i} \leq 25℃, T_{\tau_j} > 25℃\}$, then e_1 is an event describing the temperature increase and passing a certain level (i.e. 25℃), which belongs to the "Event type 1"; define $e_2 := \{< X_{\tau_i}, X_{\tau_j} > \mid\ |PLR_{\tau_j} - PLR_{\tau_i}| \geq \sigma_{PLR\ Change}\}$, e_2 represents a delta type of instant state transition that is "Event type 2". However, this type of events only specifies one state variable transition, and a more complex case is that multiple state variables vary at the same time. For simplicity, this study considers one state variable for event definitions.

All the defined events constitute an event space: $E_{space} = \{e_1, e_2, \dots, e_m\}$, where m is the size of the event space. As not all of "state transition" should be used as "events" to trigger optimization, only those "state transitions" which could cause a significant influence on concerned objectives (such as energy efficiency) will be defined as events. The design issues concerning the event space will be discussed in Chapter 4.

3.3.4 Event identification

When events are properly defined, an important process in real implementation is "event identification" which is a matching process (as shown in Figure 3. 5) between the events defined and the state transitions observed during the system operation (also termed as the observable event by (Cao 2007)). After an identification process, a value ("0" or "1") is attached to the event to indicate whether or not this event occurs, where "1" means the event occurs and "0" means no event is identified. More specifically, an event represents an abstracted description of the state transitions, while an observable event contains observed values of state variables and timestamp (represents the time of occurrence). It should be mentioned that the observable events are mutually exclusive since at least the timestamps are different (although they could belong to the same “event”).

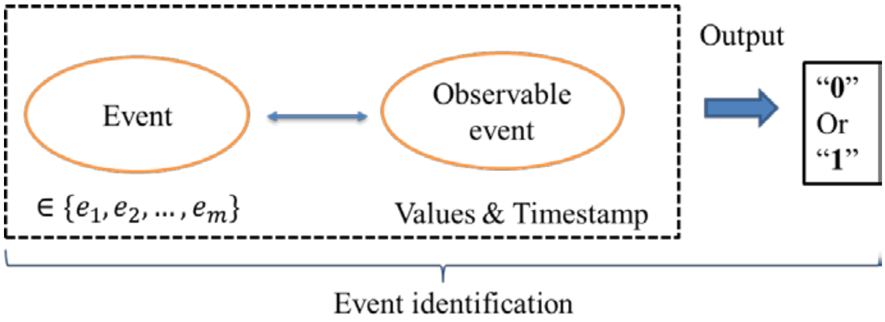

Figure 3. 5 Event identification process

3.4 Simulation Platform

3.4.1 Co-simulation platform between TRNSYS and MATLAB

The simulation platform was constructed by using the co-simulation between TRNSYS and MATLAB (see Figure 3. 6). The virtual HVAC system was established in TRNSYS, which was used to produce the online operation data. The optimization strategy and algorithm were coded and realized in a separate MATLAB module. During every optimization process, the operational data was sent to the “optimizer” at first; then, the “optimizer” optimized the decision variables through minimizing the designed objective (such as total power requirement) based on the current operational condition. Certain performance predictors will be used to evaluate the values of the cost function of different feasible settings. Finally, identified optimal settings of the decision variables were sent to the virtual HVAC system to supervise its operation, and the process will continue.

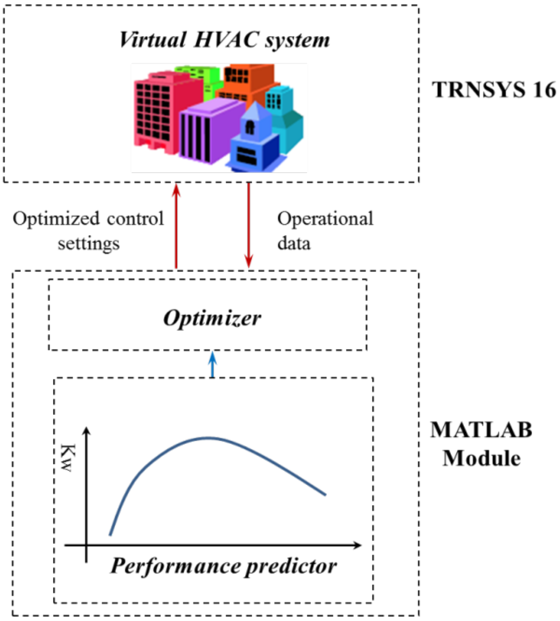

Figure 3. 6 Structure of co-simulation between TRNSYS and MATLAB

3.4.2 System description

The air conditioning system under study is an all-electric system which does not contain significant thermal storage. The system is established according to a supertall office building in Hong Kong. Circulation loops of cooling water, chilled water primary side and secondary side are simulated in details, while the air distribution system is simplified by one zone (Figure 3. 7) to simplify the optimization problem. The central system utilizes a typical two-level control hierarchy (Braun 2014) to realize the expected functions. Since this is a large-scale system containing multiple chillers, pumps, cooling towers, fans, AHUs and heat exchangers, apart from the optimization function used at the supervisory level, several fundamental control laws are embedded into the system to ensure basic functionalities. The supervisory level handles optimization problems with respect to certain cost functions. The local control loops deal with basic functions like set-point tracking (usually using PI control) and equipment sequencing control, and details will be presented in Section 3.4.3. Please also be noted that this is a very classical air conditioning system currently used in many commercial buildings, and thus quite representative. We will use it as a case system to demonstrate our EDO strategy and the design approach in the upcoming sections.

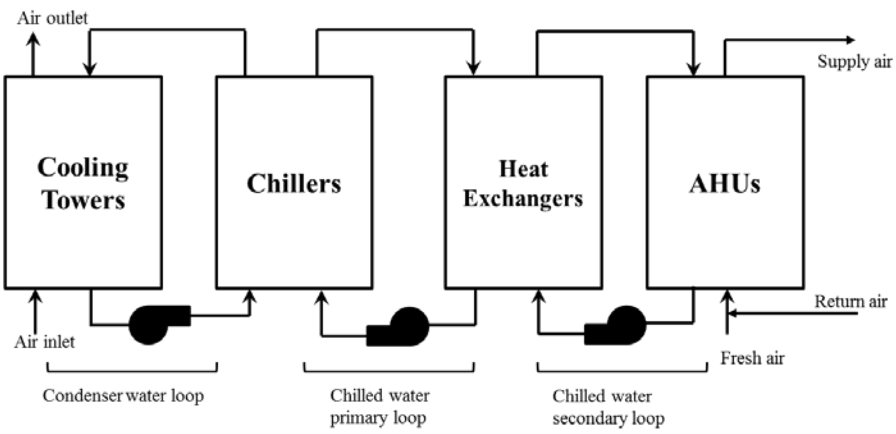

Figure 3. 7 Schematic of the HVAC system

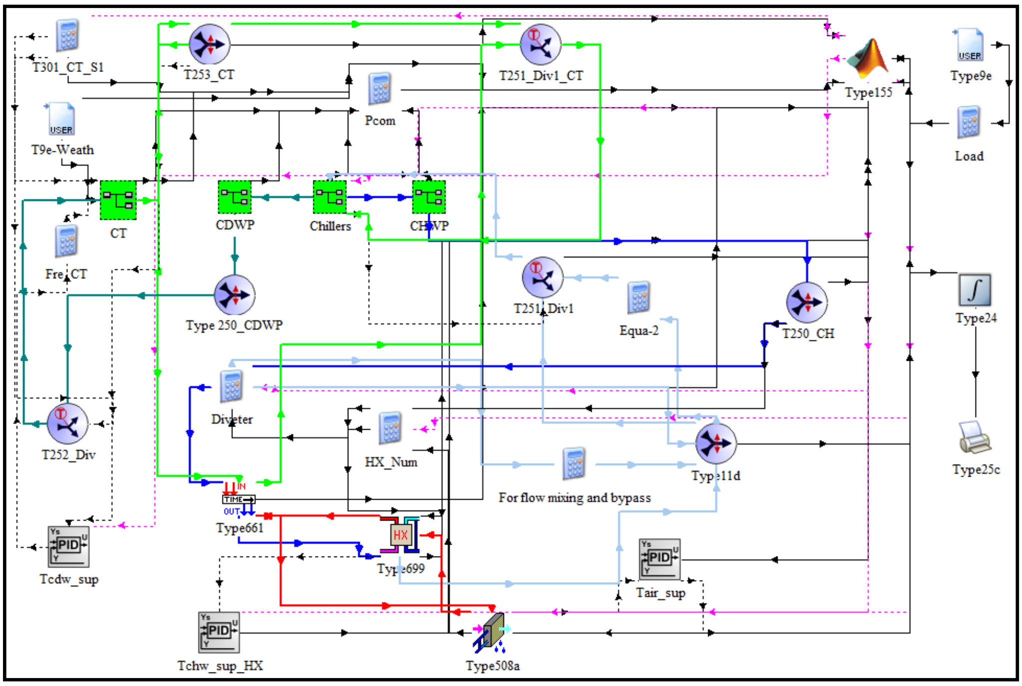

Figure 3. 8 TRNSYS model (screenshot)

Based on this physical system, the TRNSYS model was built based on validated models (Ma 2008, Wang 1998). Six water-cooled centrifugal chillers were employed in the chiller plant, each of which has the capacity of 7230 kW. The rated water flow rates of the pumps for chilled water and cooling water circulation are 345 l/s and 410 l/s. Eleven cooling towers were used and the rated water flow is 250 l/s. Details of main equipment are shown in Table 3. 2, and the model codes are shown in the Appendix A. The screenshot of the established TRNSYS model is shown in Figure 3. 8.

Table 3. 2 Details of main components

Equipment	Number	$M_{w,ev}$ (L/s)	$M_{w,cd}$ (L/s)	Capacity (kW)	Rated power (kW)
Water-cooled Chiller	6	345.0	410.0	7230	1346
		M_w (L/s)	M_a (m^3/s)	Q_{rej} (kW)	
Cooling tower - type A	6	250.0	157.2	5234	152
Cooling tower - type B	5	194.0	127.0	4061	120
		M_w (L/s)	Head (m)	Efficiency (%)	
Condenser water pump	6	410.0	41.6	83.6	202
Primary chilled-water pump	6	345	31.6	84.5	126
Secondary chilled-water pump	6	345	30.3	84.2	122

3.4.3 Fundamental local-loop controls

To guarantee a proper system operation, several fundamental controls are necessary (Sun et al. 2013b) and are briefly explained as follows.

Sequencing control of chillers: this is to stage on or off chillers based on a given load condition. Here, a total-cooling-load-based sequencing control method is adopted.

This method estimates the cooling load Q_{ch} by Eqn. (3. 1) and compares Q_{ch} with predefined thresholds $\mathrm{Q}_{\mathrm{z}}^{\mathrm{on/off}}$ to decide to switch on or off chillers. Normally, a dead band should be adopted to avoid frequent switch triggering when the load fluctuates within a narrow interval (Sun, Wang & Xiao 2013). The switch-on/off thresholds are calculated by Eqns. (3. 2) and (3. 3); a chiller and its corresponding pump(s) will be staged on when the instantaneous cooling load is greater than the threshold value for a certain time period; a chiller and its corresponding pump(s) will be staged off when the instantaneous cooling load is lower than the threshold value for a certain time period (Liao et al. 2014).

$$Q_{ch} = c_p m_w \left(T_{chw,\,rtn} - T_{chw,\,\sup}\right) \tag{3. 1}$$

$$Q_z^{on} = z \times Q_{rated} \times \left(1 + dead\_band\right) \tag{3. 2}$$

$$Q_z^{off} = \left(z - 1\right) \times Q_{rated} \times \left(1 - dead\_band\right) \tag{3. 3}$$

where c_p is the water specify heat; m_w is the mass flow rate of water; $T_{chw,rtn}$ and $T_{chw,sup}$ are the chilled water return temperature and chilled water supply temperature; Q_z^{on} is the switch-on threshold; Q_z^{off} is the switch-off threshold; z is the number of chillers in operation; Q_{rated} is the nominal cooling capacity of one chiller (here, each chiller has the same rated cooling capacity); and *dead_band* is the dead band which is a user-defined value between 0 and 1.

Sequencing control of cooling towers: this is to determine the on or off switch of towers according to the heat amount that needs to be rejected. In practice, the operating cooling towers number N_{ct} is always coupled with the operating chillers as shown in Eqn. (3. 4), where k is a coefficient that normally depends on the chiller plant configuration.

$$N_{ct} = kN_{ch} \tag{3. 4}$$

Controls of critical temperatures: Four critical temperatures are always under feedback control, including supply cooling water temperature, supply chilled water from chillers, supply chilled water from the heat exchangers and supply air temperature. The supply air temperature is maintained through modulating the water flow rate inside the AHUs. The supply cooling water temperature is controlled by changing the cooling tower fan frequency; the supply chilled water temperature from the chiller(s) is maintained by changing the refrigerant flow rate; the supply chilled water temperature from the heat exchanger(s) is controlled through modulating the water pump speed.

In these local control loops, several PI/PID controllers were used to track the set-points. A PID controller continuously computes an error signal *e(t)* as the difference between a desired value and a measured value variable, and applies a correction based on proportional, integral, and derivative terms; a PI controller uses the same principle but only has proportional and integral terms. The PI controller with the parameters P = -0.95 and I = 35s was used to control the fan speed of cooling towers so as to track the CWS temperature set-points. For the SCHW temperature from heat exchangers, the PID controller with the parameters P = -0.9, I = 10s and D = 5s was adopted to control the pump speed. An additional PI controller with the parameters P = -0.3 and I = 2s was employed to track SA temperature. Controller parameters were kept constant under different optimization mechanisms and the trial-and-error method was used in controller tuning.

3.5 Optimization Problem Formulation

The optimization problem is formulated in this section, which specifies the objective function, operational constrains and solution algorithm. This problem formulation was used in all the case studies of this thesis.

3.5.1 Problem formulation

Optimal control in HVAC area basically includes static and dynamic optimizations, which depends on the problem type and are significantly different (Wang, Ma 2008). Static optimization solves the optimization problem at a given time instant, while dynamic optimization considers the future system state and addresses the problem over a time period (ASHRAE 2015). Regarding our case building, we consider the static optimization (without the prediction for system future states) and this simplification is also adopted by ASHRAE handbook (ASHRAE 2015) when the system does not contain significant thermal storage. As a result, cost optimization will be equivalent to minimization of power at each time instance.

$$P_{sys,tot} = P_{ch,tot} + P_{ct,tot} + P_{pump,tot} + P_{fan,tot} = f\left(T_{scw}, T_{schw,prm}, T_{schw,\sec}, T_{sa}, U\right) \tag{3. 5}$$

$$\left(T^*_{scw}, T^*_{schw,prm}, T^*_{schw,\sec}, T^*_{sa}\right) = \underset{T_{scw}, T_{schw,prm}, T_{schw,\sec}, T_{sa}}{\arg\min} P_{sys,tot} \tag{3. 6}$$

Therefore, the static form of the cost function (Eqn. (3. 5)) is represented as the sum of the total power requirement of chillers, cooling towers, pumps and fans, which can be written as a function of controlled and uncontrolled variables. The controlled variables are four temperature set-points of "scw", "$schw$" at primary and secondary sides, and "sa", which are continuous. The uncontrolled variables are written as "U"

which is a vector containing parameters like outdoor dry-bulb temperature. In summary, this is a model-based static optimal control problem for the typical all-electric HVAC system without significant thermal storage.

Different set-point combinations can satisfy a given cooling output requirement, while only one set of combination could lead to the minimal energy consumption. Trade-offs are made by solving Eqn.(3. 6), which outputs the optimal settings. Besides, the controlled variables may have lower and upper limits in real operations, which will be treated as operational constraints (shown in Eqn. (3. 7)-(3. 10)). The temperature set-point change of each updating is also limited (Eqn. (3. 11)) to avoid potential instability issues (Asad, Yuen & Huang 2016, Asad, Yuen & Huang 2017, Sun et al. 2013b). Meanwhile, the minimal sampling time interval (Eqn. (3. 12)) for resetting is applied since the ASHRAE handbook (ASHRAE 2015) recommends that it should be greater than the settling time for the control loops. The values of these constraints are presented in Section 4.4.2.

$$T_{scw,lower} \leq T_{scw} \leq T_{scw,upper} \tag{3. 7}$$

$$T_{schw,prm,lower} \leq T_{schw,prm} \leq T_{schw,prm,upper} \tag{3. 8}$$

$$T_{schw,\sec,lower} \leq T_{schw,\sec} \leq T_{schw,\sec,upper} \tag{3. 9}$$

$$T_{sa,lower} \le T_{sa} \le T_{sa,upper} \tag{3. 10}$$

$$\left|T_{k+1} - T_k\right| \le \Delta T_{step\ change} \tag{3. 11}$$

$$\tau_{i+1} - \tau_i \ge ST_{\min} \tag{3. 12}$$

<u>3.5.2 Operational constrains and solution algorithm</u>

Table 3. 3 shows the values of operational constraints (presented in Eqn. (3. 7(3. 10)) which are based on the previous study (Wang et al. 2016). All the three cases have the same setting except the set-point for the supply cooling water temperature.

For the search ranges of supply cooling water temperature, further explanations are given as follow. The lower bound is calculated by applying a constant approach (i.e. 2°C). “Approach” in cooling tower operation means the temperature difference between condenser water supply and the ambient wet-bulb temperature. Normally, 2.8°C is the minimal approach that cooling tower manufacturers will guarantee (Cooling Technology Institute 2016). Here, we use approach of 2°C to set the lower bound. Since 4°C is used as the approach in common cooling tower design (ASHRAE 2015), we use a higher approach of 9°C to set the upper bound in order to include all the possible temperature. Please refer to Table 3. 4 for the detailed calculation.

Table 3. 3 Operational constraints of the case system

Name	Value		
	Spring scenario	Summer scenario	Autumn scenario
$[T_{scw,lower}, T_{scw,upper}]$	[20, 28] ˚C	[28, 35] ˚C	[24, 30] ˚C
$[T_{schw,prm,lower}, T_{schw,prm,upper}]$		[5, 8] ˚C	
$[T_{schw,\sec,lower}, T_{schw,\sec,upper}]$		[6.5, 11.5] ˚C	
$[T_{sa,lower}, T_{sa,upper}]$		[12, 18] ˚C	
$\Delta T_{step\ change}$		0.5 ˚C	
$ST_{\min}$		5 minutes	

Table 3. 4 Search ranges for the set-point of cooling water supply temperature

Case	$T_{wb,mean}$(°C)	$T_{wb,mean}+2$(°C)	$T_{wb,mean}+9$(°C)	Search range
Spring	18.40	20.40	27.40	[20, 28]
Summer	26.36	28.36	35.36	[28, 35]
Autumn	21.33	23.33	30.33	[24, 30]

(Search range uses integers for the sake of convenience.)

The solution algorithm is not the main focus of this thesis, and an exhaustive search is used because of its simplicity and suitability. Here, the search space is already optimized according to the real operation. We restrict the set-point change every time and use a coarse step change (i.e. 0.1 °C) for searching, which reduces the search space to the size of 10^4. If we refine the step change to 0.01 °C, the search space will be extremely large (go to 500^4), which would make the search process to be time-consuming. Since “0.1 °C” is accurate enough in commercial HVAC systems and was also adopted by several previous studies (Wang et al. 2016, Ma, Wang 2009), it is used in this thesis.

3.6 Performance Indices

In order to evaluate the performance of different optimal control strategies, performance indices were defined based on the objective. As the main goal of using EDO is to achieve a better balance between energy and computational performances, the energy saving and computation saving percentages (*ES%* and *CS%*) are considered as the performance indices which are shown in Eqn. (3. 13) and (3. 14).

$$ES\% = 100\% \times (EC - EC_{BC}) / EC_{BC} \tag{3. 13}$$

$$CS\% = 100\% \times (CC - CC_{BC}) / CC_{BC} \tag{3. 14}$$

To select a suitable optimal control strategy, the user needs to consider both energy and computational performances, and make a trade-off. Basically, this is a multi-criterion decision-making problem, and many existing methods can be used (Fülöp 2005). Here, a simple multi-attribute rating technique (SMART) is used to compute the “performance score” since it can complete the searching through one comparison based on a linear additive model (Huang, Huang & Wang 2015). As shown in Eqn. (3. 15), performance score is defined as a weighted sum of scores of energy consumption and computation consumption.

$$PS = a \times score_{EC} + b \times score_{CC} \tag{3. 15}$$

$$score_{EC} = ES\%_{ei} / ES\%_{BC} \tag{3. 16}$$

$$score_{CC} = CS\%_{ei} \tag{3. 17}$$

Where, a and b are the weighting factors, ei is the event i, BC is the benchmark case for evaluations. The user can choose the weighting factors based on their preferences, experience or requirements. In this thesis, "$a = 1$" and "$b = 0.5$" were used as weighting factors in Eqn. (3. 15) simply because our priority was given to the energy performance while computation efficiency was considered as less important.

3.7 Summary

To facilitate the event-driven strategy for RTO in HVAC systems, this chapter established an EDO framework which contains the control diagram and fundamental terms of events. Events are formally defined and event attributes are synthesized for HVAC systems. In order to evaluate the optimization performance of the EDO strategy, a co-simulation platform between TRNSYS and MATLAB is constructed. The virtual HVAC system is built based on a supertall commercial building in Hong Kong. The optimization problem is also formulated accordingly. Besides, performance indices are defined for evaluating the performances of optimal control strategies. In the following studies, the optimal control strategies will be tested and evaluated based on this simulation platform according to those defined performance indices.

CHAPTER 4. METHODOLOGY OF EVENT SPACE ESTABLISHMENT

Section 4.1 briefly discusses the general procedure for designing the EDO strategy (presented in Section 3.1). The main tasks are to design the event space, policy and action space to form the {event, policy, action} structure. Special attention is paid to the event space since "event" determines the optimization performance of the EDO. Section 4.2 discusses the event space establishment, where the general decision-making process, overall methodology and the basic event selection criterion are introduced.

4.1 EDO Design Procedure

The general design procedure for EDO is illustrated in Figure 4. 1 and briefly described as bellows.

Step 1: Specify optimization objectives. Typical optimization objectives in HVAC systems are minimization of the operational cost, maximization of the energy efficiency and thermal comfort. Events and actions should be defined accordingly so as to achieve the goal.

Step 2: Event space establishment refers to the task of finding suitable events to form the event space. Since an event is a set of state transitions, important state variables should be identified regarding the optimization objective. As there could be numerous possible events in a complex system (such as the HVAC system), effective methods should be developed to identify the most important one.

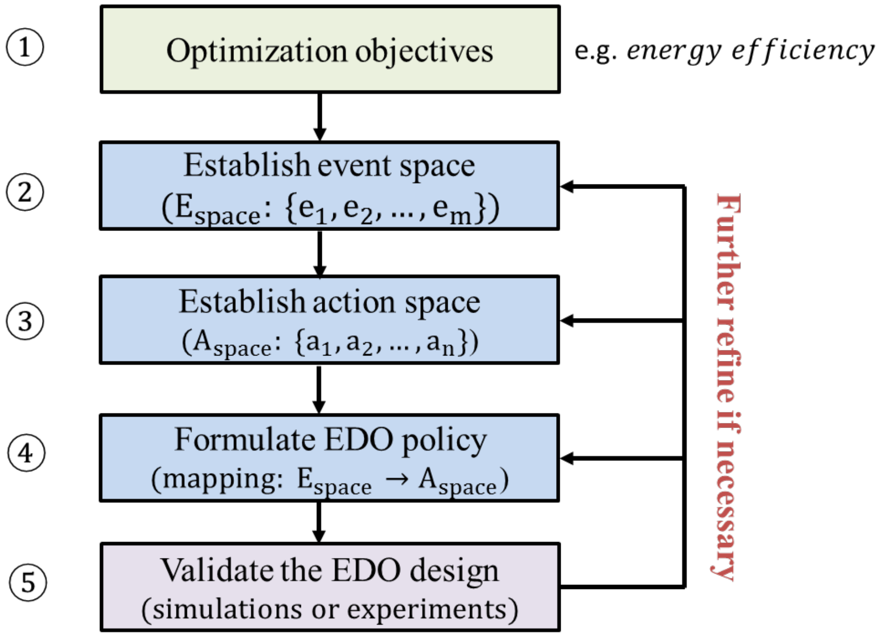

Figure 4. 1 EDO design procedure

Step 3: Establish action space. Here the "action" refers to optimization action only. Action implies which system(s) and which decision variable(s) to be optimized. For instance, optimizing the supply air temperature of an air-handling unit can be used as an "action".

Step 4: Formulate EDO policy. EDO policy formulation refers to the establishment of the mapping between the events (in the event space, E_{space}) and the actions (in the action space, A_{space}), which is used to supervise the system on what to do when an event occurs.

Step 5: Validate the EDO design. This is to test and validate the EDO design, where simulations or experiments can be used. Performance indices (Section 3.6) like energy saving or computation saving can be used to evaluate the corresponding performance using the formulated EDO policy. This step is also necessary for further refinement of the established event space, action space, and policies, when several design options are available.

It should be noted that the above steps constitute a complete design procedure; while this thesis mainly focuses on the establishment of the event space. The reason is that events dominate the proposed EDO strategy since "event" determines when to trigger the optimization and which action to be taken. To simplify the EDO design problem, the default "action" is used, i.e. optimizing all the decision variables when events happen, which is also used in traditional TDO schemes. Since there is only one action, the mapping becomes simple, i.e. each event in the E_{space} will trigger the default action.

4.2 Event Space Establishment

The main task of the event space establishment is to find suitable events to form the event space. Efforts should be made to appropriately identify the possible state transitions and define the events, which could be a gargantuan job in many complex systems (such as the HVAC system). Thus, this section develops the general process, methodology and criterion for the event space establishment.

4.2.1 General process of event space establishment

Events may originate from different sources, and there are three main event sources (Figure 4. 2) in HVAC systems. To facilitate the task of event space establishment, three main event sources are briefly discussed as follows.

- Environment: refers to both the indoor and outdoor environmental changes, such as changes in the temperature and humidity of a room or outdoor weather.

- System: refers to changes in both the hardware and software sides of a system, such as equipment faults (hardware) and operation mode switching (software). Besides, system operational constraints can also be regarded (and easily formulated) as events so as to utilize the EDO framework. For instance, the water level in a tank cannot be above or below a certain level, and the time interval between two actions cannot be lower than a certain threshold, etc.

- Human: refers to the changes associated with human, such as occupancy changes, occupants' access to controls (e.g. to change a room temperature set-point) and building operators' manipulations.

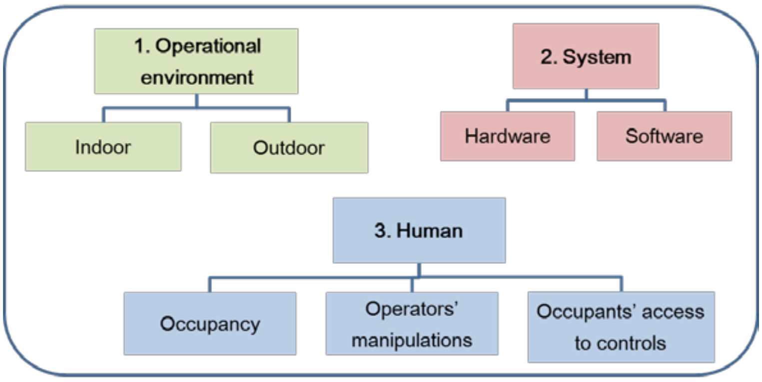

Figure 4. 2 Event sources in HVAC systems

By definition, an event is a set of state transitions (see Section 3.3.1). While state transitions are numerous in a system, only those "state transitions" which could cause a significant influence on the specific objectives (such as energy efficiency) will be selected and defined as events. Here, “deciding which state transition should be used and defined as an event” is basically a decision-making problem. The general flow chart of solving this decision-making problem is summarized in Figure 4. 3.

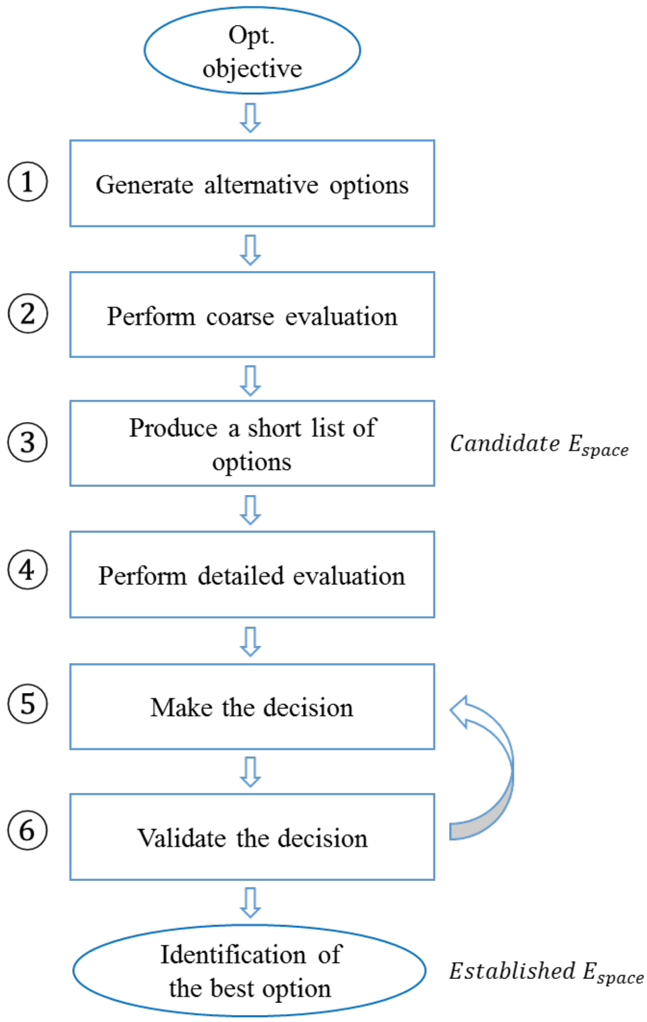

Figure 4. 3 General flow chart of event space establishment

(“opt. = “optimization”)

Step 1 is to produce as many alternative options as possible based on the decision criteria without evaluating them. A decision criterion is usually derived from the optimization objective(s), which could be the maximization (such as efficiency) or minimization (such as cost).

Step 2 is to perform the coarse evaluation for the possible alternative options. For instance, practical and impractical solutions can be differentiated.

Step 3 is to screen out obviously bad choices since too many options in the list will be too confusing. As a result, the options (or events) in the list form a candidate event space that is subject to further refinement.

Step 4 is to evaluate the option regarding each key decision criteria.

Step 5 is to make the decision on which event(s) to be selected based on the detailed evaluation.

Step 6 is to implement and validate the decision, where experiments or simulation can be used. The step 5 and step 6 will be iterated until the performance is satisfactory.

4.2.2 Overview of the methodology for event space establishment

The overall methodology for event space establishment is developed based on Figure 4. 3, which is divided into three steps (Figure 4. 4). Step 1 is state transition identification that is used to identify possible critical state transitions. Step 2 is to define the candidate event space. Firstly, based on the identified state transition space, event attributes will be extracted and expressed in mathematical forms. Then, event thresholds will be specified for continuous state variables. Step 3 is to optimize the candidate event space by event performance analysis and event redundancy analysis. Details of each step are presented as follows.

- Step 1. State transition identification

Based on the measurability of the optimization objective or availability of the explicit model of the optimization objective, the methods for state transition identification can be divided into indirect and direct methods. For direct method, the direct emulation (Miskowicz, Lunze 2015) of the optimization objective (assume the SCOP (Yao et al. 2004) is the optimization objective) is adopted using the SCOP-deviation-based method, which is discussed in Chapter 7. For indirect methods, knowledge-based and data-based methods are developed to find important state transitions when the model of the optimization objective is not available or the optimization objective cannot be measured directly. The methods are demonstrated by case studies in Chapter 5 and 6 respectively.

For the direct method, since the direct method emulates the optimization objective, the identified state transition is just the state of the optimization objective.

For the indirect method, the diagram (Figure 4. 4) is explained as follows.

- Generate candidate state transition space: this step aims to generate a comprehensive state transition space that contains the influencing factors of the optimization objective. No evaluation about the optimization performance is required at this step.

- Estimate optimization reward: this is to evaluate the optimization reward of candidate state transitions coarsely. For the indirect method, the knowledge-based and data-based method can be used.

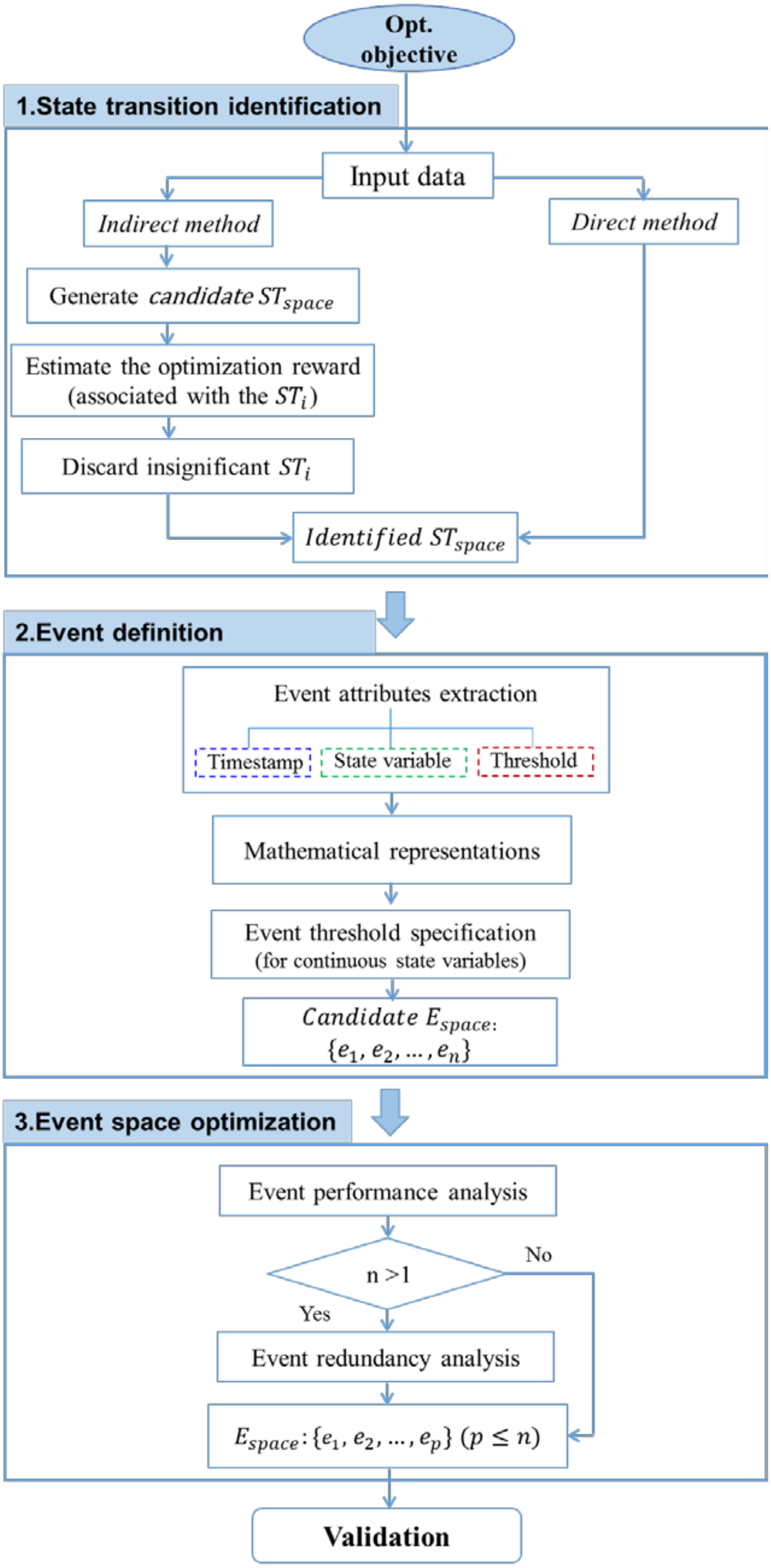

Figure 4. 4 Diagram of the methodology for event space establishment

- ➢ Discard insignificant state transitions: some bad or impractical candidates will be excluded because otherwise, it would be time-consuming and inefficient to perform detailed evaluations on all the state transitions.

- Step 2. Event definition

- ➢ Event attributes extraction: three event attributes (i.e. timestamp, state variable and threshold) will be extracted from the identified state transition space. The attribute types will be specified in order to facilitate the event definition.

- ➢ Mathematical representations for events: based on the event attributes, the state transitions are represented by mathematical expressions that the machine can process.

- ➢ Event threshold specification: for continuous state variables, the event threshold needs to be specified so as to quantify the state transition.

- Step 3. Event space optimization

- ➢ Event performance analysis: this refers to the analysis of event performance in terms of key indices, i.e. energy performance, computational performance and performance score (see Section 3.6). Please note the performance is evaluated and analyzed on a single-event basis. When necessary, the event threshold can also be optimized by selecting the best value based on the optimization performance.

- Event redundancy analysis: this is to prevent the event overlap in the same event source since it is unnecessary to repeat the optimization action for the similar state transitions.

- Event selection: based on the event performance analysis and event redundancy analysis, suitable events can be selected from the candidate event space.

The final step is to validate the EDO design which consists of the established event space, action space and policy. As discussed in Section 4.1, this step is just the step 5 in Figure 4. 1. Please note that the performance is evaluated on a multiple-event basis because the established event space normally contains multiple events. In fact, we cannot expect that a single event can reflect all the critical changes in a complex and dynamic operational environment as changes are diverse and numerous. Therefore, (when applicable) multiple events will be used to formulate the EDO policy so as to capture the critical operational change as much as possible.

4.2.3 Criteria for state transition identification: optimization reward

In Step 1 of the diagram of the event space establishment (Figure 4. 4), the most crucial step is to select important state transitions. Following criteria are developed regarding the evaluation and comparison of different state transition candidates.

- If the *optimization reward* associated with the state transition is “significant”, then we will select it;
- If the *optimization reward* associated with the state transition is “small”, then we will not select it.

(Please note: as each state transition will be defined as events and used to trigger the action of optimization, a “state transition” can be considered as an “action”, i.e. triggering the optimization.)

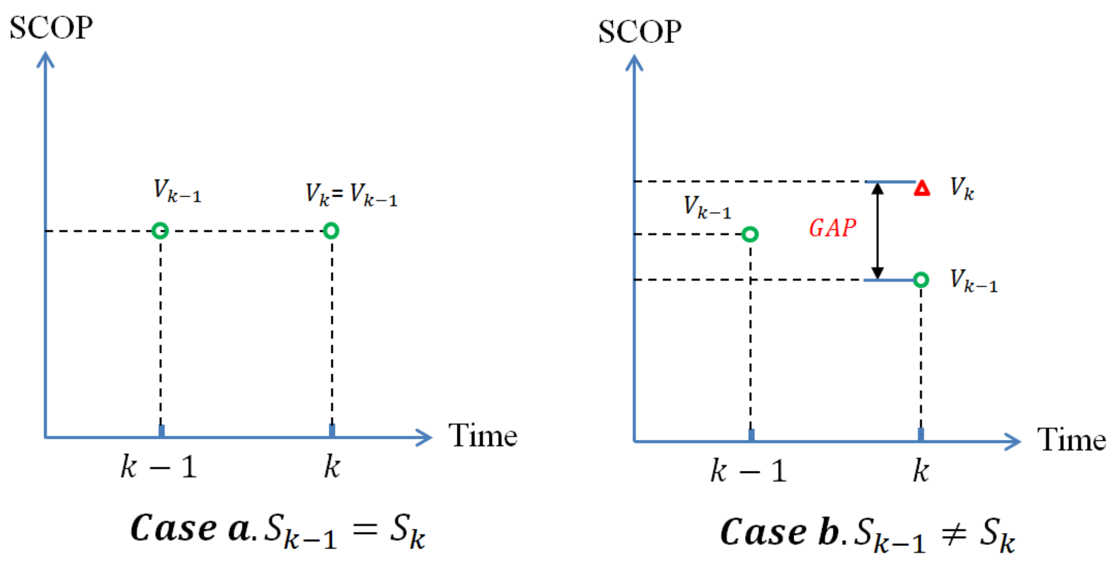

Figure 4. 5 Illustration of optimization reward

An Illustration of the *optimization reward* is shown in Figure 4. 5 to explain the idea. To proceed, suppose (i) the "operation efficiency" (i.e. SCOP) is the optimization objective; (ii) there are *n* decision variables for a decision variable vector $V = v_1, \cdots, v_n$; (iii) there are *m* state variables $S = s_1, \cdots, s_m$; (iv) at time *k*, the optimal decision variable vector is $V_k = v_{1,k}, \cdots, v_{n,k}$, and system state is $S_k = s_{1,k}, \cdots, s_{m,k}$;(v) time *k*-1 is the previous optimization time instance.

Typically, we have the following observations in a system operation:

(1) If the system state remains the same at time *k*-1 and *k* (case *a* of Figure 4. 5), the optimal decision variable vector V_k should be the same with the previous step V_{k-1} and SCOP stays at the same level;

(2) When the system state varies (case b of Figure 4. 5), if still using the previous settings V_{k-1} at time k, the $SCOP_k^{V_{k-1}}$ (SCOP at time k using V_{k-1}) would not be optimal since V_{k-1} is no long optimal for S_k.

By optimizing the decision variable vector from V_{k-1} to V_k, the SCOP would be improved. The performance gap between $SCOP_k^{V_k}$ and $SCOP_k^{V_{k-1}}$ is called *optimization reward* (shown in Figure 4. 5). This optimization reward will be used as the basic criteria to establish the event space.

4.3 Summary

This chapter develops a design approach for the EDO in order to ensure that the EDO can achieve satisfactory optimization performance. A five-step design procedure is introduced for building the {event, policy, action} structure. Considering the possible events (or state transitions) could be numerous in a system, a general decision making flow chart is given to establish the event space. A more specific methodology for event space establishment is developed, which contain three steps, namely, state transition identification, event definition and event space optimization. Both direct and indirect methods are suggested to identify important state transitions, which will be discussed and demonstrated by the case studies in Chapter 5, 6 and 7. At last, the idea of the optimization reward is illustrated, which will be used as the basic criterion for the event space establishment.

CHAPTER 5. KNOWLEDGE-BASED METHOD FOR EVENT SPACE ESTABLISHMENT

This chapter presents the knowledge-based method for event space establishment. Firstly, the literature is extensively visited in Section 5.1 to find critical events. For continues state variables, since different threshold values may lead to different optimization performance, an algorithm for event threshold selection is proposed in Section 5.2. Section 5.3 introduces the case study setup. The extracted event attributes, mathematical representations and results are presented in Section 5.4. Finally, findings are summarized in Section 5.5.

5.1 Knowledge-based Method for State Transition Identification

As the event space establishment contains three steps (shown in Figure 4.4 of Chapter 4), this section will introduce the specific algorithm for state transition identification (i.e. Step 1 in Figure 4.4). In the field of HVAC optimal control, numerous knowledge or experiences have been accumulated. These knowledge are used as the data input to the algorithm of the knowledge-based method for the state transition identification (Figure 5. 1). The identified state transition space will then be used as the input for event definition and event space optimization (i.e. Step 2 and 3 in Figure 4.4).

In Figure 5. 1, firstly, the engineering handbooks or design guide books are used as the knowledge base to enumerate the possible state transitions, such as the ASHRAE handbook and CIBSE Guide.

After the candidate state transition space ($Candidate\, ST_{space}$) is formed, the optimization reward associated with each state transition is estimated based on literature review. The literature, like academic papers, technical reports, patents, theses and books, can be used.

Then, a judgment regarding the associated optimization reward is performed. If the optimization reward estimation is "significant", the state transition will be included in a list; otherwise, it will be discarded. This process is iterated until the entire candidate state transition space is visited. Finally, the algorithm outputs the identified state transition space (ST_{space}).

For instance, the literature states variable "A" is more important than variable "B" in terms of optimization. Then, the important state transitions can be identified. Please note that this estimation of the optimization reward is qualitative.

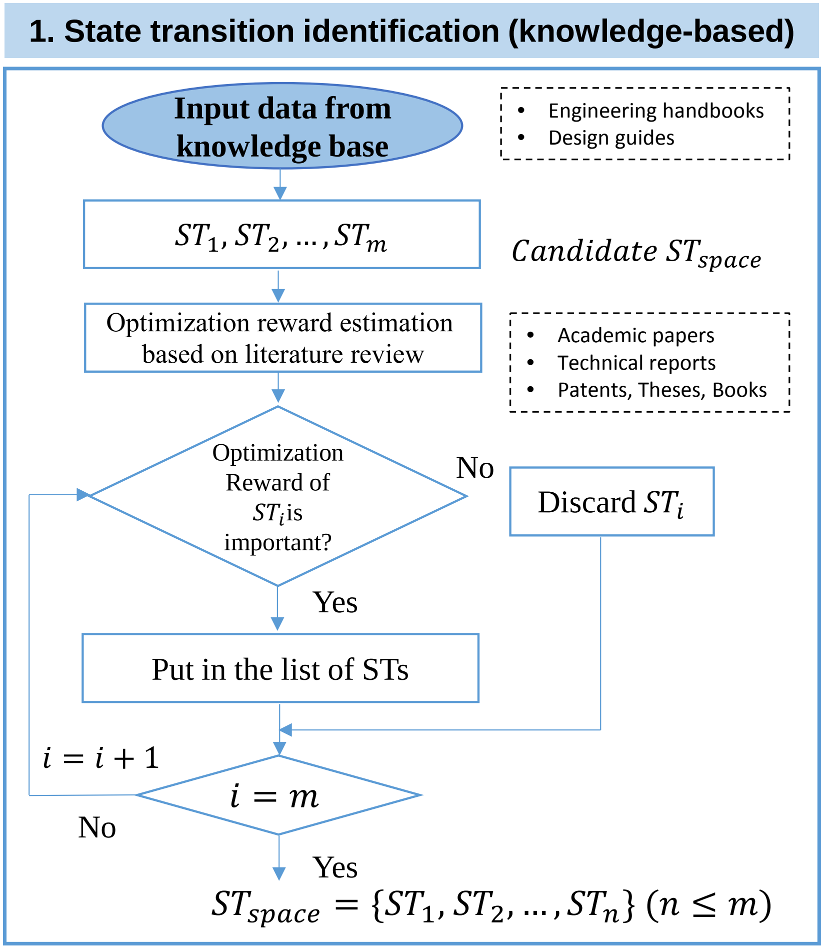

Figure 5. 1 Algorithm of knowledge-based method for state transition identification

(*ST* = state transition.)

5.2 Algorithm for Event Threshold Selection

The event threshold needs to be specified for events represented by continuous state variables. Since different event thresholds may lead to different optimization performances, a simple algorithm (Figure 5. 2) is developed and suggested to specify the suitable event threshold based on simulations. Explanations of each step are given as follows.

- Step 1: input an event for which threshold is not quantified.

- Step 2, base case: set a base case threshold (σ_{BC}). Rational analyses, such as design guides and manufactures' catalogs, can be used to find a suitable base case threshold.

- Step 3, design space: the σ_{BC} is then used as the design space center, and a certain perturbation (δ) is applied to this center.

- Step 4, performance evaluation: the optimization performance of events will be quantified by energy saving, computation saving and performance score (see Section 3.6) in this study.

- Step 5, output event threshold: based on the event performance evaluation, a suitable threshold can be selected, i.e. the threshold with highest performance score in the design space will be selected. Then, the algorithm will go to next event to conduct performance evaluation until all the events in E_{space} are visited.

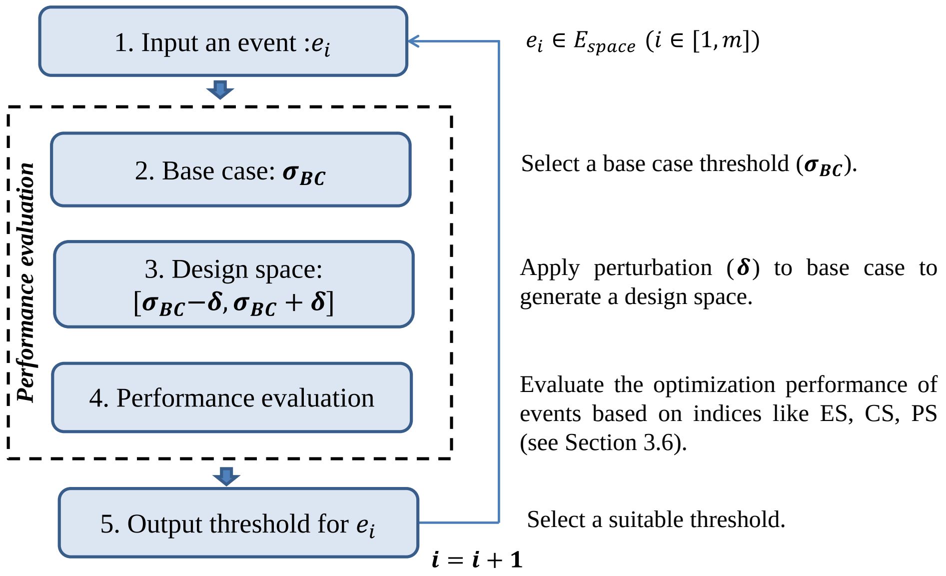

Figure 5. 2 Algorithm for event threshold selection

5.3 Case Study: State Transition Identification and Data Preparation

<u>*5.3.1 State transition identification*</u>

Possible state transitions can be found in chapter 42 of ASHRAE handbook-HVAC applications, including state variables like PLR, equipment operating status and number, temperature, humidity, water flow rate, air flow rate, frequency of VFDs (variable-frequency drive), opening percentage of valves or dampers, occupants' thermal and air quality requirements, etc. There are many options in the HVAC systems. Since not every state transition is suitable to be used as the event, simple evaluations were performed based on the literature review regarding the optimization rewards. Using the knowledge-based method, five critical state transitions acquired from the literature are listed in Table 5. 1.

Table 5. 1 List of critical state transitions for HVAC optimal control from literature

Descriptions of state transitions	**References**
Part-load-ratio change	(ASHRAE 2015, Yu, Chan 2010, Yu, Chan 2008, Ahn, Mitchell 2001, Sun, Reddy 2005, Abou-Ziyan, Alajmi 2014)
Chiller sequence change	(Huang, Zuo & Sohn 2016, Ma 2008)
Ambient wet-bulb temperature change	(Huang, Zuo & Sohn 2017, Yao et al. 2004, Ahn, Mitchell 2001, Sun, Reddy 2005, Lam, Wan & Cheung 2009)
Cooling tower approach temperature (difference between cooling tower outlet temperature and entering wet-bulb temperature)	(Chang et al. 2015, ASHRAE 2015)
The average enthalpy difference between the specific saturated (at inlet and outlet cooling water temperature) and bulk air	(Chang et al. 2015, ASHRAE 2015)

I. Part-load-ratio change: when the PLR changes by a significant amount since the last optimization, an action should be taken to optimize the control settings since previous settings may not be optimal.

II. Chiller sequence change: when a chiller is switched on or off, the load distribution among chillers will have a sudden change, and thus an optimization is needed.

III. Ambient wet-bulb temperature change, cooling tower approach temperature, and average enthalpy difference basically all reflect the efficiency of heat rejection process in cooling towers (Chang et al. 2015), thus they can be used as the optimization indicators.

5.3.2 Data preparation

Table 5. 2 Load and weather data (in the year of 2013)

Case	Date	Load (kW)			T_{db} (˚C)			T_{wb} (˚C)		
		mean	max	min	mean	max	min	mean	max	min
Autumn	Oct-21	10629	15675	3623	24.8	27.3	22.9	21.5	22.5	19.5
Summer	Aug-27	14971	23416	3586	29.3	32.6	27.0	26.4	28.3	25.4
Spring	Apr-9	6466	10163	3351	19.4	21.1	18.5	18.8	20.2	17.6

The identified candidate state transition space will be used to form the event space. The event space will be tested on three typical load profiles measured from the real building to evaluate the optimization performance. Load and weather data are shown in Table 5. 2, and their profiles are shown in Figure 5. 3-Figure 5. 5.

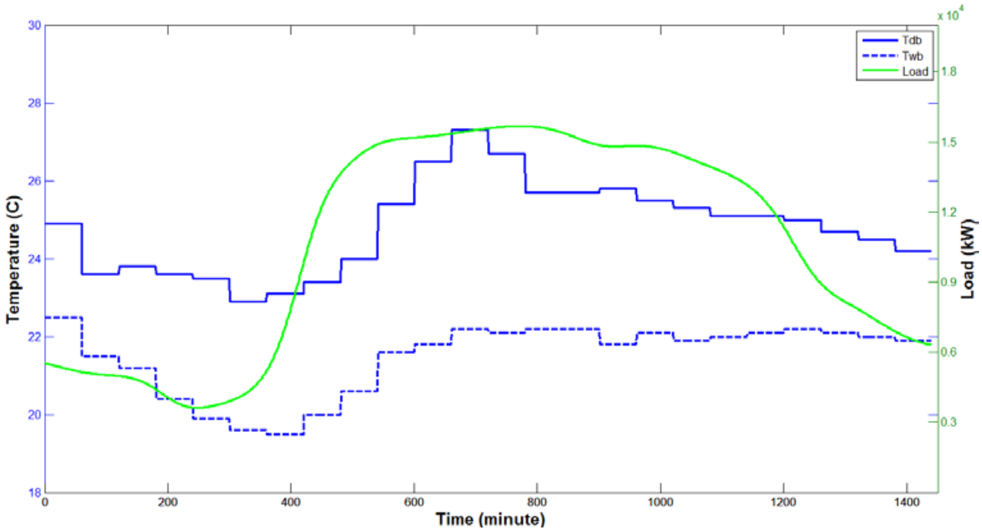

Figure 5. 3 Autumn load and weather profile (Oct-21)

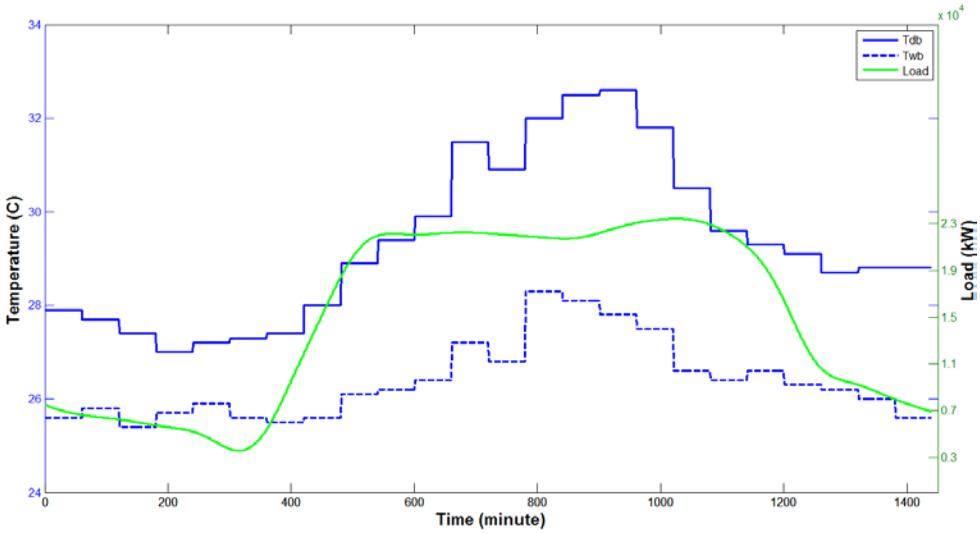

Figure 5. 4 Summer load and weather profile (Aug-27)

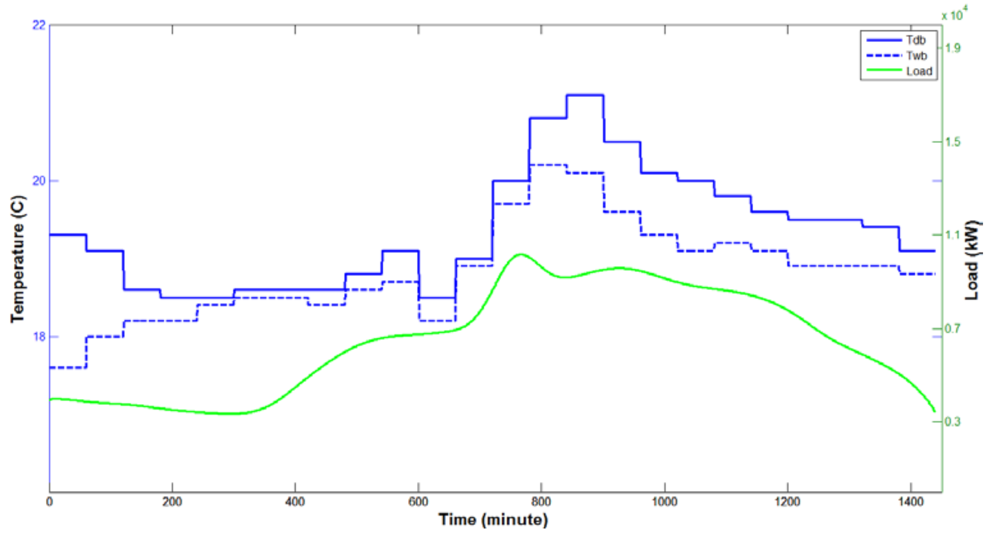

Figure 5. 5 Spring load and weather profile (Apr-9)

5.4 Case Study: Event Space Optimization and Validation

5.4.1 Event attributes extraction and mathematical representations

By abstraction, three basic event attributes of the discovered critical state transitions (Table 5. 1) are extracted in Table 5. 3. Timestamps of all the events belong to the type of instant. Except the “Chiller On/Off” uses discrete state variable, all the others use the continuous state variable and require threshold values to further quantify the state transitions. The events "T_{apr}" and "Δh" use the absolute-type threshold, while the others adopt the delta-type threshold.

Based on the types of extracted event attributes, the state transitions listed in Table 5. 1 are mathematically represented as events (Table 5. 4), and are explained as follows.

Table 5. 3 Types of extracted event attributes

State variable	**State variable type**	**Timestamp type**	**Threshold type**
PLR	continuous	instant	delta
Chiller operating status	discrete	instant	delta
T_{wb}	continuous	instant	delta
T_{apr}	continuous	instant	absolute
Δh	continuous	instant	absolute

Table 5. 4 Mathematical representations of events

Event name	**Definition**
PLR Change	$e_{PLR\ Change} := \{< X_{\tau_i}, X_{\tau_j} > \mid \lvert PLR_{\tau_i} - PLR_{\tau_j} \rvert \geq \sigma_{PLR\ Change}\}$
Ch. On/Off	$e_{Ch.\ On/Off} := \{< X_{\tau_i}, X_{\tau_j} > \mid \lvert Num_{\tau_i}^{CH} - Num_{\tau_j}^{CH} \rvert > 0\}$
T_{wb} Change	$e_{T_{wb}\ Change} := \{< X_{\tau_i}, X_{\tau_j} > \mid \lvert T_{\tau_i}^{wb} - T_{\tau_j}^{wb} \rvert \geq \sigma_{T_{wb}\ Change}\}$
T_{apr}	$e_{T_{apr}} := \{< X_{\tau_i}, X_{\tau_j} > \mid T_{\tau_i}^{apr} \geq \sigma_{T_{apr}}, T_{\tau_j}^{apr} < \sigma_{T_{apr}}\}$
Δh	$e_{\Delta h} := \{< X_{\tau_i}, X_{\tau_j} > \mid \Delta h_{\tau_i} \geq \sigma_{\Delta h}, \Delta h_{\tau_j} < \sigma_{\Delta h}\}$

(Note: τ_j is the current decision time instant and τ_i is the previous decision time instant.)

- "PLR Change" is the PLR difference between the current and previous time instants; when the PLR difference is larger than a threshold, the event will be recognized.
- "Ch. On/Off" stands for the change of chiller operating status; when the chiller(s) is/are turned on or off, the event will be recognized.

- "T_{wb} Change" is the T_{wb} difference between the current and previous time instants; when the T_{wb} difference is larger than a threshold, the event will be recognized.
- "T_{apr}" will be recognized when the approach temperature is lower than a threshold.
- "Δh" will be recognized when the average enthalpy difference between the cooling tower and bulk air is lower than a threshold.

(Please note that "T_{apr}" and "Δh" both represent the natural driving force of heat rejection process in cooling towers, and a large value is preferable.)

5.4.2 Design space for event threshold selection

To make the performance comparison and optimize the event threshold, each event was assigned with seven candidate thresholds within the corresponding threshold design space. The threshold design space was decided according to the normal range of the variable or engineering design handbook. It was discretized by a step change of 10% of "σ_{BC}", and thus the threshold design space is "$[70\% * \sigma_{BC}, 130\% * \sigma_{BC}]$".

For the event "T_{wb} Change", due to the coarse resolution of wet-bulb temperature data, the step change can only be 0.1 °C instead of 10% of "σ_{BC}". This results in a slightly different set-up, but it is still acceptable since it would not affect the general pattern of the results. In Table 5. 5, the base case values, design space as well as the lower and upper bounds (if available) of four events are presented. Based on the "Cooling Technology Institute" - CTI STD-201 (Cooling Technology Institute 2016), 2.8 ºC was selected as the lower bound of the approach temperature (T_{apr}).

Table 5. 5 Design space for event threshold selection

Parameter	**Lower bound**	**Upper bound**	**Base case**	**Design space**	**Base case calculation**
PLR change (%)	/	/	10	[7, 13]	Recommended by (ASHRAE 2015)
T_{wb} change (ºC)	0.1	2.2	0.4	[0.1,0.7]	$0.4 = 0.1 + 0.1 * 3$
T_{apr} (ºC)	2.8	6.8	4.0	[2.8, 5.2]	$4.0 = 2.8/70\%$
Δh (kJ/kg)	23*	/	33	[24, 42]	$33 \approx 23/70\%$

*: see "Appendix C" for the calculation details.

With these threshold values specified, the Step 2 (see Figure 4.4) was completed and the candidate event space was established. The candidate event space was input to the step 3 (see Figure 4.4), in which the algorithm of event threshold selection (Figure 5. 2) was executed.

5.4.3 Energy and computational performance analyses

To evaluate the optimization performances of events from Table 5. 4, three typical operation scenarios were simulated, namely autumn, summer and spring cases. The average energy and computational performances (averaged over seven candidate threshold values) of different events are listed in Table 5. 6-

Op. methods	**EC (Kwh)**	**ES**	**Op. times**	**CT(s)**	**CS**	**Event threshold**
No Op.	197663	N/A	0	N/A	N/A	
15 mins	177456	10.22%	96	138.9	0.00%	
Ch. On/Off	182030	7.91%	3	1.83	98.68%	
PLR Change	177603	10.15%	33.0	55.0	60.38%	7%-13%
STD		0.60%			7.12%	
T_{wb} Change	182153	7.85%	7.6	11.6	91.62%	0.1-0.7 ºC
STD		0.85%			10.22%	
T_{apr}	180031	8.92%	50.3	58.9	57.58%	2.8-5.2 ºC
STD		1.00%			37.93%	
Δh	178231	9.83%	50.3	68.6	50.64%	24-42kJ/kg
STD		1.02%			33.58%	

Table 5. 8, while the detailed data is presented in Appendix D. Please note that the energy saving is computed based on the “No Op.” case (no optimization is performed), and the computation saving is calculated based on the case “15 mins” (i.e. one optimization per 15 minutes).

Regarding the energy performance, “PLR Change” has the highest saving in autumn and summer cases, while “Δh” achieves the highest in spring case. The reason may be due to the high relative humidity (RH) in spring case (i.e. RH is 94.6%), which makes the "Δh" more important since the heat rejection process in cooling towers becomes critical in terms of the operating efficiency. (The RH in autumn and summer cases are 75% and 79.8%.)

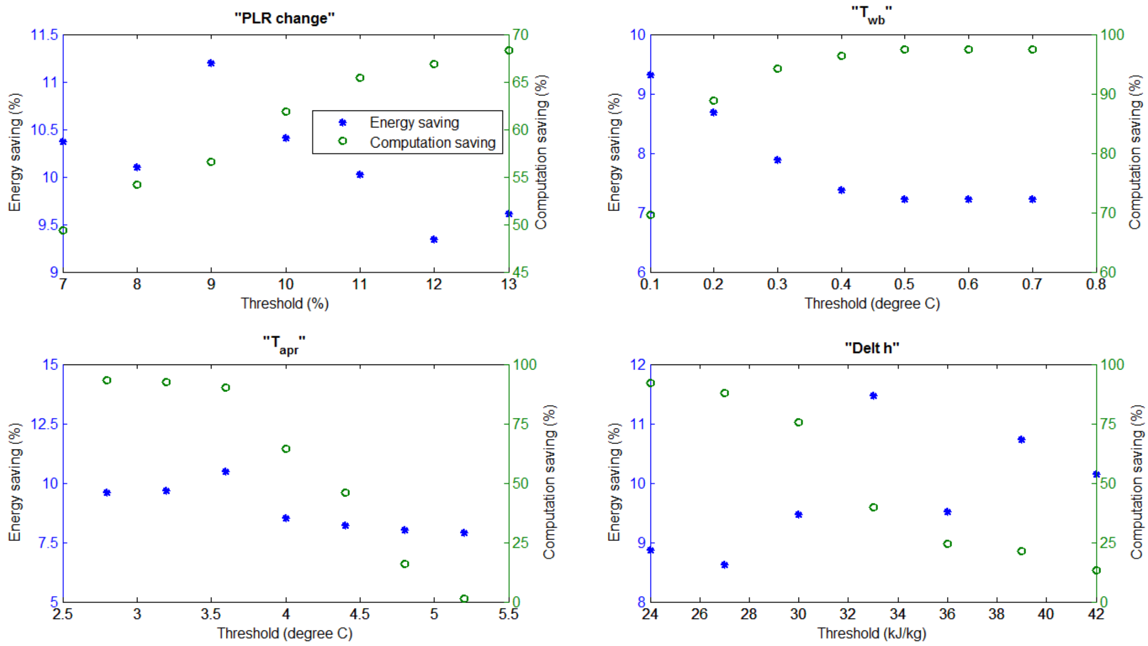

Figure 5. 6 Threshold values vs. energy savings (summer case)

(Data is shown in Appendix D.)

In terms of STDs of the energy saving, "PLR Change" has the lowest value, which shows good robustness under different event threshold values. In order to investigate the relationship between energy savings and event threshold values, the summer case is plotted in Figure 5. 6. Except "T_{wb} Change", all the events do not have a clearly linear relationship with the event threshold, which makes it hard to decide the threshold. For "T_{wb} Change", since the step change is larger (more than 10% of the base case), the relationship pattern appears to be linear. Another observation is that a higher optimization times (or triggering times) do not necessarily mean a higher energy saving. For instance, in the autumn case, "Δh" has a higher optimization times than "PLR change", but the energy saving is lower. Even with the same optimization times, as shown by the cases of "Δh" and "T_{apr}" in summer case, the average energy saving can be quite different (the difference is 0.91%).

Table 5. 6 Average energy and computational performances of different events (autumn case)

Op. methods	**EC (Kwh)**	**ES**	**Op. times**	**CT(s)**	**CS**	**Event threshold**
No Op.	132416	N/A	0	N/A	N/A	
15 mins	120429	9.05%	96	109.8	0.00%	
Ch. On/Off	125437	5.27%#	4	6.69	93.91%^	
PLR Change	120667	8.92%	26.0	26.6	75.80%	7%-13%
STD		0.08%			3.19%	
T_{wb} Change	123970	6.38%	7.6*	8.7	92.07%	0.1-0.7 ºC
STD		1.80%			7.72%	
T_{apr}	121728	8.07%	9.3	11.9	89.20%	2.8-5.2 ºC
STD		0.48%			10.16%	
Δh	120912	8.69%	42.6	43.0	60.87%	24-42kJ/kg
STD		0.39%			42.86%	

("Op." = "Optimization"; "CT" = "Computation Time"; "STD" = "Sample Standard Deviation")

$\text{Energy saving} = \frac{(132416-125437)}{132416} \times 100\% = 5.27\%$

^ $\text{Computation saving} = \frac{(109.8-6.69)}{109.8} \times 100\% = 93.91\%$

* The decimal is because this is the average optimization times

Table 5. 7 Average energy and computational performances of different events (summer case)

Op. methods	**EC (Kwh)**	**ES**	**Op. times**	**CT(s)**	**CS**	**Event threshold**
No Op.	197663	N/A	0	N/A	N/A	
15 mins	177456	10.22%	96	138.9	0.00%	
Ch. On/Off	182030	7.91%	3	1.83	98.68%	
PLR Change	177603	10.15%	33.0	55.0	60.38%	7%-13%
STD		0.60%			7.12%	
T_{wb} Change	182153	7.85%	7.6	11.6	91.62%	0.1-0.7 ºC
STD		0.85%			10.22%	
T_{apr}	180031	8.92%	50.3	58.9	57.58%	2.8-5.2 ºC
STD		1.00%			37.93%	
Δh	178231	9.83%	50.3	68.6	50.64%	24-42kJ/kg
STD		1.02%			33.58%	

Table 5. 8 Average energy and computational performances of different events (spring case)

Op. methods	**EC (Kwh)**	**ES**	**Op. times**	**CT(s)**	**CS**	**Event threshold**
No Op.	77279	N/A	0	N/A	N/A	
15 mins	73670	4.67%	96	97.6	0.00%	
Ch. On/Off	75299	2.56%	2	3.00	96.93%	
PLR Change	74028	4.21%	16.9	17.8	79.96%	7%-13%
STD		0.35%			8.39%	
T_{wb} Change	74846	3.15%	5.1	6.9	92.62%	0.1-0.7 ºC
STD		0.87%			10.60%	
T_{apr}	75737	2.00%	1.4	2.1	97.82%	2.8-5.2 ºC

STD		2.02%			3.13%	
Δh	73196	5.28%	69.6	57.6	38.63%	24-42kJ/kg
STD		0.75%			87.35%	

Regarding the computational performance, except the case of "Δh" in the spring day, more than 50% of computation was saved, which demonstrates a great potential of EDO in reducing the computational load. All computation savings have a monotonic variation pattern with the varying threshold. This is because the times of event triggering is proportional to the quantity of the variations of continuous parameters. This is also a useful property for computation estimation when a reference case is available. Specifically, three cases of "Δh" achieved negative computation savings in autumn and spring days since too much optimization were triggered (i.e. 155, 102 and 255 times of optimization, see Appendix D). This can be regarded as the case of frequent event triggering, which should be prevented by the careful threshold selection. For STDs of the computation saving, "PLR Change" has relatively good performance in three cases.

5.4.4 Performance score and event redundancy analyses

This section presents detailed analyses regarding the performance score. "T_{wb} Change", "T_{apr}" and "Δh" are grouped into one event group as they indicate the efficiency of heat rejection process in cooling towers (Chang et al. 2015).

The performance scores, averages and CVs (coefficient of variation) of different events are shown in Table 5. 9. “1.000” was set as the benchmark performance score. A higher performance score means the combined energy and computational performance is better, while a lower CV indicates a steadier performance in terms of the varying thresholds. For each event, the performance was evaluated by averaging the performances under three operation scenarios. The event threshold with the highest average performance score was underlined for the easy reference, which will be used when that event is chosen.

“PLR Change” performed the best since it has the highest averaged performance score (1.322) and lowest performance variance (2.3%). In the group of “T_{wb} Change”, “T_{apr}” and “Δh”, “Δh” got the highest averaged performance score (1.268), but the CV is also the highest (20.5%), which means it performed the best but not very robust under different event threshold values. “T_{wb} Change” achieved the moderate performance score (1.176) and CV (9.2%), while “T_{apr}" had the lowest performance score (1.138) and CV (4.3%). It should be pointed out that the low-performance scores in autumn and spring cases of the event “Δh” (threshold=42 kJ/kg) are partially due to the bad choice of the threshold, which can be prevented by careful analyses.

Thus, given its high-performance score, "Δh" is selected, and "T_{wb} Change" and "T_{apr}" are discarded to avoid the potential event redundancy. In terms of seasonal patterns, it is observed that (except the event "Δh") the average performance score of each event is low in spring case comparing with the other two cases. Moreover, the CVs are all very high in spring case. The possible reason is that the weather condition in the spring case is extreme, i.e. the outdoor condition is humid (RH is 94.6%).

Table 5. 9 Performance scores of different events in three operation scenarios

Event name	**Threshold**	**Autumn**	**Summer**	**Spring**	**Average**
Ch. On/Off		1.052	1.267	1.033	1.117
PLR Change	7%	1.422	1.261	1.225	1.303
	8%	1.342	1.259	1.317	1.306
	9%	1.340	1.379	1.271	1.330
	10%	1.362	1.327	1.375	1.355
	<u>11%</u>	1.362	1.307	1.435	<u>1.368</u>
	12%	1.365	1.248	1.242	1.285
	13%	1.358	1.281	1.275	1.304
Average		1.364	1.295	1.306	**1.322**
CV		2.0%	3.6%	5.8%	**2.3%**
T_{wb} Change					
	0.1 ºC	1.304	1.260	1.325	1.296
	<u>0.2 ºC</u>	1.342	1.294	1.360	<u>1.332</u>
	0.3 ºC	1.336	1.242	1.068	1.215
	0.4 ºC	1.199	1.203	1.052	1.151
	0.5 ºC	1.040	1.194	1.052	1.095
	0.6 ºC	0.967	1.194	1.052	1.071
	0.7 ºC	0.967	1.194	1.052	1.071
Average		1.165	1.226	1.137	**1.176**
CV		14.7%	3.3%	12.4%	**9.2%**
T_{apr}					

	2.8 ºC	1.269	1.405	0.500	1.058
	3.2 ºC	1.400	1.408	0.500	1.103
	3.6 ºC	1.385	1.478	0.500	1.121
	4 ºC	1.357	1.155	0.982	1.165
	4.4 ºC	1.340	1.033	1.110	1.161
	<u>4.8 ºC</u>	1.335	0.865	1.433	<u>1.211</u>
	5.2 ºC	1.278	0.779	1.391	1.149
Average		1.338	1.160	0.916	**1.138**
CV		3.7%	24.1%	45.7%	**4.3%**
Δh	24 kJ/kg	1.362	1.328	1.407	1.366
	<u>27 kJ/kg</u>	1.362	1.283	1.865	<u>1.503</u>
	30 kJ/kg	1.392	1.303	1.634	1.443
	33 kJ/kg	1.383	1.321	1.630	1.445
	36 kJ/kg	1.278	1.053	1.252	1.194
	39 kJ/kg	1.187	1.156	1.166	1.170
	42 kJ/kg	*0.882*	1.059	*0.319*	0.753
Average		1.264	1.215	1.325	**1.268**
CV		14.5%	10.1%	38.1%	**20.5%**

("CV" = "Coefficient of Variation")

5.4.5 Validation: optimization performance of established event space

This section is to validate the optimization performance of the identified event space. Based on the performance analyses of different events presented in the previous sections, following event space and EDO policy are established (Figure 5. 7). Please note that the default action is adopted, i.e. if any of events in the event space happened, the action will be taken to optimize all the decision variables.

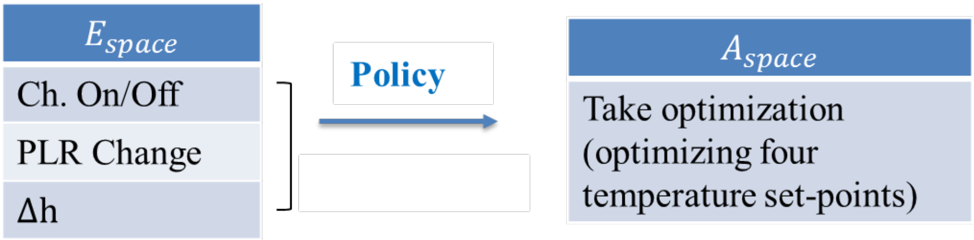

Figure 5. 7 EDO policy

Table 5. 10 Comparison of EDO using multiple events and single event

Op. methods	**ES**	**CS**	**PS**
Autumn case			
15mins	9.05%	0.00%	1.000
Multiple events	**9.09%**	**71.66%**	**1.363**
Δh	8.35%	88.05%	1.362
PLR Change	8.72%	79.71%	1.362
Summer case			
15mins	10.22%	0.00%	1.000
Multiple events	**10.39%**	**62.23%**	**1.327**
Δh	8.63%	87.76%	1.283
PLR Change	10.02%	65.41%	1.307
Spring case			
15mins	4.67%	0.00%	1.000
Multiple events	**6.56%**	**84.76%**	**1.829**
Δh	6.46%	96.27%	1.865
PLR Change	4.71%	85.42%	1.435

(Please note: “multiple events” contains “Ch. On/Off”, “PLR Change” and “Δh”; threshold of “PLR Change” is 11%; threshold of “Δh” is 27 kJ/kg.)

The performances of “multiple events” and single event are compared in Table 5. 10. It can be seen that “multiple events” achieves similar or higher performance scores than the single event (i.e. “PLR Change” or “Δh”). Compared with TDO benchmark or single event, the energy savings of “multiple events” are higher in all the three cases. Another observation is that, comparing with the single event, the computational performance of “multiple events” is slightly deteriorated (more events trigger more optimizations) while the energy performance is further improved.

5.4.6 Discussions

In this section, a comprehensive evaluation of the optimization performance is presented for the critical events found from literature. The algorithm of event threshold selection is proposed. The findings are summarized as follows.

- The potential advantage of EDO for reducing computation is demonstrated by the case study. However, the poor-selected threshold would increase the computation due to the frequent event triggering, which is illustrated by the autumn and spring cases of the event “Δh”.

- Results show that the event “PLR change” has relatively good performance among all the five event candidates. It has relatively high energy saving and good robustness under different threshold values and operational scenarios. Therefore, “PLR change” is suggested as a general event that can be used to trigger the optimization.

- The performance score (combines the energy and computation performances) can be used as a more comprehensive index for optimization performance evaluation and decision making. The case study demonstrates that good events can be selected out with the help of performance scores.

- It is found that the optimization performances of the EDO is related to the operation conditions. For instance, the "Δh" is more critical when the RH is high (e.g. more than 90%), which is demonstrated by the case of spring scenario.

- The relationship between the energy saving and event threshold is no linear, which makes the threshold selection difficult. Extensive simulation or testing efforts are always required.

- The assumption of "multiple events" is confirmed in the case study. It is found that a better performance can be obtained when more critical events are properly defined and used.

5.5 Summary

The aim of this chapter is to demonstrate the knowledge-based method (an indirect method) for event space establishment. The possible state transitions are enumerated based on the engineering handbooks. From these state transitions, five critical ones are identified by the evaluations based on literature review, which are defined as events. An algorithm is also developed to optimize the threshold of events with continuous-form state variables. For each event, seven different thresholds are tested based on three typical load and weather profiles. The optimization performances of events are analyzed in a comprehensive way regarding the energy saving, computation saving and performance score. With the help of performance score, the best event and event threshold are identified in each event source. Insignificant events are not used to avoid potential event duplications. These identified events are then used to formulate the EDO policy. The result validates that the formulated EDO policy can effectively reduce the computational load while the energy performance is further improved. The possible reasons are that the EDO can reduce the optimization action delay and avoid unnecessary optimization actions.

In summary, the results suggest that EDO might be a good alternative for the RTO of HVAC systems (compared with the conventional TDO) considering that the operation conditions are highly stochastic. The result also suggests that a key factor to the success of the EDO is the selections of events and event thresholds. Unsuitable selections can increase the computation or deteriorate the energy efficiency. As different events from various event sources may be encountered in a system, especially when the system becomes complex and when dealing with some new building systems like green buildings and zero-energy buildings, the proposed knowledge-based method is an effective way for event space establishment. However, the event threshold selection in the knowledge-based method requires extensive efforts. Thus, next chapter will present a method to quickly find out the suitable threshold.

CHAPTER 6. DATA-BASED METHOD FOR EVENT SPACE ESTABLISHMENT

This chapter presents the data-based method for event space establishment. Section 6.1 firstly introduces the algorithm used to identify state transitions, the optimization reward estimator and the random forest algorithm for variable importance evaluation. Then, a simple method to compute the suitable event threshold based on the Euclidean distance of decision variable vectors is proposed in Section 6.2. After that, Section 6.3 and 6.4 presents a case study to demonstrate and validate the effectiveness of the proposed data-based method. Finally, the summary is given in Section 6.5.

6.1 Data-based Method for State Transition Identification

6.1.1 Algorithm of data-based method for state transition identification

The algorithm of the data-based method for state transition identification is presented in Figure 6. 1, and explanations are presented as follows.

- Firstly, the candidate state transition space is formed based on the operational data which is obtained from the system with the RTO function. The estimator of the optimization reward ($\hat{r}_{opt}$) is developed based on decision variable distance, which will be discussed in Section 6.1.2.

- Then, in order to investigate the relationship between the estimated optimization reward and state transitions, a function is established through the regression.

- Next, to find the important state transitions corresponding to the optimization reward, the evaluation of the “variable importance” is performed by the random forest algorithm (presented in Section 6.1.3).

- The variable importance output by the random forest algorithm will go through a decision box to examine whether the state transition is significant or not.

- The state transition passing the decision box will be included in a list; otherwise, it will be discarded since it is unnecessary to test the insignificant state transitions. This process is iterated until the entire candidate state transition space is visited. Finally, the algorithm outputs the identified state transition space (ST_{space}), which will be used as input for event definition and event space optimization (i.e. Step 2 and 3 in Figure 4.4) to identify the event space.

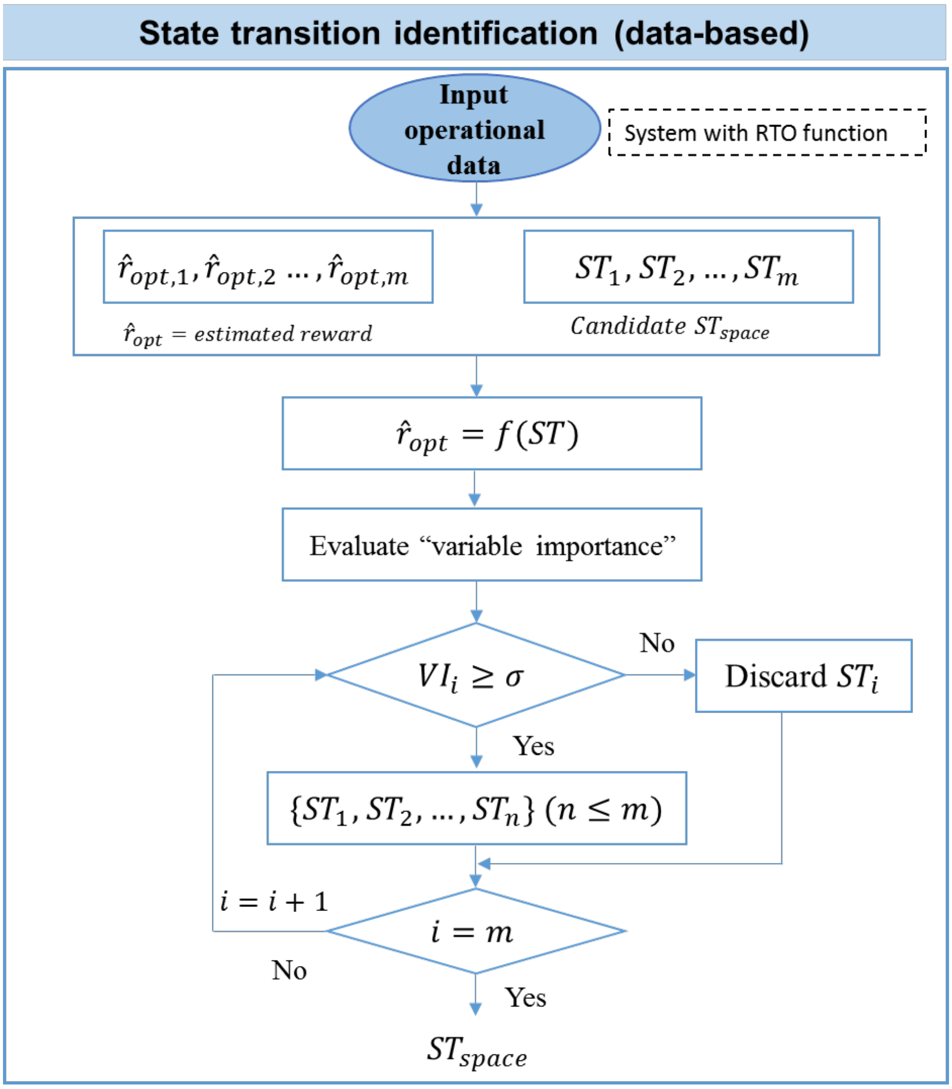

Figure 6. 1 Algorithm of data-based method for state transition identification

($\hat{r}_{opt}$=estimated optimization reward; *ST* = state transition; *VI* = variable importance)

6.1.2 Estimate the optimization reward by decision variable vector distance

In this section, a quantitative index, the Euclidean distance between adjacent decision variable vectors, is developed to estimate the optimization reward. As we know in data science, the deviation between objects is usually called "dissimilarity" (Han, Pei & Kamber 2011). The commonly used dissimilarity measure is the Euclidean distance (others are Minkowski and Mahalanobis distance), which is defined as follows (following the notation used in Section 4.2).

$$d_k = \sqrt{(v_{1,k} - v_{1,k-1})^2 + \cdots + (v_{n,k} - v_{n,k-1})^2} \tag{6. 1}$$

To eliminate the scale effect in the dataset, data normalization is always performed, and the Euclidean distance can be re-written as:

$$d_{k,norm} = \sqrt{(v_{1,k,norm} - v_{1,k-1,norm})^2 + \cdots + (v_{n,k,norm} - v_{n,k-1,norm})^2} \tag{6. 2}$$

$$v_{n,norm} = (v_n - v_{n,\min}) / (v_{n,\max} - v_{n,\min}) \tag{6. 3}$$

$$\hat{r}_{opt} = d_{k,norm} \tag{6. 4}$$

As discussed in Section 4.2.3, if system state remains unchanged, the normalized Euclidean distance ($d_{k,norm}$) is "0"; the more the system state deviates from the previous updating, the larger $d_{k,norm}$ will be. A reasonable assumption is that the larger the $d_{k,norm}$ is, the higher the optimization reward will be. Based on this assumption, the normalized Euclidean distance ($d_{k,norm}$) can be used as an estimator for the optimization reward ($\hat{r}_{opt}$) as shown in Eqn. (6. 4).

6.1.3 Evaluate variable importance by random forest

To investigate how the state variables affect the optimization reward, we can write the Euclidean distance of decision variable vectors as a function of state variables and their variations (Eqn. (6. 5)).

$$d_{k,norm} = f\left(S_k, \Delta S_k\right) = f\left(s_{1,k}, \cdots, s_{m,k}, \Delta s_{1,k}, \cdots, \Delta s_{m,k}\right) \tag{6. 5}$$

where $\Delta S_k = \left(\Delta s_{1,k}, \cdots, \Delta s_{m,k}\right)$, $\Delta s_{i,k} = \left|s_{i,k} - s_{i,k-1}\right|, i = 1, \cdots, m$.

Random forest (RF) (Breiman 2001) is used to evaluate the variable importance. RF is an ensemble learning method that constructs multiple decision trees (a forest) based on random feature selections of data. Each tree is grown by a bootstrapped sample. Variables are randomly picked at each node of the tree, which decreases the correlation between the trees in the forest, and thus decreasing the error rate.

RF have been found to be excellent comparing with other machine learning algorithms (Breiman 2001, Meyer, Leisch & Hornik 2003, Svetnik et al. 2003) and traditional logistic regression models (Hosmer Jr, Lemeshow & Sturdivant 2013). Thus, RF becomes popular recently because it is user-friendly, often accurate and is capable of handling highly correlated data. Since the prediction is based on the outputs of individual trees, RF can avoid overfitting and improve the prediction accuracy. Moreover, there are two pretty useful byproducts from RF, out-of-bag estimates of generalization error (Breiman 2001, Bylander 2002) and variable importance measures (Breiman 2001, Liaw, Wiener 2002, Svetnik et al. 2003).

The variable importance is computed from permuting out-of-bag (oob) data. The observations are considered the "oob" observations if they are not used for growing the tree. For each tree, the prediction error on the "oob" portion of the data is recorded (mean squared error (MSE) for regression). Then, the same is done after permuting each predictor variable. The difference between the prediction error on the "oob" portion of the data and the same one after permuting each predictor variable is represented by "%IncMSE" (after averaging and normalization). This is an inherent procedure of RF for evaluating variable importance, which is a robust measure (Archer, Kimes 2008) and perfect for the current application.

$$\%\mathrm{IncMSE}(v_j) = (\mathrm{MSE}(-v_j) - \mathrm{MSE})/\mathrm{MSE} \times 100\% \tag{6. 6}$$

The %IncMSE is computed in Eqn. (6. 6), where "$\mathrm{MSE}(-\mathrm{v_j})$" stands for the MSE if $\mathrm{v_j}$ is not used in the prediction. A small %IncMSE can be interpreted as: without certain variable $\mathrm{v_j}$, the MSE does not increase much, meaning that the contribution of $\mathrm{v_j}$ to the model output is not significant. On the contrary, a higher %IncMSE suggests that the variable $\mathrm{v_j}$ is more important.

6.2 Specifying the Event Threshold based on Decision Variable Distance

In the establishment of event space, event thresholds are needed to quantify the variation for continuous variables. The event threshold is user-defined parameter. Here, in order to select a suitable threshold value, a simple method is suggested based on the proposed decision variable distance.

As we know, the event should associate with the "large" optimization reward (i.e. "large" $d_{k,norm}$). Thus, the quantity of state transition associated with the "large" $d_{k,norm}$ is suggested as the event threshold. "large" $d_{k,norm}$ is firstly defined in Eqn. (6. 7) based on the common sense that "large" means more than the average.

$$d_{norm,large} := \{\forall d_{k,norm} > d_{norm,mean}, k \in \{1,2,...,n\}\} \tag{6. 7}$$

where $d_{norm,mean}$ means the mean normalized distance and $d_{k,norm}$ is defined in Eqn. (6. 2).

Then, the corresponding state variables and state variable variations associated with the large $d_{k,norm}$ can be identified using the $Index_{large}$, which are represented in Eqn. (6. 9).

$$Index_{large} := \{\forall k, d_{k,norm} > d_{norm,mean}, k \in \{1,2,...,n\}\} \tag{6. 8}$$

$$\Delta S_{large} := \{\Delta \mathrm{s_k}, k \in Index_{large}\}, \quad S_{large} := \{\mathrm{s_k}, k \in Index_{large}\} \tag{6. 9}$$

Finally, the event threshold is calculated based on Eqn. (6. 10) and (6. 11). In this case study, the mean of the state variable (or state variable variation) associated with the large $d_{k,norm}$ is suggested as the event threshold since the mean can represent the typical values in a data set.

$$\sigma s_k := \{\sum_k s_k / |S_{large}|, k \in Index_{large}\} \tag{6. 10}$$

$$\sigma \Delta s_k := \{\sum_k \Delta s_k / |\Delta S_{large}|, k \in Index_{large}\} \tag{6. 11}$$

where $|S_{large}|$ means the size of the set.

6.3 Case Study: Data Preparation and State Transition Identification

6.3.1 Data preparation

In general, the data from a BAS with RTO function can be used to identify the important variables and corresponding thresholds with the methods presented in Section 6.1 and 6.2. In this case study, the data generated from the simulation platform (see Section 3.4) with RTO function was used as an alternative. The optimization frequency “15 minutes per optimization” was used because it is a high optimization frequency that can reflect the operational condition changes in a fine resolution.

Table 6. 1 Load and weather data (in the year of 2013)

Date	Load (kW)			T_{db} (˚C)			T_{wb} (˚C)		
	mean	max	min	mean	max	min	mean	max	min
May-13	12811	19599	3860	26.0	29.8	23.9	24.4	25.9	23.4
May-14	12409	18525	4924	26.2	28.0	25.0	25.1	25.8	24.4
May-15	12148	19258	3355	28.3	30.2	26.0	26.4	27.3	25.3
May-16	11630	19964	3555	27.6	30.0	25.1	25.7	27.1	24.1
May-17	6447	9031	3354	25.9	26.9	24.9	25.2	26.2	24.3
May-20	6539	9894	3775	29.0	31.2	26.0	26.5	27.3	24.0

The real weather and load profiles on May 13-17, 2013 were used to generate data for data mining (Figure 6. 2-Figure 6. 6). Then, events were defined based on the data mining results and used to formulate the EDO policy. This EDO policy was validated using the load and weather profiles of May 20, 2013 (Figure 6. 7). All the statistical data of load and weather profiles is shown in Table 6. 1.

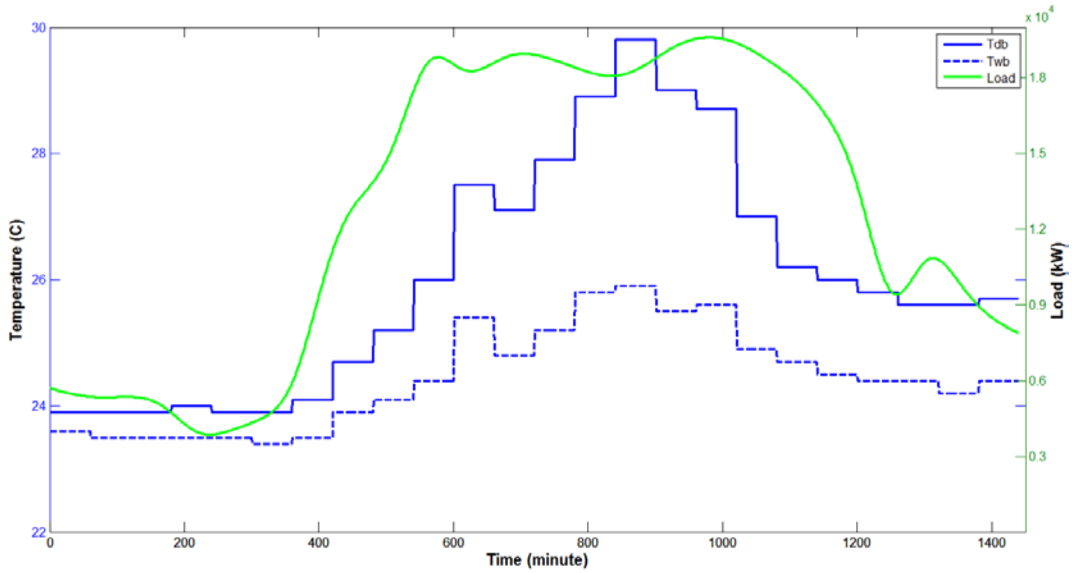

Figure 6. 2 Load and weather profile (May-13)

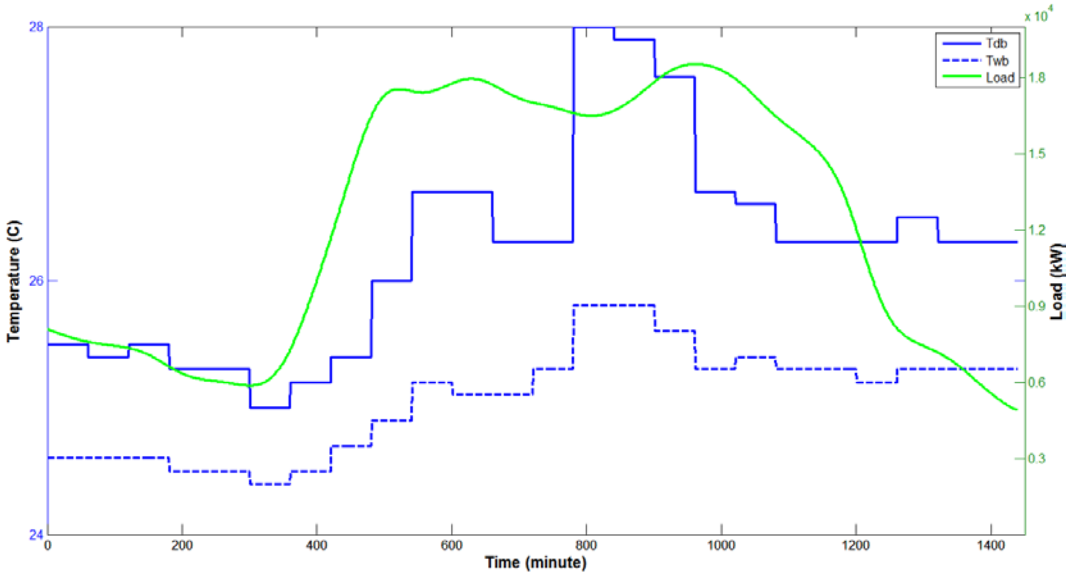

Figure 6. 3 Load and weather profile (May-14)

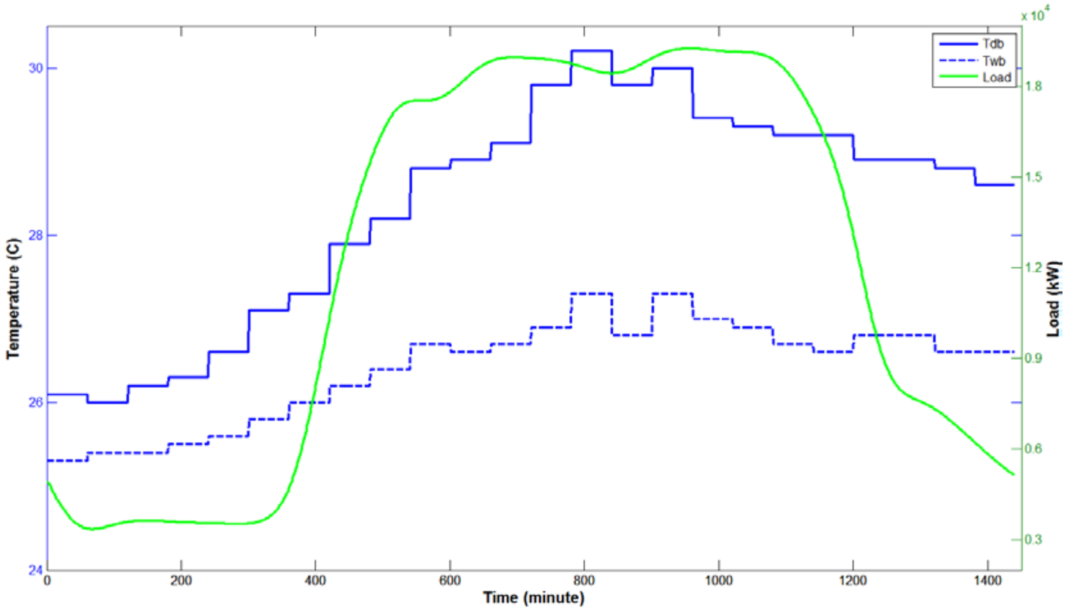

Figure 6. 4 Load and weather profile (May-15)

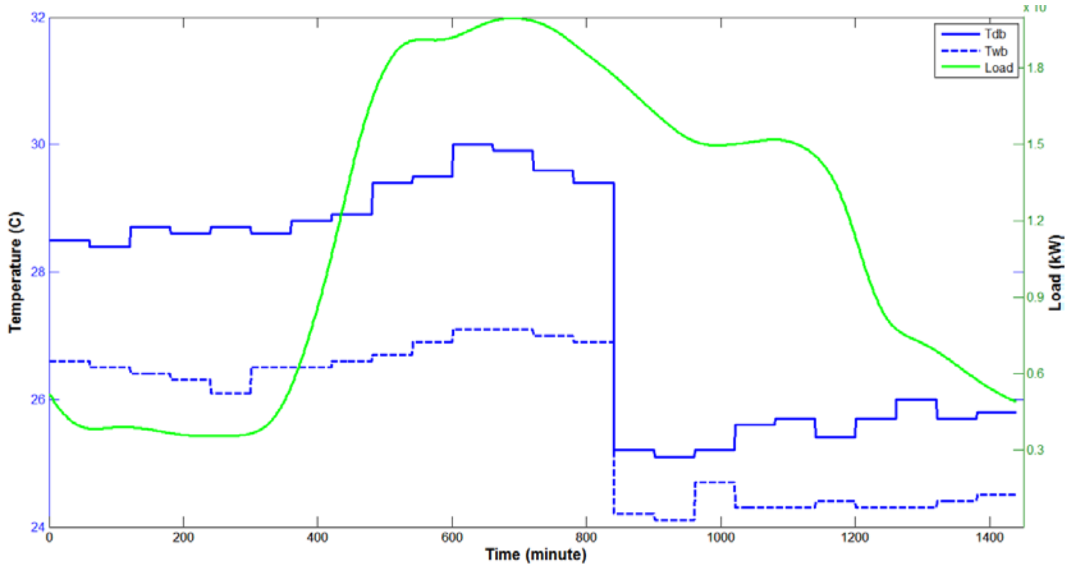

Figure 6. 5 Load and weather profile (May-16)

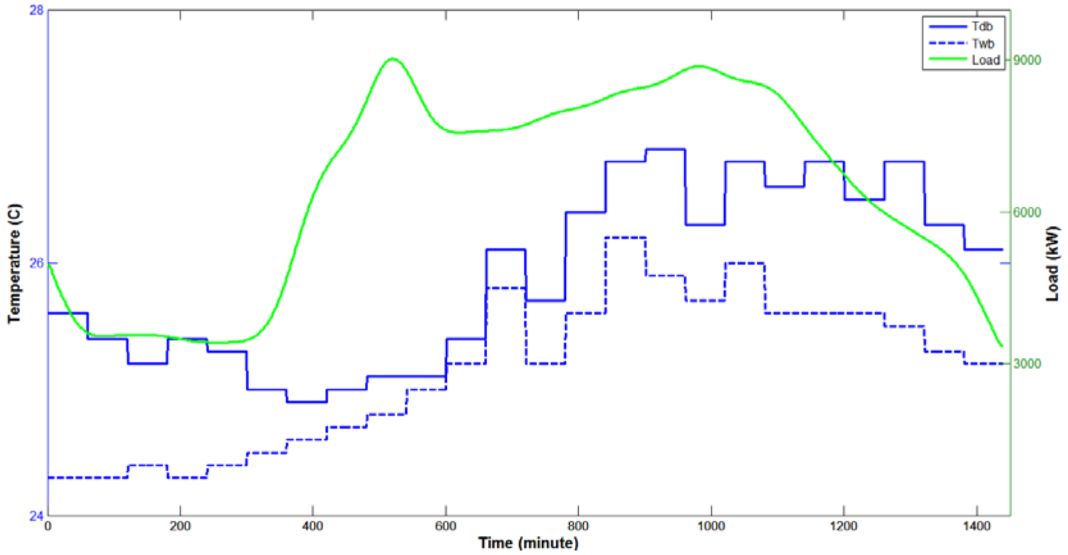

Figure 6. 6 Load and weather profile (May-17)

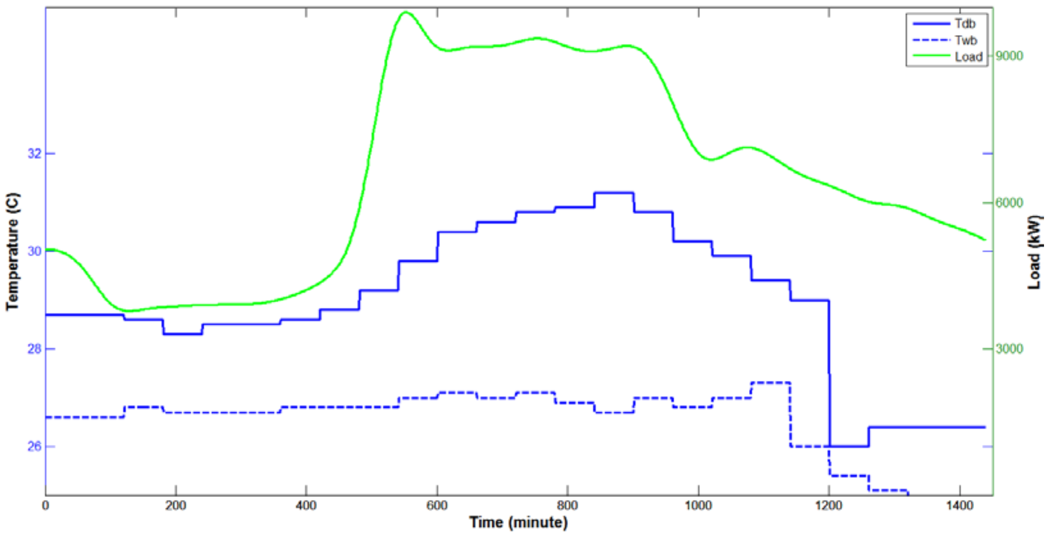

Figure 6. 7 Load and weather profile (May-20)

6.3.2 Data mining in "R"

"R" is an open platform for statistical computing, which provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, etc.) and graphical techniques. To implement the RF algorithm, the "randomForest" package (version 4.6-12) in R (version 3.3.3) (The R Foundation) is directly used. The documentation of the "randomForest" package (version 4.6-12) can be found in (Breiman et al. 2015).

The data for regression was generated by computer simulations based on the load and weather profiles of May 13-17, 2013. The R script is shown in Figure 6. 8, where 500 trees were used, and three variables were tried at each split in the random forest. The mean of squared residuals of the regression is 0.0415, which shows good prediction performance.

```r
## Regression:
## data(data_for_May_13_17_add_two_new_v2)
#library(randomForest)
#library(readr)
#library(readxl)

set.seed(11)
event1.rf <- randomForest(euclidean_distance_t_set ~ ., data=data_for_May_13_17_add_two_new_v2, mtry=3,
                          importance=TRUE, na.action=na.omit)
print(event1.rf)
## Show "importance" of variables: higher value mean more important:
round(importance(event1.rf), 2)
```

Figure 6. 8 R script of random forest algorithm

6.3.3 State transition identification based on variable importance

The variable importance output by RF is shown in Table 6. 2, where "rank 1" means the highest importance. "Δh" got the highest importance, and is greater than "T_{apr}", which agrees well with the domain knowledge and case studies presented in Chapter 5. It is observed that, using the data-based method, some critical state transitions are newly discovered in comparison with the knowledge-based method, namely "$Freq_{ct,fan}$", "$M_{w,prm,pump}$" and "$M_{w,sec,pump}$". In this study, the threshold of the variable importance was chosen to be 5% because the "%IncMSE" lower than 5% means little effects on the model output (i.e. decision variable distance). Thus, "T_{wb}", "T_{db}" and "Num_{ch}" were discarded due to the low values of %IncMSE. Finally, the critical state transitions were identified through mining the operational data.

Table 6. 2 Variable importance of state variables output by random forest

Variable	%IncMSE	Rank	Description
Δh	26.84	1	the average difference between the specific enthalpies of saturated air and bulk air
T_{apr}	21.88	2	the cooling tower approach temperature
$Freq_{ct,fan}$	18.62	3	the fan frequency of cooling tower
PLR	15.42	4	part-load ratio change
$M_{w,prm,HX}$	14.69	5	water mass flow rate of the heat exchanger at primary sided
$M_{w,sec,pump}$	11.24	6	water mass flow rate of total secondary pumps
T_{wb}	3.75	7	wet-bulb temperature change
T_{db}	3.4	8	dry-bulb temperature change
Num_{ch}	1.13	9	number of the operating chiller(s)

6.4 Case Study: Event Space Optimization and Validation

6.4.1 Event attributes extraction and mathematical representations

By abstraction, three basic event attributes of the discovered critical state transitions (Table 6. 2) are extracted in Table 6. 3. Timestamps of all the events belong to the type of instant. All the state variables belong to the type of continuous variable, which means threshold values are required. The events "Δh", "T_{apr}"and "$Freq_{ct,fan}$" use the absolute-type threshold, while the others use the delta-type threshold.

Table 6. 3 Types of extracted event attributes

State variable	State variable type	Timestamp type	Threshold type
Δh	continuous	instant	absolute
T_{apr}	continuous	instant	absolute
$Freq_{ct,fan}$	continuous	instant	absolute
PLR	continuous	instant	delta
$M_{w,prm,HX}$	continuous	instant	delta
$M_{w,sec,pump}$	continuous	instant	delta

Based on the types of extracted event attributes, the state transitions listed in Table 6. 3 are mathematically represented as events in Table 6. 4, and explanations are given as follows. Please note the explanations of "Δh", "T_{apr}" and “PLR Change” are given in Section 5.4.1, and thus will not repeat here.

Table 6. 4 Mathematical representations of events

Event name	Definition
Δh	$e_{\Delta h}:=\{<X_{\tau_i},X_{\tau_j}> \mid \Delta h_{\tau_i} \geq \sigma_{\Delta h}\ , \Delta h_{\tau_j} < \sigma_{\Delta h}\}$
T_{apr}	$e_{T_{apr}}:=\{<X_{\tau_i},X_{\tau_j}> \mid\ T_{\tau_i}^{apr} \geq \sigma_{T_{apr}}\ , T_{\tau_j}^{apr} < \sigma_{T_{apr}}\}$
$Freq_{ct,fan}$	$e_{Freq_{ct,fan}}:=\{<X_{\tau_i},X_{\tau_j}> \mid\ Freq_{\tau_i}^{ct,fan} < \sigma_{Freq_{ct,fan}}\ , Freq_{\tau_j}^{ct,fan} \geq \sigma_{Freq_{ct,fan}}\}$
PLR Change	$e_{PLR\ Change}:=\{<X_{\tau_i},X_{\tau_j}> \mid\ \lvert PLR_{\tau_i}-PLR_{\tau_j}\rvert \geq \sigma_{PLR\ Change}\}$
$M_{w,prm,HX}$ Change	$e_{M_{w,prm,HX}}:=\{<X_{\tau_i},X_{\tau_j}> \mid\ \lvert M_{\tau_i}^{w,prm,HX}-M_{\tau_j}^{w,prm,HX}\rvert \geq \sigma_{M_{w,prm,HX}}\}$
$M_{w,sec,pump}$ Change	$e_{M_{w,sec,pump}}:=\{<X_{\tau_i},X_{\tau_j}> \mid\ \lvert M_{\tau_i}^{w,sec,pump}-M_{\tau_j}^{w,sec,pump}\rvert \geq \sigma_{M_{w,sec,pump}}\}$

(Note: τ_j is the current decision time instant and τ_i is the previous decision time instant.)

- "$M_{w,prm,HX}$ Change" and "$M_{w,sec,pump}$ Change" reflect the water mass flow rates at primary and secondary side of the heat exchanger. The change of the water flow rate can partially represent the change in load condition. When the change is greater than a threshold, the event will be recognized.

- "$Freq_{ct,fan}$" is the fan frequency of cooling towers. A high fan frequency means the natural driving force is low, so the cooling water cannot reject the heat effectively, which may need an optimization. When the fan frequency is higher than a threshold, the event will be recognized.

6.4.2 Event threshold specification

The Event threshold was calculated based on Eqn. (6. 10) and (6. 11) using the operational data, and results are shown in Table 6. 5. Please note that calculation of the absolute-type threshold uses Eqn. (6. 10), while the delta-type threshold uses Eqn. (6. 11).

Table 6. 5 Event threshold specification

Event name	**Event threshold value**	**Remark**
Δh	36 kJ/kg	Eqn. (6. 10)

T_{apr}	5.5 ºC	Eqn. (6. 10)
$Freq_{ct,fan}$	36	Eqn. (6. 10)
PLR Change	6%	Eqn. (6. 11)
$M_{w,prm,HX}$ Change	4.7 L/s	Eqn. (6. 11)
$M_{w,sec,pump}$ Change	18 L/s	Eqn. (6. 11)

6.4.3 Energy and computational performance analyses

To analyze the optimization performances of events listed in Table 6. 4, energy and computational performance are presented in Table 6. 6. Please note that the energy saving is computed based on the “No Op.” case (no optimization is performed), and the computation saving is calculated based on the case “15 mins” (i.e. one optimization per 15 minutes).

Table 6. 6 Energy and computational performances of different events

Op. methods	**EC (Kwh)**	**ES**	**Op. times**	**CT(s)**	**CS**	**Event threshold**
No Op.	158197	N/A	0	N/A	N/A	
15 mins	142329	10.03%	96	130.6	0.00%	
PLR Change	142325	10.03%	60	55.35	57.62%	6%
Δh	142005	10.24%	82	76.38	41.52%	36 kJ/kg
T_{apr}	142710	9.79%	72	68.7	47.40%	5.5 ºC

$Freq_{ct,fan}$	142305	10.05%	108	145.1	-11.10%	36
$M_{w,prm,HX}$ Change	142266	10.07%	30	28.74	77.99%	4.7 L/s
$M_{w,sec,pump}$ Change	143277	9.43%	77	71.62	45.16%	18 L/s

For the energy performance, "Δh" obtains the highest energy saving, while the "$M_{w,sec,pump}$ Change" gets the lowest. Four cases are equal to or higher than the TDO benchmark ("15 mins"), while the cases of "T_{apr}" and "$M_{w,sec,pump}$ Change" are below the TDO benchmark.

Regarding the computational performance, except the case of "$Freq_{ct,fan}$", around 40-78% of computation was saved, which shows a good potential of EDO for computation reduction. The case of "$Freq_{ct,fan}$" can be regarded as the frequent event triggering since the optimization times is higher the TDO benchmark ("15 mins").

6.4.4 Performance score and event redundancy analyses

The performance scores of different events are shown in

"$M_{w,prm,HX}$ Change" performed the best since it has the highest performance score (1.394). Thus, "$M_{w,sec,pump}$ Change" in the same group was discarded. In another event group, since "Δh" got the highest performance score, "T_{apr}" and "$Freq_{ct,fan}$"

were not used. Please note that “$\mathrm{Freq}_{\mathrm{ct,fan}}$” got a performance score lower than “1.000” because its computation saving is negative. In summary, “$\mathrm{M}_{\mathrm{w,sec,pump}}$ Change”, “$\mathrm{T}_{\mathrm{apr}}$” and “$\mathrm{Freq}_{\mathrm{ct,fan}}$” are considered as the redundant events, which were discarded to avoid the potential event overlap.

Table 6. 7, where “1.000” was set as the benchmark performance score. “$\Delta\mathrm{h}$”, “$\mathrm{T}_{\mathrm{apr}}$” and “$\mathrm{Freq}_{\mathrm{ct,fan}}$” are grouped into one event group as they indicate the efficiency of heat rejection process in cooling towers (Chang et al. 2015). “$\mathrm{M}_{\mathrm{w,prm,HX}}$ Change” and “$\mathrm{M}_{\mathrm{w,sec,pump}}$ Change” are grouped into one event group as they all reflect the changes in the chilled water loop.

“$\mathrm{M}_{\mathrm{w,prm,HX}}$ Change” performed the best since it has the highest performance score (1.394). Thus, “$\mathrm{M}_{\mathrm{w,sec,pump}}$ Change” in the same group was discarded. In another event group, since “$\Delta\mathrm{h}$” got the highest performance score, “$\mathrm{T}_{\mathrm{apr}}$” and “$\mathrm{Freq}_{\mathrm{ct,fan}}$”
were not used. Please note that “$\mathrm{Freq}_{\mathrm{ct,fan}}$” got a performance score lower than “1.000” because its computation saving is negative. In summary, “$\mathrm{M}_{\mathrm{w,sec,pump}}$ Change”, “$\mathrm{T}_{\mathrm{apr}}$” and “$\mathrm{Freq}_{\mathrm{ct,fan}}$” are considered as the redundant events, which were discarded to avoid the potential event overlap.

Table 6. 7 Performance scores of different events ("PS"= "Performance score")

Event name	Threshold	PS
PLR Change	6%	1.288
Δh	36 kJ/kg	1.228
T_{apr}	5.5 ℃	1.213
$Freq_{ct,fan}$	36	0.946
$M_{w,prm,HX}$ Change	4.7 L/s	1.394
$M_{w,sec,pump}$ Change	18 L/s	1.166

6.4.5 Validation: optimization performance of established event space

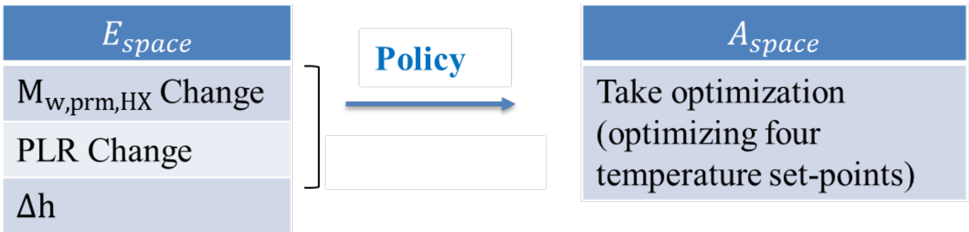

Figure 6. 9 EDO policy

Table 6. 8 Optimization performance of different methods ("Op." = "Optimization")

Op. methods	ES	CS	PS
No Op.	0.00%	\	
15mins	10.03%	0.00%	1.000
$M_{w,prm,HX}$ Change	10.07%	77.99%	1.394

PLR Change	10.03%	57.62%	1.288
Δh	10.24%	41.52%	1.228
Multiple events	11.01%	62.75%	1.411

(Please note: "multiple events" contains "$M_{w,prm,HX}$ Change", "PLR Change" and "Δh"; ; threshold of "PLR Change" is 6%; threshold of "Δh" is 36 kJ/kg; threshold of "$M_{w,prm,HX}$ Change" is 4.7 L/s.)

To validate the identified event space, this section presents the optimization performance using multiple events. Based on the performance analyses of different events presented in the previous sections, following event space and EDO policy are established (Figure 6. 9). Please note that only one default action is adopted, i.e. if any of events in the event space happened, the action will be taken to optimize all the decision variables.

The performances of "multiple events" and single event are listed in Table 6. 8. Results show that, compared with TDO benchmark("15mins") or single event, the performance score of "multiple events" is the highest. It also shows that, by adopting the events discovered by the RF algorithm, the energy saving of EDO (11.01%) is almost 1% higher than the TDO benchmark. Meanwhile, 62.75% of computation was saved.

6.4.6 Discussions

In this case study, the effectiveness of the algorithm of the data-based method for state transition identification is validated. The optimization performances of events identified from operational data are evaluated. The findings are summarized as follows.

- The proposed index, Euclidean distance of decision variable vectors, is an effective estimator of the optimization reward. The optimization reward can be quickly estimated from the operational data by establishing the function between the Euclidean distance of decision variable vectors and state transitions. Important state transitions can be identified by evaluating the variable importance using the random forest algorithm.

- Suitable threshold values can be directly computed based on the identified "significant" optimization reward using the Euclidean distance of decision variable vectors, which avoids the tedious trial-and-error method used in the knowledge-based method.

- Compared with the knowledge-based method, new events are found in the operational data, namely "$\mathrm{Freq}_{\mathrm{ct,fan}}$", "$\mathrm{M}_{\mathrm{w,prm,pump}}$" and "$\mathrm{M}_{\mathrm{w,sec,pump}}$".

- The overall optimization performances (including energy and computational performances) of the discovered events are all better than the TDO benchmark, except the case of frequent event triggering in "$\mathrm{Freq}_{\mathrm{ct,fan}}$". It is found that the bad threshold selection would increase the computation as shown by the case of "$\mathrm{Freq}_{\mathrm{ct,fan}}$".

- It is confirmed again that, compared with using the single event, a better optimization performance can be obtained when multiple critical events are properly defined and used.

- A limitation of the proposed data-based method is that the HVAC system should have the RTO function so that the operation data will contain information about the decision variables, though the detailed simulation can be used as an alternative.

6.5 Summary

Instead of using the knowledge-based method, this chapter has effectively explored another indirect method for event space establishment, i.e. the data-based method. An algorithm of the data-based method for state transition identification in operational data is proposed, and the effectiveness of the algorithm is demonstrated through case studies. The Euclidean distance of decision variable vectors is proposed to estimate the optimization reward, which is represented as a function of state transitions. The variable importance is calculated by the random forest algorithm, based on which important state transitions are identified. New events are found from the operational data comparing with from the prior knowledge. Results show that, using the discovered events, the overall energy and computational performances are both improved compared with the TDO benchmark, except a case of frequent event triggering caused by the unsuitable threshold selection.

With the data-based method, users can easily select state transitions based on the variable importance instead of domain knowledge. Besides, the event threshold selection is convenient comparing with the trial-and-error method. Moreover, the data mining technique makes it possible to customize the event space for the targeted system. For instance, the events and event thresholds can be customized for the targeted system based on the system operational data. In this way, the formulated EDO policy can handle the HVAC optimal control more precisely. Based on the above findings, we conclude that the data-based method can improve the practicability of EDO since only minor effort is required in terms of finding important events and suitable event thresholds. The data-based method is also a good supplementary of the knowledge-based method when human prior knowledge is inadequate.

CHAPTER 7. DIRECT METHOD FOR EVENT SPACE ESTABLISHMENT BASED ON SCOP DEVIATION

This chapter presents the direct method for event space establishment. The SCOP-deviation-based method is used to directly emulate the optimization objective. Section 7.1 illustrates the transient and accumulated SCOP deviations, and presents the equations for calculations as well as the event definitions based on COP·mins. Section 7.2 introduces the method for finding the reference SCOP based on simulation, where the artificial neural network is utilized. The case study is presented in Section 7.3 to illustrate the performance of the SCOP-deviation-based method. At last, a summary is given in Section 7.4.

7.1 Direct Method for State Transition Identification

The direct method for event space establishment refers to the direct emulation of the optimization objective(s) (Miskowicz, Lunze 2015). The principle of the direct emulation is simple: it triggers an action when the deviation between the current system state value and the reference value exceeds a defined threshold. The basic form of the event triggering condition is shown in Eqn. (7. 1), where y is a state variable and σ_y is a user-defined threshold. With the direct emulation, the EDO actually mimics the desirable performance trajectory (y_{ref}) with controllable precision; while the cost of the improvement is to spend more resources (such as computation, communication or energy).

$$\Delta y = | y_{ref} - y \ | < \sigma_{\mathrm{y}} \tag{7. 1}$$

7.1.1 SCOP-deviation-based method

Two examples are shown in Figure 7. 1 to illustrate the idea of SCOP-deviation-based method. Assume that desired system operation efficiency is known; σ_{trans} is the threshold for the transient SCOP (Yao et al. 2004) deviation and σ_{accum} is the threshold for the accumulated SCOP deviation. In the "case a" of Figure 7. 1, the event will be triggered between time k and l because the transient SCOP deviation is larger than the threshold σ_{trans}. However, this is not sufficient as there will be cases that the $\Delta SCOP(\mathrm{i})$ is always lower than σ_{trans}. As shown in "case b" of Figure 7. 1, although the σ_{trans} is satisfied all the time, the accumulated SCOP deviation cannot be neglected when the time period is significant. This is confirmed by the ASHRAE handbook (ASHRAE 2015), which states that "optimizations of the plant operation are most important when loads vary and when the system operation is far from the design condition for a significant period".

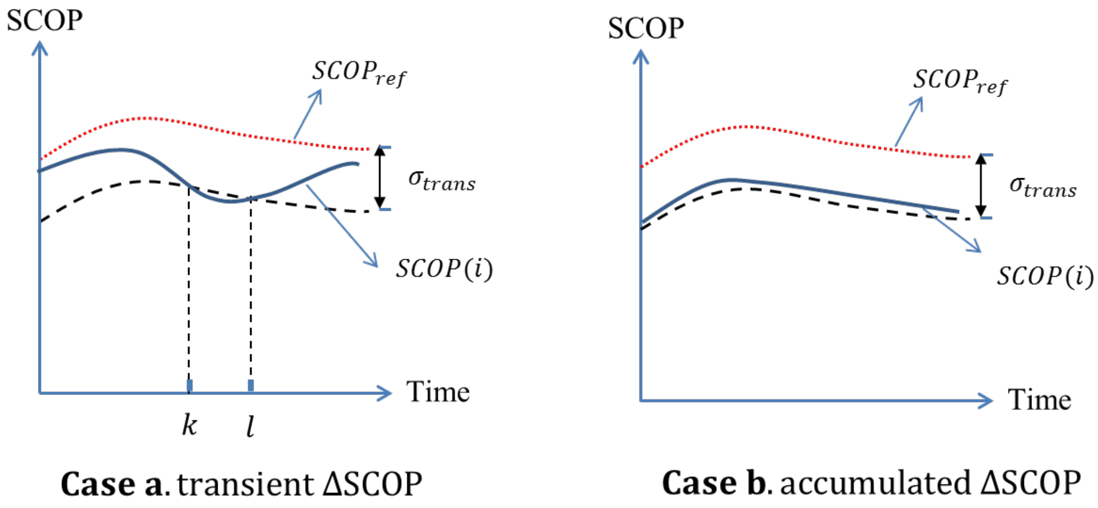

Figure 7. 1 Illustrate two forms of SCOP deviation

$$\Delta SCOP(i) = SCOP_{ref}(i) - SCOP(i) < \sigma_{\text{trans}} \tag{7. 2}$$

$$\Delta SCOP(j-k) = \int_j^k \Delta SCOP(i)dt < \sigma_{accum} \tag{7. 3}$$

Thus, both the transient and accumulated SCOP deviations are defined in Eqn. (7. 2) and (7. 3), where "i, j, k" are time instants. The accumulated SCOP deviation is represented as the integration of $\Delta \mathrm{SCOP(i)}$ over time. As long as the Eqn. (7. 2) or (7. 3) is violated, an event will be triggered, and the optimization will be performed. It is assumed that every optimization can bring the current SCOP to the desired value. Thus, a large SCOP deviation ($\Delta SCOP$) means a large optimization reward. The threshold is defined by users, and a smaller threshold can provide a better emulation of the desired SCOP, but more actions will be triggered.

7.1.2 COP·mins

Considering the discrete nature of the measurement in BAS (by sampling), the term "COP·mins" is proposed for the calculation convenience of the integration in Eqn. (7. 3). As the name indicates, one minute is chosen to be the discretization time interval, which is small enough in HVAC RTO problems. Consequently, Eqns. (7. 2) and (7. 3) can be rewritten as follows.

$$\mathrm{COP}\cdot\mathrm{mins}(i) = \Delta SCOP(i) = SCOP_{ref}(i) - SCOP(i) \tag{7. 4}$$

$$\mathrm{COP}\cdot\mathrm{mins}(j-k) = \Delta SCOP(j-k) = \int_j^k \Delta SCOP(i)dt \approx \sum\nolimits_{i=j}^{k} \Delta SCOP(i) \tag{7. 5}$$

To unify the notion, the transient SCOP deviation ("$\Delta\mathrm{SCOP}(\mathrm{i})$") is represented by "$\mathrm{COP}\cdot\mathrm{mins}(\mathrm{i})$" as shown in Eqn. (7. 4). The accumulated SCOP deviation from time k to l (i.e. "$\Delta\mathrm{SCOP}(\mathrm{j}-\mathrm{k})$") is approximated by the summation of "$\Delta\mathrm{SCOP}(\mathrm{i})$", which is denoted as "$\mathrm{COP}\cdot\mathrm{mins}(\mathrm{j}-\mathrm{k})$" (as shown in Eqn. (7. 5)). The "COP·mins" defined above is used to define the events, and the mathematical representations of the two events are given in **Error! Not a valid bookmark self-reference.**.

Table 7. 1 Mathematical representations of COP·mins

Event name	**Definition**
$\mathrm{COP} \cdot \mathrm{mins(trans)}$	$e_{\mathrm{COP \cdot mins(trans)}} := \{< X_{\tau_i}, X_{\tau_j} > \mid \mathrm{COP} \cdot \mathrm{mins(i)} \leq \sigma_{trans}, \mathrm{COP} \cdot \mathrm{mins(j)} > \sigma_{trans}\}$
$\mathrm{COP} \cdot \mathrm{mins(accum.)}$	$e_{\mathrm{COP \cdot mins(accum.)}} := \{< X_{\tau_i}, X_{\tau_j} > \mid \mathrm{COP} \cdot \mathrm{mins(i-j)} \leq \sigma_{accum}, \mathrm{COP} \cdot \mathrm{mins(i-j)} > \sigma_{accum}\}$

(Note: “trans”= “transient”; “accum”= “accumulated”; “j > i”.)

7.2 Finding the Reference SCOP by Simulation

To compute the SCOP deviation ($\Delta SCOP(\mathrm{i})$), the reference SCOP model is necessary. Here, the maximum SCOP is considered as the reference SCOP because the objective is to maximize the operating efficiency. There are several ways to get the maximum SCOP model. For instance, the SCOP curve of a chiller can be obtained by manufactures’ data or from the curve fitting of in-situ performance data (ASHRAE 2015). Typically, the SCOP can be represented as a function of PLR and other necessary variables (such as the condenser and evaporator water temperature).

For simplicity, a simulation-based method was adopted to generate the data. The maximum SCOP model was obtained by applying an extremely high optimization frequency (i.e. one optimization per 5 minutes) in the simulation for a given a load profile. In the regression, SCOP is represented as a function of nine critical variables (Yu et al. 2017) as shown in Eqn. (7. 6).

$$SCOP_{ref} = f\left(PLR, T_{db}, T_{wb}, \Delta h, T_{apr}, Num_{ch}, Freq_{ct,fan}, T_{scw}, T_{schw,prm}\right) \tag{7. 6}$$

The reference SCOP model is obtained using the artificial neural network (ANN) which can produce satisfactory predictions for complex non-liner relationships (Li et al. 2009). In this study, a feed-forward neural network of 3 hidden layers were used as given in Figure 7. 2, which is able to give satisfactory accuracy for the given data.

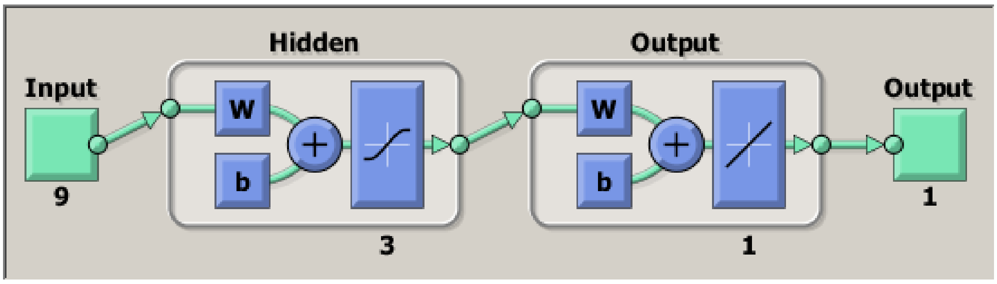

Figure 7. 2 Artificial neural network structure

7.3 Case Study

7.3.1 Load and weather profiles

Load and weather data are shown in Table 7. 2, and their profiles are shown in Figure 7. 3 and Figure 7. 4.

Table 7. 2 Load and weather data (in the year of 2013)

Case	Date	Load (kW)			T_{db} (˚C)			T_{wb} (˚C)		
		mean	max	min	mean	max	min	mean	max	min
Summer	Aug-20	13412	21463	3817	28.9	30.8	27	26.5	27.6	25.7
Spring	Apr-2	10669	15936	3757	21.1	22.5	20	20.3	21.5	18.8

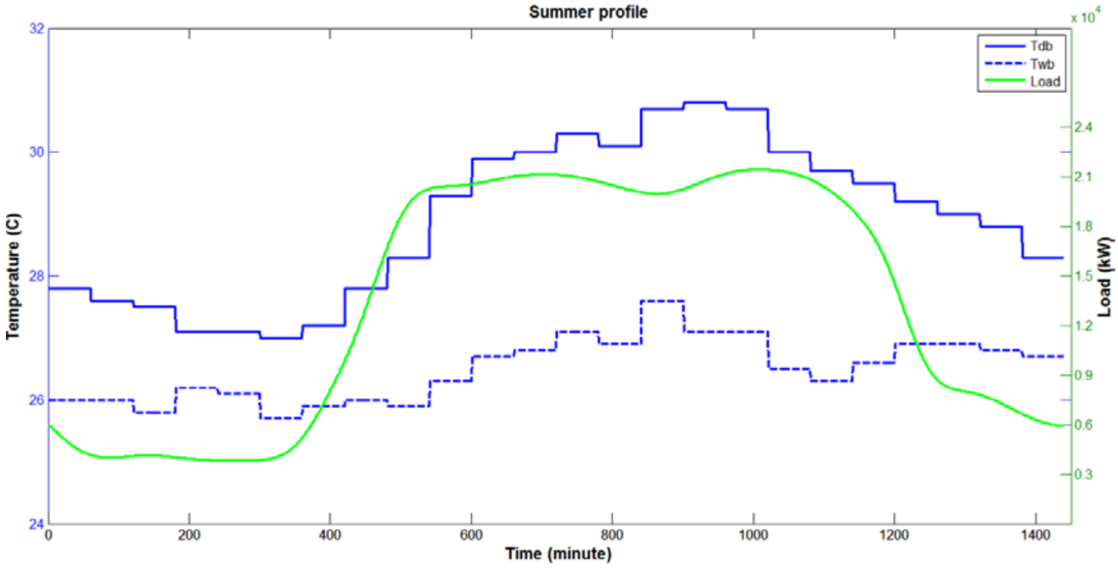

Figure 7. 3 Summer load and weather profile (Aug-20)

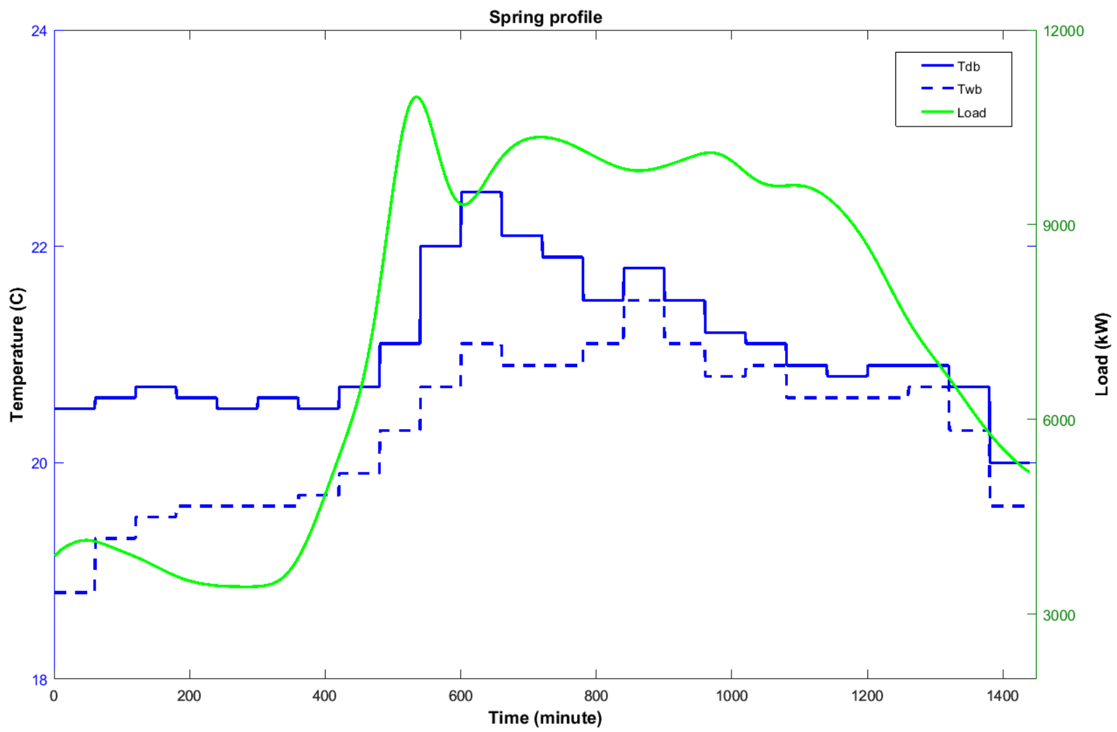

Figure 7. 4 Spring load and weather profile (Apr-2)

7.3.2 Establish the maximum SCOP model by ANN

The ANN model was developed based on the Levenberg-Marquardt backpropagation algorithm in MATLAB ANN toolbox (Beale, Hagan & Demuth 2017), and the detailed code can be found in Appendix E. The code was used to simulate the trained ANN in order to generate the reference SCOP model.

The summer and spring cases were trained separately. There are totally 2880 samples for each case, where 70% of the data was used for training, 15% was used for validation and 15% was used for testing. The regression R values are shown in Figure 7. 5 and Figure 7. 6. Regression R values (or coefficient of correlation) measure the correlation between model outputs and targets (real values). An R value of “1” means a close relationship, while “0” means a random relationship. Both cases got R values above 0.9, which shows good prediction performance.

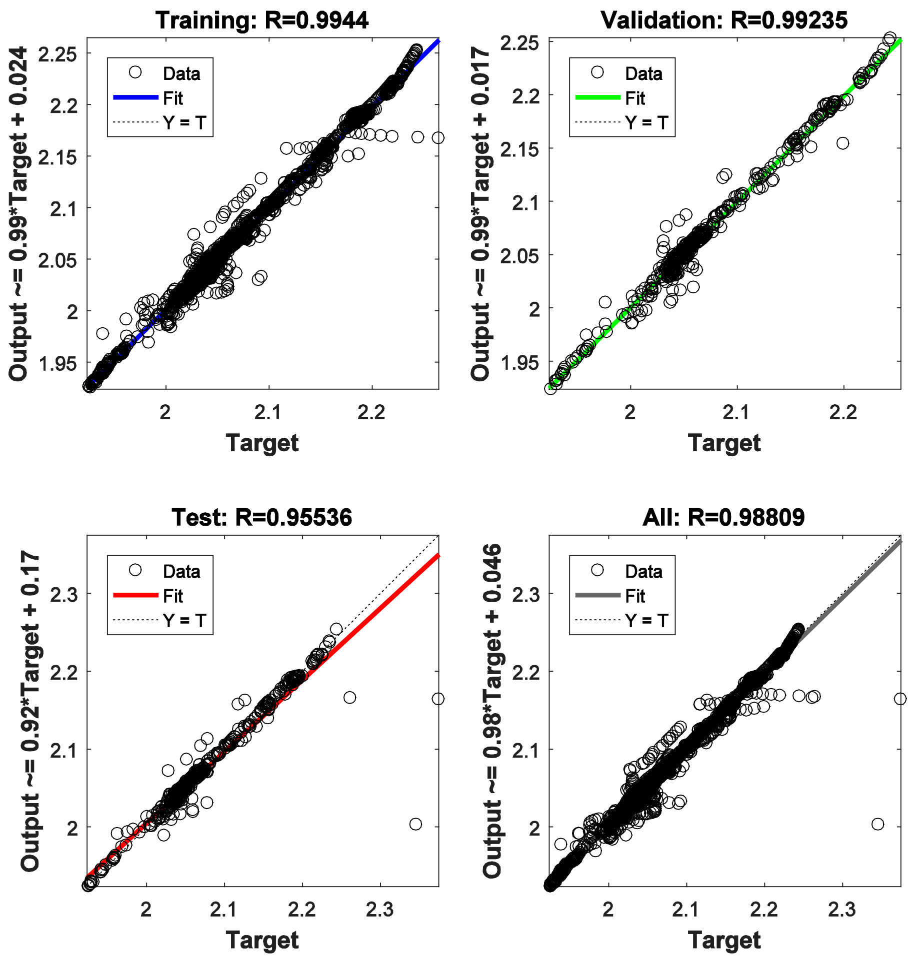

Figure 7. 5 Regression R values of spring case

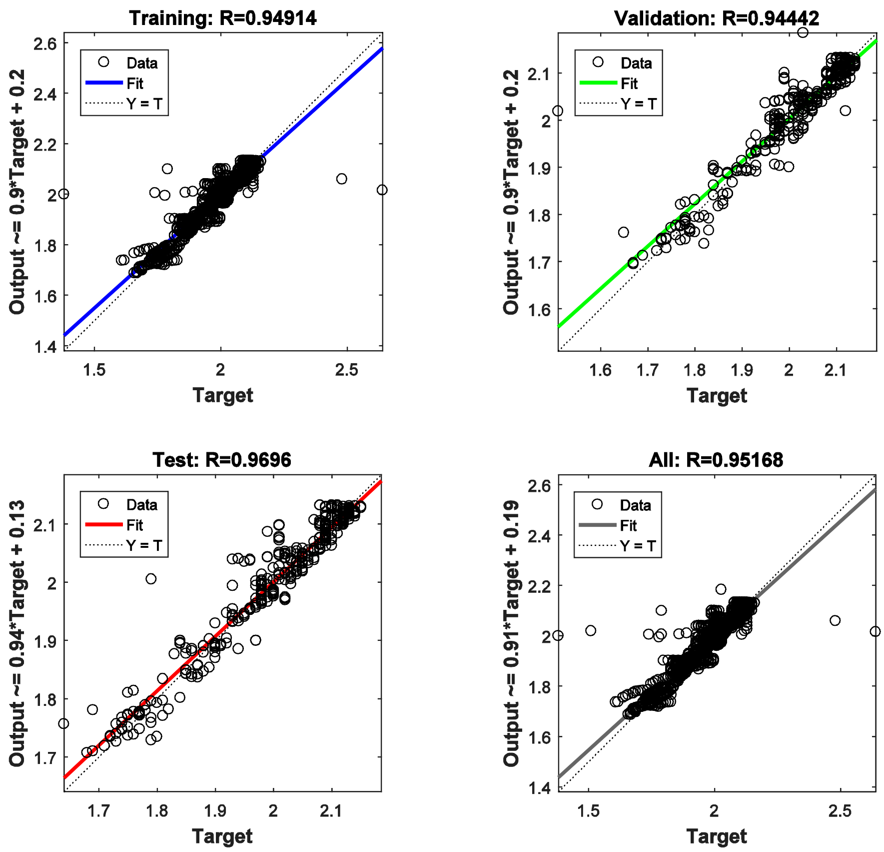

Figure 7. 6 Regression R values of summer case

7.3.3 Event threshold selection for COP·mins

To select the threshold value for the events of COP·mins, simple calculations were performed based on $SCOP_{ref}$. The power consumption difference at each time instant ($\Delta Power(i)$) that results from the SCOP deviation ($\Delta SCOP(i)$) can be calculated by Eqn. (7. 7). Substituting Eqn. (7. 8) into Eqn. (7. 7), $\Delta SCOP(i)$ can be computed by specifying the $\Delta Power(i)$ using Eqn. (7. 9). This can be used as the threshold for the event "$\mathrm{COP} \cdot \mathrm{mins(trans.)}$".

$$\Delta Power(i) = load(i) / SCOP(i) - load(i) / SCOP_{ref} \tag{7. 7}$$

$$\Delta SCOP(i) = SCOP_{ref} - SCOP(i) \tag{7. 8}$$

$$\Delta SCOP(i) = \Delta Power(i) \times SCOP_{ref}^{\ 2} \ / \ (load(i) + \Delta Power(i) \times SCOP_{ref}) \tag{7. 9}$$

Details for the threshold calculation is shown in Table 7. 3. Based on the simulation data, the means of $SCOP_{ref}$ are 2.26 and 2.46 in summer and spring cases respectively. Please note that the maximum SCOP model was obtained by applying a high optimization frequency (i.e. "one optimization per 5 minutes") in the simulation.

$\mathrm{Load_{mean}}$ is obtained from the Table 7. 2. $\mathrm{EC_{BC}}$ is the base case energy consumption, in which no optimization is taken. $\mathrm{Power_{mean}}$ is computed by $\frac{\mathrm{EC_{BC}}}{24\ hours}$, which is the average power consumption. 10% of the base case $\mathrm{Power_{mean}}$ is selected as a significant portion, which is represented by ΔPower in Table 7. 3. Based on Eqn. (7. 9), the significant SCOP deviation (ΔSCOP) is 0.255 and 0.172 corresponding to 10% power consumption difference in summer and spring cases. The average value (i.e. 0.214) is taken as the $\sigma_{trans.}$.

The threshold of "$\mathrm{COP} \cdot \mathrm{mins(accum.)}$" was obtained by taking half of the $\sigma_{trans.}$ in Table 7. 3 (i.e. $0.214 \div 2 = 0.107$), and a duration of 30 minutes was considered as a significant period. Thus, the “$\mathrm{COP} \cdot \mathrm{mins(accum.)}$” is 3.21 which will be used as the threshold ($\sigma_{accum.}$). Please note that the values were selected arbitrarily in this case study since the main purpose of this study is to demonstrate the effectiveness of the proposed COP·mins. Other threshold values can be discussed in the future when necessary.

Table 7. 3 Details for threshold calculation

Variable name	**Summer**	**Spring**
$\mathrm{Load_{mean}}$	13412	10669
$\mathrm{EC_{BC}}$ (kWh)	181379	78241
$\mathrm{Power_{mean}}$ (kW)	7557 (= $\frac{\mathrm{EC_{BC}}}{24\ hours}$)	3260
Portion	10%	10%
$\Delta\mathrm{Power}$ (kW)	756	326
$\mathrm{SCOP_{ref}}$	2.26	2.46
$\Delta\mathrm{SCOP}$	0.255	0.172
$\sigma_{trans.}$	0.214	

7.3.4 Energy and computational performances

Table 7. 4 Optimization performance of different methods (Op. = Optimization)

Op. methods	**EC (Kwh)**	**ES**	**Op. times**	**CT(s)**	**CS**	**Event threshold**
			Spring case			
No Op.	78241	0.00%	0	0	\	
15 mins	74361	4.96%	96	69.83	0.00%	
Ch. On/Off, PLR Change & Δh	73765	5.72%	18	17.44	75.03%	11%; 27kJ/kg
COP·mins	72812	6.94%	12	14.57	79.14%	0.214;3.21
			Summer case			
No Op.	181379	0.00%	0	0	\	
15 mins	161608	10.90%	96	138.4	0.00%	
Ch. On/Off, PLR Change & Δh	161521	10.95%	43	62.26	55.01%	11%; 27kJ/kg
COP·mins	160484	11.52%	16	26.27	81.02%	0.214;3.21

As shown in Table 7. 4, comparing with the TDO method, the EDO using multiple events can achieve a better energy saving (5.72% and 10.95%) together with a considerable computational reduction (75.03% and 55.01%). This agrees well with the previous findings as presented in Chapter 5 and 6. However, choosing the threshold is difficult because the threshold does not have a direct relationship with the optimization reward.

COP·mins-based EDO method achieves 6.94% and 11.52% of energy savings, 79.14% and 81.02% of computation savings in spring and summer cases respectively, which are all higher than the EDO method using multiple events "Ch. On/Off, PLR Change and Δh". The main reason is that COP·mins is a direct performance indicator that can reflect the optimization reward in a more precise way, and thus leads to a more efficient manipulation of the optimization.

7.3.5 Discussions

The proposed SCOP-deviation-based method can mimic the optimization objective directly. Results show that the SCOP-deviation-based EDO performs better than the multiple events ("Ch. On/Off, PLR Change and Δh") regarding both energy and computational efficiencies. Besides, users can easily select thresholds according to their requirements since the relationship between the SCOP deviation and optimization reward is clear. For instance, if the computation resource is limited in a system, a large threshold should be selected to prevent frequent optimization triggering which would consume a lot of computation. If the computation resource is sufficient, a small threshold can be used to achieve better emulation of the objective. However, if an indirect index (e.g. PLR) is used, this may not be easy since the relationship between the threshold and optimization performance is unclear.

7.4 Summary

This chapter presents a direct method for event space establishment. The event is defined based on the SCOP deviation which can estimate the optimization reward directly in either transient or accumulated form. COP·mins is proposed for the convenience of calculation for SCOP deviations. Results show that the direct emulation of the objective is more efficient than the indirect methods concerning the energy and computational performances. As both transient and accumulated performance deviation are defined, the robustness can be ensured for different load types, e.g. sharp load profiles or slow-change load profiles. In terms of the implementation, the threshold selection can be easily done. Meanwhile, the flexibility is offered by the controllable threshold. Users can decide the balance between energy and computation efficiencies by adjusting the thresholds. However, the applicability of the direct method depends on whether the explicit model of optimization objective (e.g. SCOP) is available, which restricts its applications. The developed SCOP-deviation-based method is general and can be applied in other systems.

CHAPTER 8. CONCLUSIONS AND FUTURE WORK

8.1 Summary

Real-time optimization (RTO) is regarded as an effective way for improving energy-efficient operations of HVAC systems. However, the growing complexity of HVAC systems makes the conventional TDO mechanism of RTO no longer efficient due to the postponed or unnecessary optimization actions when responding to stochastic state transitions. Consequently, the energy and computational performances cannot be well balanced, which restricts the applications of RTO in complex HVAC systems. Therefore, it is necessary to develop a more efficient optimization strategy so as to better balance the energy and computational performances and proliferate the applications of RTO. This thesis addresses this need through establishing an EDO framework and developing a design approach for event space establishment. The main contents and findings are summarized as follows.

8.1.1 EDO framework

To facilitate the event-driven mechanism for RTO in HVAC systems, an EDO framework was established. A control diagram of EDO was firstly presented, and the working principle of the EDO was explained by the {event, policy, action} structure.

An event describes a set of state transitions that happen instantly or continuously in a period of time. For HVAC systems, three basic event attributes were synthesized, including timestamp, descriptive state variable and threshold. Each attribute was divided into two categories. Thus, in total, events were divided into eight total event types based on event attributes. To facilitate the real-world implementations of EDO, events were express in mathematical forms. The event identification process was also presented, which is used to recognize the occurrence of the events.

The simulation results (in Chapter 5, 6 and 7) show that the EDO could reduce the computational load while still respecting the energy efficiency in comparison with the TDO. The reason is that EDO is capable of adapting to the changing environment. Thus, the responses to state transitions are quicker and unnecessary optimization actions can be avoided. Indeed, compared with the TDO benchmark, the formulated EDO can achieve even higher energy efficiencies with lower computational load in most of the cases. Thus, the EDO is suggested as a good alternative to the TDO, especially when the system dynamics are highly stochastic and difficult to predict.

8.1.2 EDO design

The essence of the EDO design problem is to capture the most critical state transitions and properly define them into events for the policy formulation. To ensure the EDO can achieve the satisfactory optimization performance, a design approach for EDO was developed. Firstly, a five-step design procedure was introduced, which mainly deals with the tasks of designing the structure of {event, policy, action}. Special attention was paid to the event space due to its dominant role in EDO. The methodology of event space establishment was developed to address the problems of the state transition identification, event definition and event space optimization. Both direct and indirect methods were established in order to identify critical state transitions under different situations. The direct method was constructed based on the SCOP deviation while two indirect methods were developed based on prior-knowledge and operational data. The relationships of these methods and their optimization performances are summarized separately as below.

- Direct method and its performance

To directly emulate the optimization objective (i.e. SCOP), a direct method was developed based on the SCOP deviation. Whenever the current SCOP deviates away from the reference level for a certain quantity (transient or accumulated), an event will be triggered, and actions will be taken to optimize the decision variables. “COP·mins” was proposed for the calculation convenience of the accumulated SCOP deviation. Results show that the SCOP-deviation-based direct method is superior to the TDO method and indirect methods concerning both the energy and computation savings. Meanwhile, the robustness can be ensured for different load types, e.g. steep load profiles or slow-change load profiles, since both transient and accumulated SCOP deviation are captured. Because the relationship between the event threshold and optimization objective is clear, the threshold selection can be easily done. Moreover, the flexibility is offered by the controllable threshold. Users can decide the balance between energy and computation efficiencies by adjusting the thresholds. However, the applicability of the proposed direct method depends on whether the explicit model of the optimization objective (e.g. SCOP) is available.

➢ Indirect methods

(1) Knowledge-based method and its performance

To utilize the well-developed prior knowledge for identifying important state transitions, a knowledge-based method was developed. The possible state transitions were found from the engineering handbooks. Based on the literature review, critical state transitions were identified from the candidates, which were defined as events. In order to select the suitable thresholds of events, an algorithm was also developed for continuous-form state variables.

Each event was assigned with several candidate thresholds, and tested based on the typical load and weather profiles. The optimization performances of events were analyzed in a comprehensive manner regarding the energy saving, computation saving and performance score. The best event and event threshold were identified in each event source based on the optimization performance analyses. Aa a result, insignificant events were discarded, and the event space was established. The results validate that the formulated EDO policy (using the established event space) can effectively reduce the computational load and further increase the energy saving. The reasons are that the EDO can reduce the optimization action delay and avoid unnecessary optimization actions. It has been demonstrated that the proposed knowledge-based method can be an effective way for event space establishment. However, the knowledge-based method only has qualitative evaluations upon the state transitions. The event and threshold selections may require extensive efforts, and sometimes are tedious. Besides, only general events can be found.

(2) Data-based method and its performance

When the well-developed knowledge is not available, the data-based method can be used since the building operational data contains meaningful information about the building operational patterns. The random forest algorithm was adopted to compute the variable importance corresponding to the optimization reward. The optimization reward was estimated by the Euclidean distance of adjacent decision variable vectors in a quantitative way. Some new events that are not discovered using the prior knowledge can be found from the operational data. Results show that, using the discovered events, the overall energy and computational performances are both higher than the TDO benchmark. The data-based method enables the easy selections of state transitions and event thresholds. Additionally, the data mining technique also makes the event space customizable for the targeted system. Thus, the data-based method can be a good supplement for the knowledge-based method when human prior knowledge is limited. The limitation of this method is that it requires enough operational data.

In summary, three methods were developed for state transition identification, which can be used in different situations. These methods are useful to improve the performance of EDO designs. The direct method, developed based on the SCOP deviation, is simple, effective and controllable. However, it requires the explicit model of the optimization objective, which may not be available in most of the complex HVAC systems. When the direct method cannot be used, the indirect methods can be used as alternatives. The knowledge-based method can be used to define general events for typical HVAC systems when the prior knowledge is applicable. The data-based method is applicable in any systems for which the operational data are available and allows events to be customized for the targeted system.

8.2 Conclusions: Main Contributions of Thesis

The presented thesis makes the following main contributions.

(1) The EDO framework is established for HVAC systems, which provides the infrastructure for the realization of the EDO. Event attributes are synthesized and event types are categorized for the first time in HVAC systems, which can help people to understand the nature of events. The developed EDO framework provides a good alternative to the traditional TDO mechanism when facing with stochastic system dynamics and enriches the HVAC optimal control techniques.

(2) An EDO design procedure is proposed to establish the {event, policy, action} structure. The methodology for event space establishment is developed, and three

methods are developed to identify important state transitions from different aspects. These methods enable much convenience and effectiveness in selecting suitable events and event thresholds for different situations of practical HVAC applications. It is also beneficial for engineers or building operators to identify key operational patterns of the system from the event-driven point of view. The contributions of the individual method are presented as below.

- The SCOP-deviation-based method can mimic the desired state trajectory directly by capturing both transient and accumulated forms of SCOP deviations. It is effective, controllable and simple to implement. The robustness can also be ensured for different load types, e.g. steep load profiles or slow-change load profiles. The limitation is that it requires the explicit model of the optimization objective. However, it is still a general method and can be applicable in other industrial systems where SCOP is the main concern.

- A knowledge-based method utilizes the prior knowledge to identify important state transitions. Since no numerical techniques are involved, the knowledge-based method is straightforward and easy to implement when prior knowledge is available. In the case study, the general events that can be used in the EDO are identified. These events are useful for building operators to construct the HVAC operation strategies in an event-driven manner.

- A data-based method is proposed to explore the building operational data for state transition identification. The Euclidean distance of decision variable vectors is proposed as a basis for estimating the corresponding optimization reward. The data-based method enables the discovery of new events and the easy identifications of state transitions and event thresholds. In addition, it also makes the event space customizable for the targeted system, which is beneficial for further improvement of the optimization performance of EDO. The data-based method is a good supplement for the knowledge-based method when the human prior knowledge is inadequate. Apart from its use in EDO, the data-based method can help the users or building operators to identify the key state transitions, and thus improving the existing operation strategies. The developed data-based method requires minor efforts and would have more applications with the increasing amount of building operational data.

(3) Comprehensive optimization performance analyses of typical events in HVAC systems are performed. General events are identified and studied in a typical HVAC

system. These results can be used to guide the design of EDO in typical HVAC systems. Some guidelines are summarized here.

- Important events in typical HVAC systems are: chiller sequence change, part-load-ratio change, the average enthalpy difference between the specific saturated (at inlet and outlet cooling water temperature) and bulk air, chilled-water mass flow rate change. These events can be used and extended to improve the operating efficiencies of similar systems.

- The ill-designed EDO can increase the computation or deteriorate the energy efficiency, especially when the operational condition is extreme or fluctuating. Thus, events and event thresholds need to be carefully selected in the design stage in order to prevent the unfavorable optimization actions, such as the frequent event triggering. Prior to the implementation of the EDO design, it is also necessary to evaluate the performance in different aspects, such as the energy efficiency, computational efficiency and robustness under different event thresholds or operation scenarios.

- Using multiple events often have a better optimization performance than using single event since more critical state transitions are captured. However, the potential event duplications should be avoided since it is unnecessary to repeat the optimization actions.

8.3 Limitations and Future Work

The limitations and possible future work are summarized as follows.

(1) It should be noted that the validation of the presented work is limited to a simulated case system, in which the air-side system is simplified. The identified events come from the system and operational environment, while the human side is not studied. More sophisticated systems and events from occupants need be investigated in the future. In addition, the in-situ implementation and validation of the developed EDO strategy should be performed.

(2) "Local events" can be used. For instance, if one event is observed at one building zone, it may not be necessary to optimize the entire system. Probably, an action at the corresponding local system will solve the optimal control problem very well, which also saves computation and reduces the fluctuations. Actually, this idea originates from the study by Wu, Jia and Guan (2015) in which they call it the local-event-based

approach. The approach they developed is still simple, and thus further investigations can be done in the future.

(3) This thesis does not answer the second fundamental question in Chapter 2, i.e. the selection of the optimization action. To simplify the problem, only the default action is used in the case study, i.e. optimization all the decision variables. However, the choice of decision variables can also be optimized according to the operational conditions. It is beneficial to allow some degree of freedom in the decision variable resetting, i.e. relax the optimization action. For instance, assume there are four decision variables. In principle, all the four decision variables should be updated at the same time to make sure the performance is optimal. However, in the premise of not deteriorating the optimization performance, two or three decision variables can be updated while the others are kept unchanged. The direct benefit is that the required computation is much less than updating all the four decision variables due to the reduced dimension of the search space. Meanwhile, the response time can be quicker, which is beneficial to the optimization performance. How to balance the performance decrease result from the relaxed optimization action and the performance increase result from the quicker response is an interesting question. Thus, it is worthwhile to investigate the optimal selection of the optimization action in the future.

(4) The event threshold can be a fixed value or a variable value. In the case study, the event threshold is predefined and keeps unchanged in the operation period as the main purpose is to demonstrate the methodology. It is interesting to study the adaptive event threshold as a variable threshold could be more adaptable than a fixed threshold in a dynamic environment. The selection of event threshold is critical as bad selections it may cause unstable performance. For instance, when the state variable fluctuate and the threshold is defined near the fluctuating point, the system may become unstable due to the frequent event triggering and optimization. This should be carefully prevented at the design stage.

(5) As there will be cases that some events happen physically, but the system may not be able to observe. For these unobservable events, how to make the reliable inference based on observed data is an interesting issue. Besides, in principle, different event types should have different event identification methods. This is an implementation issue and is worthwhile to investigate in the future.

(6) It is possible that several events will happen simultaneously in a complex system.

For instance, considering a complex building energy system incorporating with renewable energy systems and power grids, if several events happened at the same time, how to react to simultaneous events would be a challenging problem. One way is to execute the action based on the event priorities. Higher event priority can be given to those events with “higher” optimization reward.

With the wide use of BASs, the realization of EDO is not difficult provided that the events, actions and policy are well identified and suitably defined. While this thesis has established an EDO framework and addressed the critical part (i.e. event space establishment), there are still many open problems in the field of EDO. Further research efforts are required to enrich the research of EDO and improve its practicability in HVAC systems.

REFERENCES

A˚ström, K.J. & Kumar, P.R. 2014, "Control: A perspective", *Automatica,* vol. 50, no. 1, pp. 3-43.

Abou-Ziyan, H.Z. & Alajmi, A.F. 2014, "Effect of load-sharing operation strategy on the aggregate performance of existed multiple-chiller systems", *Applied Energy,* vol. 135, pp. 329-338.

Afram, A. & Janabi-Sharifi, F. 2014, "Review of modeling methods for HVAC systems", *Applied Thermal Engineering,* vol. 67, no. 1, pp. 507-519.

Ahmad, M.W., Mourshed, M., Yuce, B. & Rezgui, Y. 2016, "Computational intelligence techniques for HVAC systems: A review", *Building Simulation*Springer, , pp. 359.

Ahn, B.C. & Mitchell, J.W. 2001, "Optimal control development for chilled water plants using a quadratic representation", *Energy and Buildings,* vol. 33, no. 4, pp. 371-378.

Anta, A. & Tabuada, P. 2010, "To sample or not to sample: Self-triggered control for nonlinear systems", *IEEE Transactions on Automatic Control,* vol. 55, no. 9, pp. 2030-2042.

Araújo, J., Mazo, M., Anta, A., Tabuada, P. & Johansson, K.H. 2014, "System architectures, protocols and algorithms for aperiodic wireless control systems", *IEEE Transactions on Industrial Informatics,* vol. 10, no. 1, pp. 175-184.

Archer, K.J. & Kimes, R.V. 2008, "Empirical characterization of random forest variable importance measures", *Computational Statistics & Data Analysis,* vol. 52, no. 4, pp. 2249-2260.

Arzén, K. 1999, "A simple event-based PID controller", *Proc. 14th IFAC World Congress,* pp. 423.

Asad, H.S., Yuen, R.K.K. & Huang, G. 2017, "Multiplexed real-time optimization of HVAC systems with enhanced control stability", *Applied Energy,* vol. 187, pp. 640-651.

Asad, H.S., Yuen, R.K.K. & Huang, G. 2016, "Degree of freedom based set-point reset scheme for HVAC real-time optimization", *Energy and Buildings,* vol. 128, pp. 349-359.

ASHRAE 2015, "CHAPTER 42 SUPERVISORY CONTROL STRATEGIES AND OPTIMIZATION" in *ASHRAE Handbook - HVAC Applications.*, SI ed edn, USA: ASHRAE Inc., Atlanta.

Aste, N., Manfren, M. & Marenzi, G. 2016, "Building Automation and Control Systems and performance optimization: A framework for analysis", *Renewable and Sustainable Energy Reviews,* .

Aström, K.J. 2008, "Event based control" in *Analysis and design of nonlinear control systems* Springer, , pp. 127-147.

Astrom, K.J. & Bernhardsson, B.M. 2002, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems", *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*IEEE, , pp. 2011.

Beale, M.H., Hagan, M.T. & Demuth, H.B. 2017, *Neural Network Toolbox™ User's Guide,* The MathWorks, Inc., Natick, MA.

Bernhardsson, B. & Aström, K. 1999, "Comparison of periodic and event based sampling for first-order stochastic systems", *Preprints of the 14th IFAC World Congress*.

Braun, J.E. 2014, "**Building Control Systems**" in *Encyclopedia of Systems and Control*, eds. B. John & S. Tariq, Springer London, , pp. 1.

Braun, J.E., Klein, S.A., Beckman, W.A. & Mitchell, J.W. 1989, "Methodologies for optimal control of chilled water systems without storage", *ASHRAE Transactions,* vol. 95, pp. 652-662.

Braun, J. & Diderrich, G. 1990, "Near-optimal control of cooling towers for chilled-water systems", *ASHRAE Transactions (American Society of Heating, Refrigerating and Air-Conditioning Engineers);(United States),* vol. 96, no. CONF-9006117--.

Breiman, L. 2001, "Random forests", *Machine Learning,* vol. 45, no. 1, pp. 5-32.

Breiman, L., Cutler, A., Liaw, A. & Wiener, M. 2015, *Breiman and Cutler's Random Forests for Classification and Regression*.

Bylander, T. 2002, "Estimating generalization error on two-class datasets using out-of-bag estimates", *Machine Learning*, vol. 48, no. 1-3, pp. 287-297.

Cai, J. & Braun, J.E. 2015, "A generalized control heuristic and simplified model predictive control strategy for direct-expansion air-conditioning systems", *Science and Technology for the Built Environment*, vol. 21, no. 6, pp. 773-788.

Cai, J. 2015, *A low cost multi-agent control approach for building energy system management*, Purdue University.

Cao, X., Zhao, Y., Jia, Q. & Zhao, Q. 2013, "An Introduction to Event -Based Optimization: Theory and Applications", *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, , pp. 432-451.

Cao, X. 2007, *Stochastic learning and optimization: a sensitivity-based approach*, Springer.

Cassandras, C.G. 2014, "The event-driven paradigm for control, communication and optimization", *Journal of Control and Decision*, vol. 1, no. 1, pp. 3-17.

Cassandras, C.G. & Lafortune, S. 2009, *Introduction to discrete event systems*, Springer Science & Business Media.

Chang, Y., Chen, W., Lee, C. & Huang, C. 2006, "Simulated annealing based optimal chiller loading for saving energy", *Energy Conversion and Management*, vol. 47, no. 15, pp. 2044-2058.

Chang, C., Shieh, S., Jang, S., Wu, C. & Tsou, Y. 2015, "Energy conservation improvement and ON–OFF switch times reduction for an existing VFD-fan-based cooling tower", *Applied Energy*, vol. 154, pp. 491-499.

Cooling Technology Institute 2016, *STD-201 Thermal Certification*.

Cumali, Z. 1994, "Application of real-time optimization to building systems", *ASHRAE Transactions*, vol. 100, no. 1.

Darby, M.L., Nikolaou, M., Jones, J. & Nicholson, D. 2011, "RTO: An overview and assessment of current practice", *Journal of Process Control,* vol. 21, no. 6, pp. 874-884.

Davidsson, P. & Boman, M. 2005, "Distributed monitoring and control of office buildings by embedded agents", *Information Sciences,* vol. 171, no. 4, pp. 293-307.

Drees, K.H. & Braun, J.E. 1996, "Development and evaluation of a rule-based control strategy for ice storage systems", *HVAC&R Research,* vol. 2, no. 4, pp. 312-334.

El Gamal, A., Nair, C., Prabhakar, B., Uysal-Biyikoglu, E. & Zahedi, S. 2002, "Energy-efficient scheduling of packet transmissions over wireless networks", *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies.*IEEE, , pp. 1773.

Evans, L.C. 2005, "An introduction to mathematical optimal control theory", *Lecture Notes, University of California, Department of Mathematics, Berkeley,* .

Fan, C., Xiao, F., Madsen, H. & Wang, D. 2015, "Temporal knowledge discovery in big BAS data for building energy management", *Energy and Buildings,* vol. 109, pp. 75-89.

Fisk, D. 2013, "Optimising heating system structure using exergy Branch and Bound", *Building Services Engineering Research and Technology,* , pp. 0143624413489891.

Fong, K.F., Hanby, V.I. & Chow, T. 2009, "System optimization for HVAC energy management using the robust evolutionary algorithm", *Applied Thermal Engineering,* vol. 29, no. 11, pp. 2327-2334.

Fong, K.F., Hanby, V.I. & Chow, T. 2006, "HVAC system optimization for energy management by evolutionary programming", *Energy and Buildings,* vol. 38, no. 3, pp. 220-231.

Fülöp, J. 2005, "Introduction to decision making methods", *BDEI-3 workshop, Washington*Citeseer, .

Guan, X., Zhao, Q., Jia, Q.S., Wu, J. & Liu, T. 2016, *Cyber-Physical Energy Systems (in Chinese),* China Science Press, Beijing, China.

Han, J., Pei, J. & Kamber, M. 2011, *Data mining: concepts and techniques,* Elsevier.

Heemels, W., Gorter, R., Van Zijl, A., Van den Bosch, P., Weiland, S., Hendrix, W. & Vonder, M. 1999, "Asynchronous measurement and control: a case study on motor synchronization", *Control Engineering Practice,* vol. 7, no. 12, pp. 1467-1482.

Heemels, W., Johansson, K.H. & Tabuada, P. 2015, "Event-triggered and self-triggered control", *Encyclopedia of Systems and Control,* , pp. 384-391.

Heemels, W., Sandee, J. & Van Den Bosch, P. 2008, "Analysis of event-driven controllers for linear systems", *International Journal of Control,* vol. 81, no. 4, pp. 571-590.

Hendricks, E., Jensen, M., Chevalier, A. & Vesterholm, T. 1994, "Problems in event based engine control", *American Control Conference, 1994*IEEE, , pp. 1585.

Henningsson, T., Johannesson, E. & Cervin, A. 2008, "Sporadic event-based control of first-order linear stochastic systems", *Automatica,* vol. 44, no. 11, pp. 2890-2895.

Hinze, A., Sachs, K. & Buchmann, A. 2009, "Event-based applications and enabling technologies", *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*ACM, , pp. 1.

Hordeski, M.F. 2001, *HVAC control in the new millennium,* CRC Press.

Hosmer Jr, D.W., Lemeshow, S. & Sturdivant, R.X. 2013, *Applied logistic regression,* John Wiley & Sons.

Huang, G., Wang, S. & Sun, Y. 2008, "Enhancing the reliability of chiller control using fused measurement of building cooling load", *HVAC&R Research,* vol. 14, no. 6, pp. 941-958.

Huang, S. & Zuo, W. 2014, "Optimization of the water-cooled chiller plant system operation", *American Society of Heating, Refrigeration, and Air-Conditioning Engineers (ASHRAE).*

Huang, P., Huang, G. & Wang, Y. 2015, "HVAC system design under peak load prediction uncertainty using multiple-criterion decision making technique", *Energy and Buildings,* vol. 91, pp. 26-36.

Huang, S., Zuo, W. & Sohn, M.D. 2017, "Improved cooling tower control of legacy chiller plants by optimizing the condenser water set point", *Building and Environment,* vol. 111, pp. 33-46.

Huang, S., Zuo, W. & Sohn, M.D. 2016, "Amelioration of the cooling load based chiller sequencing control", *Applied Energy,* vol. 168, pp. 204-215.

Hussain, S., Gabbar, H.A., Bondarenko, D., Musharavati, F. & Pokharel, S. 2014, "Comfort-based fuzzy control optimization for energy conservation in HVAC systems", *Control Engineering Practice*, vol. 32, pp. 172-182.

Jia, Q. 2014, "A brief tutorial on event-based optimization with applications in smart buildings", *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pp. 1292.

Kusiak, A., Li, M. & Tang, F. 2010, "Modeling and optimization of HVAC energy consumption", *Applied Energy,* vol. 87, no. 10, pp. 3092-3102.

Lam, J.C. & Hui, S.C. 1995, "Outdoor design conditions for HVAC system design and energy estimation for buildings in Hong Kong", *Energy and Buildings,* vol. 22, no. 1, pp. 25-43.

Lam, J.C., Wan, K.K.W. & Cheung, K.L. 2009, "An analysis of climatic influences on chiller plant electricity consumption", *Applied Energy,* vol. 86, no. 6, pp. 933-940.

Lebrun, J., Silva, C.A., Trebilcock, F. & Winandy, E. 2004, "Simplified models for direct and indirect contact cooling towers and evaporative condensers", *Building Services Engineering Research and Technology,* vol. 25, no. 1, pp. 25-31.

Li, G., Hu, Y., Chen, H., Li, H., Hu, M., Guo, Y., Liu, J., Sun, S. & Sun, M. 2017, "Data partitioning and association mining for identifying VRF energy consumption patterns under various part loads and refrigerant charge conditions", *Applied Energy,* vol. 185, pp. 846-861.

Li, H. & Shi, Y. 2014, "Event-triggered robust model predictive control of continuous-time nonlinear systems", *Automatica,* vol. 50, no. 5, pp. 1507-1513.

Li, L., Ho, D.W., Zou, Y., Huang, C. & Lu, J. 2013, "Event-trigged control for discrete-time multi-agent networks", *Control Conference (ASCC), 2013 9th Asian*IEEE, , pp. 1.

Li, Q., Meng, Q., Cai, J., Yoshino, H. & Mochida, A. 2009, "Predicting hourly cooling load in the building: a comparison of support vector machine and different artificial neural networks", *Energy Conversion and Management,* vol. 50, no. 1, pp. 90-96.

Liao, Y., Huang, G., Sun, Y. & Zhang, L. 2014, *Uncertainty analysis for chiller sequencing control*.

Liaw, A. & Wiener, M. 2002, "Classification and regression by randomForest", *R news,* vol. 2, no. 3, pp. 18-22.

Liu, Q., Wang, Z., He, X. & Zhou, D. 2014, "A survey of event-based strategies on control and estimation", *Systems Science & Control Engineering: An Open Access Journal,* vol. 2, no. 1, pp. 90-97.

Lunze, J. & Lehmann, D. 2010, "A state-feedback approach to event-based control", *Automatica,* vol. 46, no. 1, pp. 211-215.

Ma, Z. 2008, *Online supervisory and optimal control of complex building central chilling systems*, The Hong Kong Polytechnic University.

Ma, Z. & Wang, S. 2009, "An optimal control strategy for complex building central chilled water systems for practical and real-time applications", *Building and Environment,* vol. 44, no. 6, pp. 1188-1198.

Ma, Z. & Wang, S. 2011, "Supervisory and optimal control of central chiller plants using simplified adaptive models and genetic algorithm", *Applied Energy,* vol. 88, no. 1, pp. 198-211.

Meyer, D., Leisch, F. & Hornik, K. 2003, "The support vector machine under test", *Neurocomputing,* vol. 55, no. 1, pp. 169-186.

Miller, C., Nagy, Z. & Schlueter, A. 2015, "Automated daily pattern filtering of measured building performance data", *Automation in Construction,* vol. 49, pp. 1-17.

Miskowicz, M. & Lunze, J. 2015, "Event-Based Control: Introduction and Survey" in *Event-Based Control and Signal Processing* CRC Press, , pp. 3-20.

Mossolly, M., Ghali, K. & Ghaddar, N. 2009, "Optimal control strategy for a multi-zone air conditioning system using a genetic algorithm", *Energy*, vol. 34, no. 1, pp. 58-66.

Nassif, N., Kajl, S. & Sabourin, R. 2005, "Optimization of HVAC control system strategy using two-objective genetic algorithm", *HVAC&R Research*, vol. 11, no. 3, pp. 459-486.

Okochi, G.S. & Yao, Y. 2016, "A review of recent developments and technological advancements of variable-air-volume (VAV) air-conditioning systems", *Renewable and Sustainable Energy Reviews,* vol. 59, pp. 784-817.

Pérez-Lombard, L., Ortiz, J. & Pout, C. 2008, "A review on buildings energy consumption information", *Energy and Buildings*, vol. 40, no. 3, pp. 394-398.

Sandee, J., Heemels, W. & Van Den Bosch, P. 2007, "Case studies in event-driven control" in *Hybrid Systems: computation and control* Springer, , pp. 762-765.

Sandee, J.H. 2006, *Event-driven control in theory and practice: Trade-offs in software and control performance*, Technische Universiteit Eindhoven (The Netherlands).

Shaikh, P.H., Nor, N.B.M., Nallagownden, P., Elamvazuthi, I. & Ibrahim, T. 2014, "A review on optimized control systems for building energy and comfort management of smart sustainable buildings", *Renewable and Sustainable Energy Reviews*, vol. 34, pp. 409-429.

Stark, W., Wang, H., Worthen, A., Lafortune, S. & Teneketzis, D. 2002, "Low-energy wireless communication network design", *IEEE Wireless Communications*, vol. 9, no. 4, pp. 60-72.

Suganthi, L., Iniyan, S. & Samuel, A.A. 2015, "Applications of fuzzy logic in renewable energy systems–a review", *Renewable and Sustainable Energy Reviews,* vol. 48, pp. 585-607.

Sun, B., Luh, P.B., Jia, Q. & Yan, B. 2015, "Event-Based Optimization Within the Lagrangian Relaxation Framework for Energy Savings in HVAC Systems", *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1396-1406.

Sun, B., Luh, P.B., Jia, Q. & Yan, B. 2013a, "Event-based optimization with non-stationary uncertainties to save energy costs of HVAC systems in buildings", *Automation Science and Engineering (CASE), 2013 IEEE International Conference on*IEEE, , pp. 436.

Sun, J. & Reddy, A. 2005, "Optimal control of building HVAC&R systems using complete simulation-based sequential quadratic programming (CSB-SQP)", *Building and Environment,* vol. 40, no. 5, pp. 657-669.

Sun, Y., Huang, G., Li, Z. & Wang, S. 2013b, "Multiplexed optimization for complex air conditioning systems", *Building and Environment,* vol. 65, pp. 99-108.

Sun, Y., Wang, S. & Xiao, F. 2013, "In situ performance comparison and evaluation of three chiller sequencing control strategies in a super high-rise building", *Energy and Buildings,* vol. 61, pp. 333-343.

Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P. & Feuston, B.P. 2003, "Random forest: a classification and regression tool for compound classification and QSAR modeling", *Journal of chemical information and computer sciences,* vol. 43, no. 6, pp. 1947-1958.

Tang, F., Kusiak, A. & Wei, X. 2014, "Modeling and short-term prediction of HVAC system with a clustering algorithm", *Energy and Buildings,* vol. 82, pp. 310-321.

The R Foundation *R: A language and environment for statistical computing.*

U.S. Department of Energy 2016, *International Energy Outlook 2016,* U.S. Energy Information Administration, Washington, DC.

van Moeseke, G., Bruyère, I. & De Herde, A. 2007, "Impact of control rules on the efficiency of shading devices and free cooling for office buildings", *Building and Environment,* vol. 42, no. 2, pp. 784-793.

Wang, S. 1998, "Dynamic simulation of a building central chilling system and evaluation of EMCS on-line control strategies", *Building and Environment,* vol. 33, no. 1, pp. 1-20.

Wang, S. & Burnett, J. 1998, "Variable-air-volume air-conditioning systems: Optimal reset of static pressure setpoint", *Building Services Engineering Research and Technology,* vol. 19, no. 4, pp. 219-231.

Wang, S. & Ma, Z. 2008, "Supervisory and optimal control of building HVAC systems: a review", *HVAC&R Research,* vol. 14, no. 1, pp. 3-32.

Wang, J., Huang, G., Sun, Y. & Liu, X. 2016, "Event-driven optimization of complex HVAC systems", *Energy and Buildings,* vol. 133, pp. 79-87.

Windham, A. & Treado, S. 2016, "A review of multi-agent systems concepts and research related to building HVAC control", *Science and Technology for the Built Environment,* vol. 22, no. 1, pp. 50-66.

Wu, Z., Jia, Q. & Guan, X. 2016, "Optimal control of multiroom HVAC system: An event-based approach", *IEEE Transactions on Control Systems Technology,* vol. 24, no. 2, pp. 662-669.

Wu, Z., Jia, Q. & Guan, X. 2015, "Local and global event-based optimization: Performace and complexity", *Automation Science and Engineering (CASE), 2015 IEEE International Conference on*IEEE, , pp. 1375.

Wu, Z., Jia, Q. & Guan, X. 2014, "Event-based optimization for multi-room HVAC system", *Control Conference (CCC), 2014 33rd Chinese*IEEE, , pp. 3927.

Xia, L., Jia, Q. & Cao, X. 2014, "A tutorial on event-based optimization—a new optimization framework", *Discrete Event Dynamic Systems,* vol. 24, no. 2, pp. 103-132.

Xiao, F. & Fan, C. 2014, "Data mining in building automation system for improving building operational performance", *Energy and Buildings,* vol. 75, pp. 109-118.

Xu, W., Ho, D.W., Li, L. & Cao, J. 2017, "Event-triggered schemes on leader-following consensus of general linear multiagent systems under different topologies", *IEEE transactions on cybernetics,* vol. 47, no. 1, pp. 212-223.

Yao, Y., Lian, Z., Hou, Z. & Zhou, X. 2004, "Optimal operation of a large cooling system based on an empirical model", *Applied Thermal Engineering,* vol. 24, no. 16, pp. 2303-2321.

You, K. & Xie, L. 2013, "Survey of Recent Progress in Networked Control Systems", *Acta Automatica Sinica,* vol. 39, no. 2, pp. 101-108.

Yu, F., Ho, W., Chan, K. & Sit, R. 2017, "Critique of operating variables importance on chiller energy performance using random forest", *Energy and Buildings,* vol. 139, pp. 653-664.

Yu, Z., Haghighat, F., Fung, B.C. & Yoshino, H. 2010, "A decision tree method for building energy demand modeling", *Energy and Buildings,* vol. 42, no. 10, pp. 1637-1646.

Yu, F.W. & Chan, K.T. 2010, "Economic benefits of optimal control for water-cooled chiller systems serving hotels in a subtropical climate", *Energy and Buildings,* vol. 42, no. 2, pp. 203-209.

Yu, F.W. & Chan, K.T. 2008, "Optimization of water-cooled chiller system with load-based speed control", *Applied Energy,* vol. 85, no. 10, pp. 931-950.

Zaheer-uddin, M. & Zheng, G.R. 2000, "Optimal control of time-scheduled heating, ventilating and air conditioning processes in buildings", *Energy Conversion and Management,* vol. 41, no. 1, pp. 49-60.

Zhang, J. & Feng, G. 2014, "Event-driven observer-based output feedback control for linear systems", *Automatica,* vol. 50, no. 7, pp. 1852-1859.

Zhong, M. & Cassandras, C.G. 2010, "Asynchronous distributed optimization with event-driven communication", *IEEE Transactions on Automatic Control,* vol. 55, no. 12, pp. 2735-2750.

APPENDIX

Appendix A – Component Models

A.1 Chiller performance model

A simplified physical chiller model was used to predict the power consumption (P_{ch}) after the variables of condenser inlet water temperature, evaporator cooling energy and its outlet water temperature set-point were given. The overall heat transfer coefficients of the evaporator and condenser (UA_{ev} , UA_{cd}) were needed for calculation of the refrigerant vapor compression power (P_{com}) and they were calculated in Eq. (A.1) and (A.2) respectively (Ma, Wang 2009). The chiller power consumption (P_{ch}) was estimated based on P_{com}, as in Eq. (A.3). The nine parameters ($c_{1\text{-}9}$) used in the model can be identified using a regression approach (Huang, Wang & Sun 2008).

$$c_1 M_{w,ev}^{-0.8} + c_2 Q_{ev}^{-0.745} + c_3 = 1/UA_{ev} \tag{A.1}$$

$$c_4 M_{w,ev}^{-0.8} + c_5 \left(Q_{ev} + P_{ch}\right)^{1/3} + c_6 = 1/UA_{cd} \tag{A.2}$$

$$P_{ch} = c_7 + c_8 P_{com} + c_9 P_{com}^2 \tag{A.3}$$

where $M_{w,ev}$ and Q_{ev} are chilled water flow rate and chiller supplied cooling respectively.

A.2 Cooling tower performance model

In this study, a simplified cooling tower model developed by (Lebrun et al. 2004) was used to estimate the required air flow rate (M_a) for maintaining the cooling water supply temperature at its set-point. Eq. (A.4) was used to predict the needed air flow rate (M_a). $C_{p,af}$ was the fictitious air specific heat which was computed by Eq. (A.5). Model parameters D_0, m and n can be identified using a regression approach.

$$UA = D_0\left(M_w / M_{w,des}\right)^m \left(M_a / M_{a,des}\right)^n \left(C_{p,af} / C_{p,a}\right) \tag{A.4}$$

$$C_{p,af} = h_{a,out} - h_{a,in} / T_{wb,out} - T_{wb,in} \tag{A.5}$$

where M is mass flow rate, T is temperature and h represents enthalpy; subscripts w, a, des and wb are water, air, designed value and wet bulb temperature respectively.

A.3 AHU coil performance model

The performance model of AHU coil was shown in Eq. (A.6) and (A.7). The relationship between the water temperatures and the air temperatures was approximated as Eq. (A.6) where coefficient α can be identified using the data from the constructed platform.

$$T_{w,out,AHU} = \alpha T_{air,out,AHU} - T_{w,in,AHU} \tag{A.6}$$

$$M_{w,AHU} = Q / C_p \left(T_{w,out,AHU} - T_{w,in,AHU} \right) \tag{A.7}$$

where Q is the building cooling load; C_p is the water specific heat; M is mass flow rate and T is temperature; subscripts *w, air, in* and *out* are water, air, inlet and outlet respectively.

A.4 Heat exchanger performance model

The heat exchanger performance model was used to predict the required chilled water flow rate ($M_{w,prm,HX}$) and outlet water temperature ($T_{out,prm,HX}$) at the primary side as shown in Eq. (A.8) and (A.9).

$$M_{w,prm,HX} = Q / C_{p,w} \left(T_{in,\sec,HX} - T_{in,prm,HX} \right) \tag{A.8}$$

$$T_{out,prm,HX} = T_{in,prm,HX} + Q / C_{p,w} M_{w,prm,HX} \tag{A.9}$$

where $C_{p,w}$ is the water specific heat; subscripts *HX, prm* and *sec* are heat exchanger, primary side and secondary side; others are same with above.

A.5 Pump and fan performance models

The power consumptions (P) of the cooling tower fans and the variable speed water pumps were estimated simply using the affinity law, as shown in Eq. (A.10). For simplicity, the coefficient β took the value 0.6 for fans and 3 for pumps in this study.

$$P = \beta M^3 \tag{A.10}$$

Appendix B – TRNSYS Settings

Table A.1 Major TRNSYS settings in “control cards”

Item	**Value**	**Unit**
Simulation start time	0	min
Simulation stop time	1440	min
Simulation time step	30	s
Solution method	Successive	/
The minimum relaxation factor	1	/
The maximum relaxation factor	1	/
Equation solver	0	/
Equation trace	False	/
Debug mode	False	/
Tolerance integration	0.001	dimensionless
Tolerance convergence	0.001	dimensionless
Tolerance value	Absolute	/

Appendix C – Calculation of "Δh"

"Δh" is the average difference between the specific enthalpies of saturated (at inlet and outlet cooling water temp.) and bulk air. Since the case study of this thesis uses the spring, summer and autumn operation scenarios in Hong Kong, the corresponding outdoor air temperatures are listed in (Lam, Hui 1995) Table A.2. The averages dry-bulb and wet-bulb temperatures from April to November were used as the outdoor air condition. In Table A.3, the enthalpy difference was calculated in a reverse way to find the lower bound, where the bulk air condition was assumed to be constant in the cooling tower for the sake of calculation convenience; the outlet cooling water temperature is selected by adding a 2.8 ºC approach to the average wet-bulb temperature (from Table A.2); the inlet cooling water temperature is selected by applying a 5 ºC range to the outlet cooling water temperature.

Table A.2 Monthly average dry-bulb and wet-bulb temperatures of Hong Kong (45-year data)

	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	**Ave**
T_{db}(ºC)	22	25.8	27.6	28.6	28.3	27.5	25	21.2	**25.8**
T_{wb}(ºC)	20.1	23.7	25.4	26	25.8	24.7	21.1	17.7	**23.1**

Table A.3 Enthalpy difference calculation

Parameter Name	**T_{db}(ºC)**	**RH (%)**	**Enthalpy (kJ/kg)**	**Enthalpy difference from bulk air (kJ/kg)**	**Remark**
Bulk air	25.8	80 (T_{wb}=23.1ºC)	69	/	Data from Table A.2
Inlet cooling water temp.	30.9	100	104	35	$30.9 = 25.9 + 5$ (ºC)
Outlet cooling water temp.	25.9	100	80	11	$25.9 = 23.1 + 2.8$ (ºC)
Average enthalpy difference			$\frac{\mathbf{35+11}}{\mathbf{2}} = \mathbf{23}$ (kJ/kg)		

(Note: “temp.” = “temperature”)

Appendix D – Energy and Computational Performances of Different Events

Table A.4 Energy and computational performances of different optimization methods (autumn case)

Op. methods	**EC (Kwh)**	**ES**	**Triggering times**	**CT(s)**	**CS**	**Event threshold**
No Op.	132416	N/A	0	N/A	N/A	
15 mins	120429	9.05%	96	109.8	0.00%	
Ch. On/Off	125437	5.27%[#]	4	6.69	93.91%[^]	
PLR Change	120169	9.85%	36	36.44	66.81%	7%
PLR Change	120455	9.00%	30	33.37	69.61%	8%
PLR Change	120646	8.80%	29	28.85	73.72%	9%
PLR Change	120724	8.80%	25	24.17	77.99%	10%
PLR Change	120891	8.72%	22	22.28	79.71%	11%
PLR Change	120891	8.67%	20	20.43	81.39%	12%
PLR Change	120891	8.61%	20	20.43	81.39%	13%
Average		**8.92%**			**75.80%**	
STD		**0.08%**			**3.19%**	
T_{wb} Change	121386	8.33%	27	25.57	76.71%	0.1 ºC
T_{wb} Change	121521	8.23%	13	14.6	86.70%	0.2 ºC
T_{wb} Change	122087	7.80%	4	5.67	94.84%	0.3 ºC
T_{wb} Change	123727	6.56%	4	5.67	94.84%	0.4 ºC
T_{wb} Change	125724	5.05%	3	3.91	96.44%	0.5 ºC
T_{wb} Change	126672	4.34%	1	2.78	97.47%	0.6 ºC
T_{wb} Change	126672	4.34%	1	2.78	97.47%	0.7 ºC
Average		**6.38%**			**92.07%**	
STD		**1.80%**			**7.72%**	
T_{apr}	123126	7.02%	1	1.31	98.81%	2.8 ºC
T_{apr}	121457	8.28%	2	3.23	97.06%	3.2 ºC
T_{apr}	121546	8.21%	3	4.76	95.66%	3.6 ºC
T_{apr}	121766	8.04%	5	6.88	93.73%	4.0 ºC
T_{apr}	121493	8.25%	11	15.70	85.70%	4.4 ºC
T_{apr}	121387	8.33%	13	18.60	83.06%	4.8 ºC

T_{apr}	121321	8.38%	30	32.50	70.40%	5.2 ºC
Average		**8.07%**			**89.20%**	
STD		**0.48%**			**10.16%**	
Δh	121362	8.35%	13	13.12	88.05%	24 kJ/kg
Δh	121362	8.35%	13	13.12	88.05%	27 kJ/kg
Δh	120708	8.84%	14	18.50	83.15%	30 kJ/kg
Δh	120793	8.78%	15	18.93	82.76%	33 kJ/kg
Δh	121538	8.22%	25	28.40	74.13%	36 kJ/kg
Δh	120449	9.04%	63	68.37	37.73%	39 kJ/kg
Δh	120174	9.25%	155	140.30	*-27.78%*	42 kJ/kg
Average		**8.69%**			**60.87%**	
STD		**0.39%**			**42.86%**	

("Op." = "Optimization"; "EC" = "Energy Consumption"; "ES" = "Energy Saving"; "CT" = "Computation Time"; "CS" = "Computation Saving"; "STD" = "Sample Standard Deviation")

$\text{Energy saving} = \frac{(132416-125437)}{132416} \times 100\% = 5.27\%$

^ $\text{Computation saving} = \frac{(109.8-6.69)}{109.8} \times 100\% = 93.91\%$

Table A.5 Energy performances, computational performances and performance scores of different optimization methods (summer case)

Op. methods	**EC (Kwh)**	**ES**	**Triggering times**	**CT(s)**	**CS**	**Event threshold**
No Op.	197663	N/A	0	N/A	N/A	
15 mins	177456	10.22%	96	138.9	0.00%	
Ch. On/Off	182030	7.91%	3	1.83	98.68%	
PLR Change	177171	10.37%	45	70.28	49.40%	7%
PLR Change	177707	10.10%	39	63.59	54.22%	8%
PLR Change	175522	11.20%	37	60.27	56.61%	9%
PLR Change	177095	10.41%	32	52.94	61.89%	10%
PLR Change	177855	10.02%	29	48.04	65.41%	11%
PLR Change	179201	9.34%	25	46.07	66.83%	12%

PLR Change	178668	9.61%	24	44.07	68.27%	13%
Average		**10.15%**			**60.38%**	
STD		**0.60%**			**7.12%**	
T_{wb} Change	179232	9.32%	28	42.29	69.55%	0.1 ºC
T_{wb} Change	180479	8.69%	10	15.59	88.78%	0.2 ºC
T_{wb} Change	182088	7.88%	6	8.03	94.22%	0.3 ºC
T_{wb} Change	183094	7.37%	3	5.02	96.39%	0.4 ºC
T_{wb} Change	183392	7.22%	2	3.53	97.46%	0.5 ºC
T_{wb} Change	183392	7.22%	2	3.53	97.46%	0.6 ºC
T_{wb} Change	183392	7.22%	2	3.53	97.46%	0.7 ºC
Average		**7.85%**			**91.62%**	
STD		**0.85%**			**10.22%**	
T_{apr}	178689	9.60%	9	9.52	93.15%	2.8 ºC
T_{apr}	178534	9.68%	10	10.68	92.31%	3.2 ºC
T_{apr}	176916	10.50%	11	13.50	90.28%	3.6 ºC
T_{apr}	180822	8.52%	35	49.52	64.35%	4.0 ºC
T_{apr}	181421	8.22%	51	75.19	45.87%	4.4 ºC
T_{apr}	181791	8.03%	83	116.90	15.84%	4.8 ºC
T_{apr}	182045	7.90%	153	137.10	1.30%	5.2 ºC
Average		**8.92%**			**57.58%**	
STD		**1.00%**			**37.93%**	
Δh	180137	8.87%	12	10.81	92.22%	24 kJ/kg
Δh	180611	8.63%	17	17.00	87.76%	27 kJ/kg
Δh	178953	9.47%	27	34.15	75.41%	30 kJ/kg
Δh	174986	11.47%	60	83.59	39.82%	33 kJ/kg
Δh	178851	9.52%	71	105.00	24.41%	36 kJ/kg
Δh	176461	10.73%	74	109.20	21.38%	39 kJ/kg
Δh	177619	10.14%	91	120.20	13.46%	42 kJ/kg
Average		**9.83%**			**50.64%**	
STD		**1.02%**			**33.58%**	

Table A.6 Energy performances, computational performances and performance scores of different optimization methods (spring case)

Op. methods	**EC (Kwh)**	**ES**	**Triggering times**	**CT(s)**	**CS**	**Event threshold**
No Op.	77279	N/A	0	N/A	N/A	
15 mins	73670	4.67%	96	97.6	0.00%	
Ch. On/Off	75299	2.56%	2	3.00	96.93%	
PLR Change	74059	4.17%	23	31.31	66.64%	7%
PLR Change	73812	4.49%	21	26.93	71.31%	8%
PLR Change	74212	3.97%	17	14.73	84.30%	9%
PLR Change	73848	4.44%	16	14.12	84.95%	10%
PLR Change	73642	4.71%	15	13.68	85.42%	11%
PLR Change	74368	3.77%	13	12.06	87.15%	12%
PLR Change	74252	3.92%	13	12.06	87.15%	13%
Average		**4.21%**			**79.96%**	
STD		**0.35%**			**8.39%**	
T_{wb} Change	73763	4.55%	22	28.04	70.12%	0.1 ºC
T_{wb} Change	73981	4.27%	7	10.03	89.31%	0.2 ºC
T_{wb} Change	75109	2.81%	3	6.24	93.35%	0.3 ºC
T_{wb} Change	75267	2.60%	1	1.05	98.88%	0.4 ºC
T_{wb} Change	75267	2.60%	1	1.05	98.88%	0.5 ºC
T_{wb} Change	75267	2.60%	1	1.05	98.88%	0.6 ºC
T_{wb} Change	75267	2.60%	1	1.05	98.88%	0.7 ºC
Average		**3.15%**			**92.62%**	
STD		**0.87%**			**10.60%**	
T_{apr}	77279	0.00%	0	0.00	100.00%	2.8 ºC
T_{apr}	77279	0.00%	0	0.00	100.00%	3.2 ºC
T_{apr}	77279	0.00%	0	0.00	100.00%	3.6 ºC
T_{apr}	75524	2.27%	1	0.89	99.05%	4.0 ºC
T_{apr}	75052	2.88%	2	1.32	98.59%	4.4 ºC
T_{apr}	73827	4.47%	3	4.50	95.21%	4.8 ºC
T_{apr}	73916	4.35%	4	7.64	91.86%	5.2 ºC
Average		**2.00%**			**97.82%**	
STD		**2.02%**			**3.13%**	

Δh	73950	4.31%	4	2.88	96.93%	24 kJ/kg
Δh	72286	6.46%	5	3.50	96.27%	27 kJ/kg
Δh	72963	5.58%	18	11.61	87.63%	30 kJ/kg
Δh	72880	5.69%	27	16.78	82.12%	33 kJ/kg
Δh	73744	4.57%	76	42.78	54.42%	36 kJ/kg
Δh	72981	5.56%	102	98.5	*-4.95%*	39 kJ/kg
Δh	73567	4.80%	255	227.1	*-141.98%*	42 kJ/kg
Average		**5.28%**			**38.63%**	
STD		**0.75%**			**87.35%**	

Appendix E – Code of Neural Network Simulation Function in MATLAB

E.1- Code for Spring Case

```
function [y1] = myNeuralNetworkFunction(x1)

%MYNEURALNETWORKFUNCTION neural network simulation function.

%

% Generated by Neural Network Toolbox function genFunction, 03-Aug-2017 15:54:27.

%

% [y1] = myNeuralNetworkFunction(x1) takes these arguments:

%     x = 9xQ matrix, input #1

% and returns:

%     y = 1xQ matrix, output #1

% where Q is the number of samples.


%#ok<*RPMT0>


% ===== NEURAL NETWORK CONSTANTS =====


% Input 1

x1_step1.xoffset =

[0.460139984439836;2.9362168204308;14.1539656461813;20;18.8;1;0;21.7362168204308;5.008615
48815778];

x1_step1.gain =

[3.23019393173495;0.378877066324021;0.0367910153978271;0.000688494228701654;0.740740740
739662;2;0.0491265987231754;0.296630191288634;0.663530013671568];

x1_step1.ymin = -1;
```

```
% Layer 1

b1 = [3.6250332062058996;-3.9521385655819925;-6.7165056873757836];

IW1_1 = [2.3733484841448269 1.8819935850095633 -5.1765568714416235 4.4856560325512742

-1.5538174461666483 -0.90683327727021379 0.0029723761220809414 0.51692657961245048

-0.17880690187438697;-0.38134973328905059 -0.39173715699228873 -0.17213764917095242

-2.4629290303259239 0.97084796405211471 -3.0553140793474896 2.0220836492217078

0.19424982219241702 -1.3566325500029237;0.89632286323076715 7.3500489699524785

-14.728861145392237 5.8092446046493622 -0.21942430646773642 -0.60049088118299654

6.8047938016715879 4.2747919967918167 0.021792958319730667];


% Layer 2

b2 = -0.42897355504984491;

LW2_1 = [0.56836497335322811 -0.46818554712929816 0.54454377695323763];


% Output 1

y1_step1.ymin = -1;

y1_step1.gain = 4.45162316371251;

y1_step1.xoffset = 1.92545393854397;


% ===== SIMULATION ========


% Dimensions

Q = size(x1,2); % samples


% Input 1
```

```
xp1 = mapminmax_apply(x1,x1_step1);

% Layer 1
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);

% Layer 2
a2 = repmat(b2,1,Q) + LW2_1*a1;

% Output 1
y1 = mapminmax_reverse(a2,y1_step1);
end

% ===== MODULE FUNCTIONS ========

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end
```

```
% Map Minimum and Maximum Output Reverse-Processing Function

function x = mapminmax_reverse(y,settings)

x = bsxfun(@minus,y,settings.ymin);

x = bsxfun(@rdivide,x,settings.gain);

x = bsxfun(@plus,x,settings.xoffset);

end
```

E.2- Code for Summer Case

```
function [y1] = myNeuralNetworkFunction(x1)

%MYNEURALNETWORKFUNCTION neural network simulation function.

%

% Generated by Neural Network Toolbox function genFunction, 03-Aug-2017 15:32:48.

%

% [y1] = myNeuralNetworkFunction(x1) takes these arguments:

%     x = 9xQ matrix, input #1

% and returns:

%     y = 1xQ matrix, output #1

% where Q is the number of samples.


%#ok<*RPMT0>


% ===== NEURAL NETWORK CONSTANTS =====


% Input 1

x1_step1.xoffset = [0.477;0;0;27;25.4;1;0;24.6;5.01];
```

```
x1_step1.gain =
[3.3167495854063;0.284900284900285;0.0241545893719807;0.000354421407052986;0.689655172413793;1;0.0421052631578947;0.238095238095238;0.56980056980057];
x1_step1.ymin = -1;

% Layer 1
b1 = [-3.2480340257091358;-0.39706110132746958;-2.7047685650128046];
IW1_1 = [-1.3972899242192005 -0.49478163838702993 0.87574765043826042
2.3537803012377636 1.0997802122248643 -2.50241028226651 4.2632569903417918
-1.9007762531993735 3.389380240409801;2.8474144197495126 1.8704569620886109
-0.9803878395665887 -0.11877528973702328 -0.48625386903836498 -1.3571961375102204
2.9877642331337606 -2.4898881509520052 2.3119454799894812;-1.4168147158870206
-0.91020353982264735 0.20176568930739958 -0.68208759915879202 0.1234549942898804
-0.13407388991308655 2.4050539285695107 -0.3497274330110488 1.1223750820299703];

% Layer 2
b2 = -0.056027787891006001;
LW2_1 = [0.25781689498681559 0.15927246701372047 -0.35013740072612409];

% Output 1
y1_step1.ymin = -1;
y1_step1.gain = 1.58730158730159;
y1_step1.xoffset = 1.38;

% ===== SIMULATION ========
```

```
% Dimensions

Q = size(x1,2); % samples


% Input 1

xp1 = mapminmax_apply(x1,x1_step1);


% Layer 1

a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);


% Layer 2

a2 = repmat(b2,1,Q) + LW2_1*a1;


% Output 1

y1 = mapminmax_reverse(a2,y1_step1);

end


% ===== MODULE FUNCTIONS ========


% Map Minimum and Maximum Input Processing Function

function y = mapminmax_apply(x,settings)

y = bsxfun(@minus,x,settings.xoffset);

y = bsxfun(@times,y,settings.gain);

y = bsxfun(@plus,y,settings.ymin);

end


% Sigmoid Symmetric Transfer Function
```

```
function a = tansig_apply(n,~)

a = 2 ./ (1 + exp(-2*n)) - 1;

end


% Map Minimum and Maximum Output Reverse-Processing Function

function x = mapminmax_reverse(y,settings)

x = bsxfun(@minus,y,settings.ymin);

x = bsxfun(@rdivide,x,settings.gain);

x = bsxfun(@plus,x,settings.xoffset);

end
```

Appendix F – EDO Code in TRNSYS

```
% Event-driven optimization for HVAC optimal control
% Author: WANG Junqi ; Sep-2016
% Department of Architecture and Civil Engineering, City University of Hong Kong
% Solution algorithm: use exhaustive search to update four decision variables;
% Problem type: model-based optimal control & static optimization
% ******************************************************************
% Inputs:
%     1.  Twevin   Evaporator inlet water temperature            (C)
%     2.  Twcdin   Condenser inlet water temperature             (C)
%     3.  Qev      Single chiller cooling load                   (kW)
%     4.  C1       Coefficient for evaporator UA calculation     (-)
%     5.  C2       Coefficient for evaporator UA calculation     (-)
%     6.  C3       Coefficient for evaporator UA calculation     (-)
%     7.  C4       Coefficient for condenser UA calculation      (-)
%     8.  C5       Coefficient for condenser UA calculation      (-)
%     9.  C6       Coefficient for condenser UA calculation      (-)
%     10.  a0      Coefficient for the chiller power calculation (-)
%     11. a1       Coefficient for the chiller power calculation (-)
%     12. a2       Coefficient for the chiller power calculation (-)
% ******************************************************************
% Outputs:
%     1. Twcdout   Condeser outlet water temperature             (C)
%     2. Pcom      Chiller power consumption                     (kW)
```

% **%

% Example M-file called by TRNSYS Type 155

%

% Data passed from / to TRNSYS

% trnTime (1x1): simulation time

% trnInfo (15x1): TRNSYS info array

% trnInputs (nIx1): TRNSYS inputs

% trnStartTime (1x1): TRNSYS Simulation Start time

% trnStopTime (1x1): TRNSYS Simulation Stop time

% trnTimeStep (1x1): TRNSYS Simulation time step

% mFileErrorCode (1x1): Error code for this m-file. It is set to 1 by TRNSYS and the m-file should set it to 0 at the end to indicate that the call was successful. Any non-zero value will stop the simulation

% trnOutputs (nOx1): TRNSYS outputs

% Notes:

% ------

% You can use the values of trnInfo(7), trnInfo(8) and trnInfo(13) to identify the call (e.g. first iteration, etc.)

% Real-time controllers (callingMode = 10) will only be called once per time step with trnInfo(13) = 1 (after convergence)

% The number of inputs is given by trnInfo(3)

% The number of expected outputs is given by trnInfo(6)

% WARNING: if multiple units of Type 155 are used, the variables passed from/to TRNSYS will be sized according to

% the maximum required by all units. You should cope with that by only using the part of the arrays that is

% really used by the current m-File. Example: use "nI = trnInfo(3); myInputs = trnInputs(1:nI);"

```
% rather than "MyInputs = trnInputs;"
% Please also note that all m-files share the same workspace in Matlab (they are "scripts", not
"functions") so
% variables like trnInfo, trnTime, etc. will be overwritten at each call.
% "Local" variables like iCall, iStep in this example will also be
% shared by all units
% (i.e. they should be given a different name in each m-File if required)
-----------------------------------------------------------------------------------------------------
% This example implements a very simple component. The component is iterative (should be called at
each TRNSYS call)
% MKu, October 2004
-----------------------------------------------------------------------------------------------------


% TRNSYS sets mFileErrorCode = 1 at the beginning of the M-File for error detection
% This file increments mFileErrorCode at different places. If an error occurs in the m-file the last
successful step will be indicated by mFileErrorCode, which is displayed in the TRNSYS error message
% (Alan)-so the indicated value of mFileErrorCode can help you to find the error
% At the very end, the m-file sets mFileErrorCode to 0 to indicate that everything was OK


mFileErrorCode = 100;      %   Beginning of the m-file
% Part 0
% --- Process Inputs and global parameters---
nI = trnInfo(3);     % The number of inputs is given by trnInfo(3)
nO = trnInfo(6);    % The number of expected outputs is given by trnInfo(6)
Flow_Inlet = trnInputs(1);                    % Inlet water flow rate in primary loop
Temperature_Inlet = trnInputs(2);             % Chilled water (from CH) entering temp. to HX (cold side)
```

```
Temperature_Outlet = trnInputs(3);          % Chilled water leaving temp. from HX (cold side)

%PLR = trnInputs(4);                        % imported from txt file

T_HX_cold_out = trnInputs(5);               % Type 699

T_CT_out = trnInputs(6);                    % T253_CT

T_HX_hot_out = trnInputs(7);                % Type 661

Tw_CT_in = trnInputs(8);                    % T252_Div, added new on 8-Sep-2016

Tw_ev_in = trnInputs(9);                    % T251_Div1, added new on 12-Sep-2016

Ctrl_sig_HX = trnInputs(10);                % from 'Tchw_sup_HX', flow rate = [5,2075]

Ctrl_sig_Tair_sup = trnInputs(11);          % from 'Tair_sup', flow rate = [5,2100]

Q_load_act = trnInputs(39);

Tair_AHU_sup = trnInputs(40);

Wair_AHU_sup = trnInputs(41);

Pow_chiller = trnInputs(42);                % one chiller

Pow_CT = trnInputs(43);                     % one CT fan

Pow_pump_constant = trnInputs(44);          % from 'Pcom'

Mw_pump_HX_prm = trnInputs(45);             % overall primary side flow rate of HX

%Mw_pump_HX_sec = trnInputs(46);            % overall secondary side flow rate of HX, 'AHU_Num'

Mw_AHU = trnInputs(47);                     % water flow rate in one AHU, 'Tair_sup'

Mw_pump_HX_sec = Mw_AHU;

T_db_amb = trnInputs(48);

T_wb_amb = trnInputs(49); % wet bulb

Mwct = trnInputs(50);                       % water flow rate in single CT

Num_ct = trnInputs(51);

Thot_in_HX = trnInputs(55);

Tcold_in_HX = trnInputs(56);

Num_HX=round(max(1,Mw_pump_HX_sec/10));
```

```
mFileErrorCode = 110;                                   % After processing inputs


% Part 1

% --- First call of the simulation: initial time step (no iterations) ---

% (note that Matlab is initialized before this at the info(7) = -1 call, but the m-file is not called)

if ( (trnInfo(7) == 0) && (trnTime-trnStartTime < 1e-6) )

  % This is the first call (Counter will be incremented later for this very first call)

       iCall = 0; % "Local" variables like iCall, iStep in this example will also be shared by all units (i.e.

       they should be given a different name in each m-File if required).

       % This is the first time step

       iStep = 1;

       % Do some initialization stuff,

       % e.g. initialize variables and history of the variables for plotting at the end of the simulation

       % set the format of variables, i.e. value or vector/matrix.

       % (uncomment lines if you wish to store variables)

       nTimeSteps = (trnStopTime-trnStartTime)/trnTimeStep + 1;

       % Chillers' statement

       CH1.ONOFF = 1;

       CH1.OTime = trnTime;              % the latest operating (on or off) time

       CH2.ONOFF = 0;

       CH2.OTime = trnTime;

       CH3.ONOFF = 0;

       CH3.OTime = trnTime;

       CH4.ONOFF = 0;

       CH4.OTime = trnTime;
```

```
CH5.ONOFF = 0;
CH5.OTime = trnTime;
CH6.ONOFF = 0;
CH6.OTime = trnTime;
CH=[CH1,CH2,CH3,CH4,CH5,CH6]; % CH1's type is 'struct'
history.states = CH;
OTime = 0;                      % the latest time when a chiller is opened
history.OTime = OTime;
Num_previous=1;
history.Num_previous=Num_previous;
Pre_oper = 1;                   % assuming at the beginning the chiller switch one more
history.Pre_oper = Pre_oper;
op_times=0;
event_index=0;
t_compute = 0;
t_tot = 0;
Tcdw_set=26;                    % settings of spring case
Tchw_set=7.5;
Tchw_set_HX=9;
Tair_set=14;
T_threshold=0.5;
Tcdw_set_previous=26;
Tchw_set_previous=7.5;
Tchw_set_HX_previous=9;
Tair_set_previous=14;
```

```
    history.Tcdw_set=Tcdw_set;
    history.Tchw_set=Tchw_set;
    history.Tchw_set_HX=Tchw_set_HX;
    history.Tair_set=Tair_set;
    Q_load_act_vec=[];
    history.Q_load_act_vec=Q_load_act_vec;
    history.PLR=[];
    history.op_times=0;
    history.event_index=event_index;
    history.t_tot=t_tot;
    history.T_wb_amb=T_wb_amb;
    history.P_ch=0;
    history.P_ct=0;
    history.P_Tot_act=0;
    Num_AHU_prv=40;
    history.Num_AHU_prv=Num_AHU_prv;
    Num_HX_prv=40;
    history.Num_HX_prv=Num_HX_prv;
    % some varaibles for event-driven strategy
    dev_Twb = 0;
    T_approach = 0;
    delt_h = 0;
    % No return, normal calculations are also performed during this call
    mFileErrorCode = 120;     % After initialization call
end
```

```
% --- Very last call of the simulation (after the user clicks "OK") ---
% ----------------------------------------------------------------------------------------------------------------------
if ( trnInfo(8) == -1 )
    mFileErrorCode = 1000;
    % Do stuff at the end of the simulation,
    % e.g. calculate stats, draw plots, etc...
    mFileErrorCode = 0; % Tell TRNSYS that we reached the end of the m-file without errors
    return
end

% --- Post convergence calls: store values ---
% ----------------------------------------------------------------------------------------------------------------------
if (trnInfo(13) == 1)
    mFileErrorCode = 200;     % Beginning of a post-convergence call
    % This is the extra call that indicates that all Units have converged. You should do things like:
    % - calculate control signal that should be applied at "next time step"
    % - Store history of variables (for th euse of comparison at next step)
    history.Num_previous=Num_previous;
    history.Num_AHU_prv=Num_AHU_prv;
    history.Num_HX_prv=Num_HX_prv;
    history.states = CH;
    history.OTime = OTime;
    history.Pre_oper = Pre_oper;
    history.Tcdw_set=Tcdw_set;
    history.Tchw_set=Tchw_set;
    history.Tchw_set_HX=Tchw_set_HX;
```

```
        history.Tair_set=Tair_set;

        history.op_times=op_times;

        history.event_index=event_index;

        history.t_tot=t_tot;

        history.T_wb_amb=T_wb_amb;

        history.P_ch=P_ch;

        history.P_ct=P_ct;

        history.P_Tot_act=P_Tot_act;

        %history.Ctrl_sig_HX=Ctrl_sig_HX;

        % Note: If Calling Mode is set to 10, Matlab will not be called during iterative calls.

        % In that case only this loop will be executed and things like incrementing the "iStep" counter

        should be done here

        mFileErrorCode = 0; % Tell TRNSYS that we reached the end of the m-file without errors

        return    % Do not update outputs at this call

end


% Part 2

% --- All iterative calls ---------------------------------------------------------------------------------------------

% -------------------------------------------------------------------------------------------------------------------------

% --- If this is a first call in the time step, increment counter ---

if ( trnInfo(7) == 0 )

        iStep = iStep+1; % 'iStep' is the time step indicator, whose unit is the 'simulation time step' in

end


% --- Process Inputs ---

mFileErrorCode = 130;      % Beginning of iterative call
```

```
% --- Take value from history.(must have) ---

Num_previous=history.Num_previous;

Tcdw_set=history.Tcdw_set;

Tchw_set=history.Tchw_set;

Tchw_set_HX=history.Tchw_set_HX;

Tair_set=history.Tair_set;

op_times=history.op_times;

t_tot=history.t_tot;


%*********************************************************************

% "Traditional Chiller Sequence Control"

% chiller on/off event

%*********************************************************************

OPeriodThreshold = 10/60;       % the time period (hour) for two operations with same direction

CPeriodThreshold = 10/60;       % the time period (hour) for two operations with opposite directions

water_specific_heat = 4.19;

DeadBand = 8;

CAP=7230;

% calculate the number of chillers in operation

OTime = history.OTime;

Pre_oper = history.Pre_oper;

% the number of chillers in operation

% the period

Period = trnTime-OTime;

% the threshold for switching on/off a chiller

Q_state_threshold = Num_previous*CAP*(1+DeadBand/100);
```

```
Q_Destate_threshold = (Num_previous-1)*CAP*(1-DeadBand/100);


mFileErrorCode = 132;

% determine the number of chillers should be in operation

% open a chiller

Q_load=Q_load_act; % Q_load_act = trnInputs(39)

ChOnOffIndicator=0;

if Q_load > Q_state_threshold & Num_previous < 6

    if Period >= OPeriodThreshold & Pre_oper==1

        Num_previous = Num_previous+1;ChOnOffIndicator=1;

        Pre_oper=1;

        OTime = trnTime;

    elseif Period >= CPeriodThreshold & Pre_oper==-1

        Num_previous = Num_previous+1;ChOnOffIndicator=1;

        Pre_oper=1;

        OTime = trnTime;

    end

end


% close a chiller

if Q_load < Q_Destate_threshold & Num_previous > 1

    if Period >= OPeriodThreshold & Pre_oper==-1

        Num_previous = Num_previous-1;ChOnOffIndicator=1;

        Pre_oper=-1;

        OTime = trnTime;

    else if    Period >= CPeriodThreshold & Pre_oper==1
```

```
        Num_previous = Num_previous-1;ChOnOffIndicator=1;
        Pre_oper=-1;
        OTime = trnTime;
    end
end

if Num_previous<6
    for i=1:Num_previous
        CH(i).ONOFF = 1;
    end
    for i=Num_previous+1:6
        CH(i).ONOFF = 0;
    end
else
    for i=1:Num_previous
        CH(i).ONOFF = 1;
    end
end
%************************************************************************
% End "Traditional Chiller Sequence Control"
%************************************************************************

% --- Event-driven strategies are coded here ---
%************************************************************************
% “PLR change”
%************************************************************************
```

```
mFileErrorCode = 140;

CAP=7230;

Available_CAP = Num_previous*CAP;

PLR = Q_load_act / Available_CAP;

event_index=history.event_index;

history.PLR(iStep) = PLR;

PLRIndicator=0;

event_duration = iStep - event_index;

if event_duration >= 10      %    a minimal time interval

if event_index > 10           %    every step=30 s

% compare with the last contrl change, calculate 5mins-average 'PLR'

deviation = abs(PLR-mean(history.PLR((event_index-9):(event_index))));

if deviation >= 0.07

        PLRIndicator=1;

  end

end

end

%***********************************************************************

% End "PLR change"

%***********************************************************************


%***********************************************************************

% “Twb change”

%***********************************************************************

TwbIndicator=0;

dev_Twb = abs(T_wb_amb-history.T_wb_amb);
```

```
if dev_Twb >= 0.4    % threshold of "Twb change"= [0.2,0.3,0.4]
    TwbIndicator=1;
end
%***********************************************************************
% End "Twb"
%***********************************************************************

mFileErrorCode = 150;
%***********************************************************************
% Tapproach change
%***********************************************************************
T_approachIndicator=0;
T_approach = T_CT_out-T_wb_amb;
if event_duration >= 10
if T_approach < 4.0           % threshold of "T_approach"= [2.8, 5.2], BC=4.0 degree C
      T_approachIndicator=1;
end
end
%***********************************************************************
% delt_enthalpy_diff change
%***********************************************************************
% delt_h = h(Tw_CT_in,Tw_CT_in)- h(T_db_amb,T_wb_amb); fucntion, h(Tdb,Twb)
% [ref] C.C Chang, 2015, Applied Energy
% FHAIR(TDB,W)                    HAIR = ENTHALPY OF MOIST AIR (KJ/KG DRY AIR)
% FWTWB(TDB,TWB,PATM)    W = HUMIDITY RATIO(KG MOIST./KG DRY AIR
PATM = 101325;
```

```
HR_1 = FWTWB(T_db_amb,T_wb_amb,PATM);
h_1 =    FHAIR(T_db_amb,HR_1);
h_2 = FHSAT(Tw_CT_in,PATM);
delt_h = h_2 - h_1;      % unit=KJ/KG
delt_hIndicator=0;
if event_duration >= 10
if delt_h < 33                              % threshold of "delt_h"= [24, 42], BC=33 KJ/KG
    delt_hIndicator=1;
end
end
%**********************************************************************
% End "--- Event-driven strategies are coded here --- "
%**********************************************************************

%**********************************************************************
% --- Control Optimization ---
%**********************************************************************
mFileErrorCode=180;
Qev=Q_load_act/max(0.1,Num_previous);
C1=0;C2=0;C3=1.014712E-3;C4=0;C5=0;C6=1.247556E-3;a0=175;a1=0.8;a2=0;
b0=4.33096417708899;    b1=0.67687427213009; b2=-0.18025853127327;b3=-0.53797104466615;
b4=0.15777630465005;b5=-5.74761612912195;b6=68.2830117147551;

mFileErrorCode=141;
x_coeff = [C1,C2,C3,C4,C5,C6,a0,a1,a2,b0,b1,b2,b3,b4,b5,b6];
x1=[Tcdw_set,Tchw_set,Tchw_set_HX,Tair_set];
```

```
mFileErrorCode=160;

x2=[Q_load_act,Flow_Inlet,Qev,x_coeff,Mw_AHU,T_db_amb,T_wb_amb,Mwct,Num_ct,Num_previ
ous,Pow_pump_constant];


mFileErrorCode=182;

T_threshold=0.5;

Time_interval=120/60;            % frequency of time-driven optimization

Steplength=0.1;

t_compute=0;

% trnTime_duration=trnTime-trnStartTime; % the unit of 'trnTime' is 'hour'

% if mod(trnTime_duration,Time_interval)==0


if iStep == 11           % added on Mar-2-2016

    event_index=iStep; % initialization, first 'event_index' should > 10

end


if    PLRIndicator==1 | delt_hIndicator==1 | ChOnOffIndicator==1

  tic

  event_index=iStep;

  POWER_ARRARY=[];

  INDEX_ARRARY=[];

mFileErrorCode=183;


% set the ranges of 4 decision variables

% The dramatic change of the setting is not allowed in practice

if Tcdw_set+T_threshold>30    % set this bound based on outdoor Twb.
```

```
    Tcdw_up=30;
else Tcdw_up=Tcdw_set+T_threshold;
end
if Tcdw_set-T_threshold<24
    Tcdw_low=24;
else Tcdw_low=Tcdw_set-T_threshold;
end


if Tchw_set+T_threshold>8
    Tchw_up=8;
else Tchw_up=Tchw_set+T_threshold;
end
if Tchw_set-T_threshold<5
    Tchw_low=5;
else Tchw_low=Tchw_set-T_threshold;
end


if Tchw_set_HX+T_threshold>11.5
    Tchw_HX_up=11.5;
else Tchw_HX_up=Tchw_set_HX+T_threshold;
end
if Tchw_set_HX-T_threshold<Tchw_up+0.8
    Tchw_HX_low=Tchw_up+0.8;
else Tchw_HX_low=Tchw_set_HX-T_threshold
end
```

```
if Tchw_HX_low > Tchw_HX_up % added new to avoid error.
   Tchw_HX_up =   Tchw_HX_low+T_threshold;
end
if Tair_set+T_threshold>15
   Tair_up=15;
else Tair_up=Tair_set+T_threshold;
end
if Tair_set-T_threshold<12
   Tair_low=12;
else Tair_low=Tair_set-T_threshold;
end

% Exhaustive search method
for a1= Tcdw_low:Steplength:Tcdw_up
  for a2= Tchw_low:Steplength:Tchw_up
    for a3= Tchw_HX_low:Steplength:Tchw_HX_up
      for a4= Tair_low:Steplength:Tair_up
mFileErrorCode=144;
     x1=[a1,a2,a3,a4];
     POWER=GA_P_W([x1,x2]);
     POWER_ARRARY=[POWER_ARRARY;POWER];
     INDEX_ARRARY=[INDEX_ARRARY;x1];
      end
    end
  end
end
```

```
mFileErrorCode=185;
[C,I]=min(POWER_ARRARY);
X=INDEX_ARRARY(I,:);


mFileErrorCode=186;
    Tcdw_set=X(1);
    Tchw_set=X(2);
    Tchw_set_HX=X(3);
    Tair_set=X(4);
    toc
    t_compute=toc;
    op_times=op_times+1;
    t_tot=t_compute+history.t_tot;
end
%*************************************************************************
% End "--- Control Optimization ---"
%*************************************************************************
Flow_prm_makeup=0;


%*************************************************************************
% Calculate power consumption of components
%*************************************************************************
x1=[Tcdw_set,Tchw_set,Tchw_set_HX,Tair_set];
x1_act=[T_CT_out,Temperature_Inlet,T_HX_hot_out,Tair_AHU_sup];
P_tot_predict = GA_P_W([x1,x2]);
mFileErrorCode=190;
```

```
x2_act=[Q_load_act,Flow_Inlet,Qev,x_coeff,
Mw_pump_HX_prm,T_db_amb,T_wb_amb,Mwct,Num_ct,Num_previous,Pow_pump_constant,Ctrl_s
ig_HX,Ctrl_sig_Tair_sup];
[P_Tot_act,P_ch,P_ct,P_fan_AHU,P_pump_HX_prm,P_pump_HX_sec]=GA_Pow_act([x1_act,
x2_act]);
%******************************END**************************************
% End "Calculate power consumption of components"
%**********************************************************************
% --- Set outputs ---
trnOutputs(1) = Num_previous;
trnOutputs(2) = CH(1).ONOFF;
trnOutputs(3) = CH(2).ONOFF;
trnOutputs(4) = CH(3).ONOFF;
trnOutputs(5) = CH(4).ONOFF;
trnOutputs(6) = CH(5).ONOFF;
trnOutputs(7) = CH(6).ONOFF;
trnOutputs(8) = Q_load_act;
trnOutputs(9) = cop;
trnOutputs(11) = PLR;
trnOutputs(12) = T_approach;
trnOutputs(13) = delt_h;
mFileErrorCode = 300;
trnOutputs(16) = Tcdw_set;          % set-point of cooling water supply temperature
trnOutputs(17) = Tchw_set;          % set-point of chilled water supply temperature
%trnOutputs(18) = m_air_flow_tot;
trnOutputs(19) = Tchw_set_HX;       % set-point
```

```
trnOutputs(20) = P_tot_predict;            % PcomTot;
trnOutputs(21) = Tair_set;                 % set-point
trnOutputs(31) = Num_AHU;                  % not used
trnOutputs(32) = Num_HX;
mFileErrorCode = 301;
trnOutputs(33) = (P_ch+P_pump_HX_prm+P_pump_HX_sec+Pow_pump_constant); % chiller plant energy consumption
trnOutputs(34) = P_Tot_act;                % changed 12-Sep-2016
trnOutputs(35) = P_ch;
trnOutputs(36) = P_ct;
trnOutputs(37) = P_fan_AHU;                % changed 12-Sep-2016
trnOutputs(38) = P_pump_HX_prm;            % changed 12-Sep-2016
trnOutputs(39) = P_pump_HX_sec;            % changed 12-Sep-2016
mFileErrorCode = 302;
trnOutputs(40) = t_compute;
trnOutputs(41) = Ctrl_sig_HX;
trnOutputs(42) = Mw_HX_prm_each;
trnOutputs(43) = event_index;
trnOutputs(44) = Flow_prm_makeup;
trnOutputs(45) = op_times;
trnOutputs(46) = t_tot;
mFileErrorCode = 0;      % Tell TRNSYS that we reached the end of the m-file without errors
return
```

END “Appendix C – Control code”