Diagnostic Classifiers based on Fuzzy Bayesian Belief Networks and Deep Neural Networks for Demand-Controlled Ventilation and Heating Systems

DISSERTATION
zur Erlangung des Grades eines Doktors
der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegt von
M.Sc. Ali Behravan

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen
Siegen 2021

gedruckt auf alterungsbeständigem holz- und säurefreiem Papier

Betreuer und erster Gutachter
Prof. Dr.-Ing. habil. Roman Obermaisser
Universität Siegen

Zweiter Gutachter
Prof. Dr.-Ing. habil. Madjid Fathi
Universität Siegen

Mitglieder der Promotionskommission:
Prof. Dr. Roman Obermaisser
Prof. Dr. Madjid Fathi
Prof. Dr. Frank Gronwald
Prof. Dr. Kristof Van Laerhoven

Vorsitzender: Prof. Dr. Kristof Van Laerhoven

Tag der mündlichen Prüfung
08.12.2021

Diagnostic Classifiers based on Fuzzy Bayesian Belief Networks and Deep Neural Networks for Demand-Controlled Ventilation and Heating Systems

This dissertation is submitted for the degree of
Doctor of Engineering

By
M.Sc. Ali Behravan

Submitted to the Faculty of Science and Technology of
the University of Siegen
Siegen 2021

Dedicated to my mother, father, and sisters.

ACKNOWLEDGMENTS

First and foremost, I would like to express my special appreciation, regards, gratitude, and thanks to my advisor Prof. Dr. Roman Obermaisser for his professional and valuable assistance and his patience in guidance. He has been an excellent mentor and life coacher for me and I've learned a lot from him for my research and societal behavior. I would like to thank him for encouraging and advising my research and allowing me to pass the challenges and grow as a research scientist.

Special thanks to my second advisor Prof. Dr. Madjid Fathi for his remarkable advice to create a fruitful thesis. His guidance helped me a lot to provide mature content in this research work and to be on the right path.

Above all, I am profoundly and forever indebted to my parents, Mrs. Monireh Rasekhi and Prof. Dr. Hossein Behravan, for their unconditional love, support, and their encouragement throughout my entire life. Also, I received endless and uncountable love and support from my sisters, especially, in this part of my life.

Finally, it is my pleasure to express my gratitude to all the people who contributed, in whatever manner, to the success of my Ph.D. studies, especially, Dr. Daniel Müller, the head of the House of Young Talents at the University of Siegen by offering a scholarship in my scientific journey.

ABSTRACT

The building sector and its embedded control systems, especially the Heating, Ventilation, and Air-Conditioning (HVAC) systems, consume a considerable part of the global energy and produce gaseous emissions such as CO_2. On the other hand, the air exchange based on natural ventilation is a cost-efficient method to improve indoor air quality, dilute indoor CO_2concentration and odors, or remove pollutants or airborne virus particles (e.g., Covid-19) from the building zones. This air exchange during the cold seasons accounts for a heating load for the heating system that causes an increase in energy consumption. Therefore, optimization of HVAC systems to decrease harmful emissions considering potential energy saving is vital. Moreover, if the CO_2 generated by human metabolism is not correctly controlled to some limits, it can degrade indoor air quality, reduce the occupants' efficiency, lead to severe mental problems, or considerably impair the thinking ability. Thus, implementing a robust ventilation control system for the buildings particularly crowded office buildings is momentous. Demand-Controlled Ventilation (DCV) systems are promising solutions that control and optimize the ventilation rates based on thermal comfort and indoor air quality demands with a high potential in energy saving. Many researchers in the literature study DCV systems or adaptive thermal control separately while a comprehensive model containing both DCV and thermal control strategies is missing. Therefore, this thesis contains the combination of the DCV and heating systems with embedded sensors and actuators with the fault injection capabilities in a simulation framework to study such a complex system due to its numerous functions, inputs, and outputs for an in-depth assessment of the involved components' functionality and effective parameters, especially in case of component failures. Indoor air quality and comfort parameters in an office building can be monitored and controlled in real-time for various architectures based on a high-level specification of the building characteristics. The developed model is scalable based on the modular composability scheme. The user can generate different types of buildings with various architectures with many rooms and floors. The system model, fault injection capabilities, and diagnostic modules are automatically extended.

The high complexity of the DCV and heating systems with their many components makes them error-prone, more susceptible to faults, and more fragile. Faults in system components such as sensors and actuators can result in different types of failures and severe implications on efficiency with discomfort and performance degradation of occupants, energy waste, shortened component lifetime, and increased maintenance costs. Failure detection and fault diagnosis is the combination of system failure detection, which is the implication of the fault in a component of a system, with fault diagnosis that is finding the type, severity, time of occurrence, and locality of faults. The state-of-the-art of fault diagnosis methods for building energy systems, e.g., HVAC systems, contains data-driven and knowledge-driven diagnostic methods with corresponding strengths and shortcomings. The knowledge-driven methods are mainly based on expert knowledge and simulate the diagnostic thoughts of domain experts with the argumentation of uncertainties, diagnosis of different fault severities, and understandability. But they need a higher and more time-consuming effort to deeply understand the causal relationships among system inputs, faults, and symptoms. Moreover, the knowledge-based methods still lack automatic strategies to improve efficiency and they are less accurate than the data-driven methods. The data-driven methods, on the other hand, depend on similarities and patterns with high sensitivity to any change of patterns and more accuracy than the knowledge-driven methods. However, the data-driven methods require a huge amount of data for training the neural network for fault classification and they cannot provide the reason behind the results. In addition, the data-driven strategies indicate black boxes with low understandability.

The research gap filled by this thesis is therefore the combination of knowledge-driven and data-driven fault diagnosis in DCV and heating systems to gain advantages from both categories. The diagnostic method presented in this thesis involves an automatic strategy with low expert effort

without necessitation of in-depth understanding of the causal relationships compared to existing knowledge-driven methods with high understandability and high accuracy compared to the existing data-driven methods. The fault diagnosis method in this thesis combines a data-driven classifier with knowledge-driven inference, e.g., fuzzy logic and a Bayesian Belief Network (BBN) to provide an automatic diagnostic classifier that can diagnose any stuck-at or constant-valued faults in sensors and actuators. The combination of BBN and fuzzy logic itself analyzes the dependencies of the system signals based on the mutual information theory. In offline mode, a Relation-Direction Probability (RDP) table for each fault class is computed and stored in an offline fault library. The online mode determines the similarities between the real case RDP in the runtime and the offline library's RDPs. On the other hand, a data-driven strategy is specifically established using deep neural networks to compare and evaluate the performance of the presented composed diagnostic classifier. The data-driven classifier uses observed signals from faulty and healthy operations of the system to train and evaluate the performance of the designed neural network model. The diagnostic technique in this thesis is independent of the historical data, independent of expert knowledge, and computing-resource efficient. For the evaluation, four types of stuck-at faults at different components such as temperature sensor, CO_2 sensor, heater actuator, and damper actuator with various fault values at different instants of time were investigated. A fault injection framework artificially injects the faults to serve the diagnostic classifiers, e.g., training the models and evaluations. Results show the combined classifier introduced in this thesis has comparable performance to the data-driven method while advantaging the strengths of knowledge-driven methods.

KURZFASSUNG

Der Gebäudesektor und seine eingebetteten Kontrollsysteme, insbesondere die Heizungs-, Lüftungs- und Klimaanlagen (HLK), verbrauchen einen beträchtlichen Teil der weltweiten Energie und produzieren gasförmige Emissionen wie CO_2. Andererseits ist der Luftaustausch auf der Grundlage natürlicher Belüftung eine kosteneffiziente Methode zur Verbesserung der Luftqualität in Innenräumen, zur Verdünnung der CO_2-Konzentration und der Gerüche in Innenräumen oder zur Beseitigung von Schadstoffen oder luftgetragenen Viruspartikeln, (z.B. Covid-19) aus den Gebäudebereichen. Dieser Luftaustausch während der kalten Jahreszeit stellt eine große Heizlast für das Heizsystem dar, was einen Anstieg des Energieverbrauchs verursacht. Daher ist die Optimierung von HLK-Systemen zur Verringerung der Schadstoffemissionen unter Berücksichtigung möglicher Energieeinsparungen unerlässlich. Darüber hinaus kann das durch den menschlichen Stoffwechsel erzeugte CO_2, wenn es nicht bis zu einem gewissen Grad kontrolliert wird, die Luftqualität in Innenräumen verschlechtern, die Leistungsfähigkeit der Bewohner beeinträchtigen, schwere psychische Probleme hervorrufen oder zu einer erheblichen Herabsetzung des Denkvermögens führen. Daher ist die Implementierung eines robusten Lüftungssteuerungssystems für Gebäude, insbesondere für stark ausgelastete Bürogebäude, von großer Bedeutung. Systeme der bedarfsgesteuerten Lüftung (BGL) sind eine vielversprechende Lösung, die die Lüftungsraten auf der Grundlage des thermischen Komforts und der Anforderungen an die Raumluftqualität steuern und optimieren und ein hohes Potenzial für Energieeinsparungen bieten. Viele Forscher in der Literatur untersuchen BGL-Systeme oder adaptive thermische Steuerung separat, während ein umfassendes Modell, das sowohl BGL- als auch thermische Steuerungsstrategien enthält, fehlt. Daher beinhaltet diese Arbeit die Kombination von BGL- und Heizsystemen mit eingebetteten Sensoren und Aktoren mit der Fähigkeit zur Fehlerinjektion in einem Simulationsrahmen, um ein solch komplexes System aufgrund seiner zahlreichen Funktionen, Eingänge und Ausgänge für eine eingehende Bewertung der Funktionalität der beteiligten Komponenten und der effektiven Parameter zu untersuchen, insbesondere im Falle von Komponentenausfällen. Daher wird in dieser Arbeit ein Modell eines BGL- und Heizungssystems mit eingebetteten Sensoren und Aktoren erstellt und in MATLAB/Simulink simuliert. Innenraumluftqualität und Behaglichkeitsparameter in einem Bürogebäude können in Echtzeit für verschiedene Architekturen auf der Grundlage einer Spezifikation der Gebäudeeigenschaften überwacht und gesteuert werden. Das entwickelte Modell ist aufgrund des Designs der modularen Zusammensetzbarkeit skalierbar. Der Benutzer kann verschiedene Gebäudetypen mit unterschiedlichen Architekturen mit vielen Räumen und Stockwerken erzeugen. Das Systemmodell, die Fehlerinjektionsmöglichkeiten und die Diagnosemodule werden automatisch erweitert.

Die hohe Komplexität der BGL- und Heizungssysteme mit ihren Komponenten macht sie fehleranfällig und störanfälliger. Fehler in Systemkomponenten wie Sensoren und Aktoren können zu verschiedenen Arten von Ausfällen führen und schwerwiegende Auswirkungen auf die Effizienz haben, indem sie das Wohlbefinden und die Leistung der Bewohner beeinträchtigen, Energie verschwenden, die Lebensdauer der Komponenten verkürzen und die Wartungskosten erhöhen. Fehlererkennung und Fehlerdiagnose ist die Kombination von Fehlererkennung, d. h. die Erkennung von Fehlern in einem System, und Fehlerdiagnose, d. h. die Ermittlung der Art, des Schweregrads, des Zeitpunkts des Auftretens und des Ortes von Fehlern. Der Stand der Technik bei Fehlerdiagnosemethoden für Gebäudeenergiesysteme, z. B. HLK-Systeme, hat datengesteuerte und wissensgesteuerte Diagnosemethoden mit ihren Stärken und Schwächen untersucht. Die wissensbasierten Methoden beruhen hauptsächlich auf Expertenwissen und simulieren die diagnostischen Überlegungen von Fachleuten mit der Argumentation von Unsicherheiten, der Diagnose verschiedener Fehlerschweregrade und der Verständlichkeit. Sie erfordern jedoch einen höheren und zeitaufwändigen Aufwand, um die kausalen Beziehungen zwischen Systemeingaben, Fehlern und Symptomen zu verstehen. Außerdem fehlt es den wissensbasierten Methoden noch an

automatischen Strategien zur Verbesserung der Effizienz, und sie sind nicht weniger genau als die datengesteuerten Methoden. Die datengesteuerten Methoden hängen von Ähnlichkeiten und Mustern ab und reagieren sehr empfindlich auf Änderungen der Muster und sind genauer als die wissensgesteuerten Methoden. Die datengesteuerten Methoden benötigen jedoch große Datenmengen für das Training; sie können den Grund für die Ergebnisse nicht liefern, und ihre Strategien stellen Blackboxes mit geringer Verständlichkeit dar.

Die Forschungslücke, die in dieser Arbeit geschlossen wird, ist daher die Kombination von wissensbasierter und datenbasierter Diagnose in BGL- und Heizungssystemen, um die Vorteile beider Kategorien zu nutzen. Die in dieser Arbeit vorgestellte Diagnosemethode umfasst einen geringen Expertenaufwand, ohne dass ein vollständiges Verständnis der kausalen Zusammenhänge erforderlich ist, im Vergleich zu bestehenden wissensgetriebenen Methoden mit hoher Verständlichkeit und hoher Genauigkeit. Die Fehlerdiagnose kombiniert einen datengetriebenen Klassifikator mit wissensgetriebener Inferenz, z. B. Fuzzy-Logik und einem Bayesian Belief Network (BBN). Die Kombination aus BBN und Fuzzy-Logik analysiert selbst die Abhängigkeiten der Systemsignale auf der Grundlage der Theorie der Mutual Information . Im Offline-Modus wird eine Relation-Direction Probability (RDP)-Tabelle für jede Fehlerklasse berechnet und in einer Offline-Fehlerbibliothek gespeichert. Im Online-Modus werden die Ähnlichkeiten zwischen der RDP des realen Falls zur Laufzeit und der RDP der Offline-Bibliothek ermittelt. Andererseits wird eine datengesteuerte Strategie unter Verwendung tiefer neuronaler Netze entwickelt, um die Leistung des vorgestellten zusammengesetzten Diagnoseklassifikators zu vergleichen und zu bewerten. Der datengesteuerte Klassifikator verwendet beobachtete Signale aus dem fehlerhaften und gesunden Betrieb des Systems, um die Leistung des entwickelten Netzwerkmodells zu trainieren und zu bewerten. Die Diagnosetechnik in dieser Dissertation ist unabhängig von den historischen Daten, unabhängig vom Expertenwissen und rechenressourceneffizient. Für die Bewertung wurden vier Arten von Festgefahrener oder Konstantwertiger Fehlern an verschiedenen Komponenten wie Temperatursensor, CO_2-Sensor, Heizungsantrieb und Klappenantrieb mit unterschiedlichen Fehlerwerten zu verschiedenen Zeitpunkten untersucht. Ein Fehlerinjektionssystem injiziert die Fehler künstlich, um die Diagnoseklassifikatoren zu unterstützen, z. B. beim Training der Modelle und bei der Auswertung. Die Ergebnisse zeigen, dass der in dieser Arbeit vorgestellte kombinierte Klassifikator eine vergleichbare Leistung wie die datengesteuerte Methode aufweist und gleichzeitig die Stärken der wissensgesteuerten Methoden ausnutzt.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AHU	Air Handling Unit
AI	Artificial Intelligence
ANN	Artificial Neural Network
ASHRAE	The American Society for Heating, Refrigeration, and Air-Conditioning Engineers
BBN	Bayesian Belief Network
CPS	Cyber-physical System
CPT	Conditional Probabilities Table
DBN	Deep Belief Network
DCV	Demand-Controlled Ventilation
DL	Deep Learning
DM	Degree of Membership
DNN	Deep Neural Network
FBBN	Fuzzy Bayesian Belief Network
FDD	Fault Detection and Diagnosis
FDFD	Failure Detection and Fault Diagnosis
FI	Fault Injection
GUI	Graphical User Interface
HVAC	Heating, Ventilation, and Air-conditioning
IAQ	Indoor Air Quality
ITTM	Intersection Triangular Top Matrix
LMTD	Logarithmic Mean Temperature Difference
LSTM	Long-short-term Memory
MF	Membership Function
MI	Mutual Information
ML	Machine Learning
NN	Neural Network
PPV	Positive Predictive Value
RC	Resistance Capacitance
RDP	Relation Direction Probability
RDT	Relational Data Table
SLT	Subdomain Label Table
SLV	Subdomain Label Vector
SRT	Subdomains Relation Table
TPR	True Positive Rate
UML	Unified Modeling Language
VAV	Variable Air Volume
WFRDT	Weighted Fuzzy Relational Data Table
WSAN	Wireless Sensors and Actuators Network

LIST OF PUBLICATIONS FROM THIS THESIS

1. **Ali Behravan**, Bahareh Kiamanesh, Roman Obermaisser, "Fault Diagnosis of DCV and Heating Systems based on Causal relations in Fuzzy Bayesian Belief Networks using Relation Direction Probabilities," *Energies*, *Special Issue: Fault Identification and Fault Impact Analysis of Ventilation System in Buildings*, 14(20), 6607, ISSN 1996-1073, 2021.

2. **Ali Behravan**, Roman Obermaisser, Mohammad Abboush, "Deep Learning Application in Mechatronics Systems' Fault Diagnosis, a Case Study of the Demand-Controlled Ventilation and Heating System," *In Proc. of the 2019 Internet of Things, Mechatronics and their Applications International Conference (IoTMAC'2019) in Advances in Science and Engineering Technology (ASET'2019) Conference*, Dubai, United Arab Emirates, 2019.

3. **Ali Behravan**, Nadra Tabassam, Osama Al-Najjar, Roman Obermaisser, "Composability Modeling for the Use Case of Demand-controlled Ventilation and Heating System," *In Proc. of the 6th International Conference on Control, Decision and Information Technologies (CODiT 2019)*, Paris, France, 2019.

4. **Ali Behravan**, Roman Obermaisser, Mohammad Abboush, "Fault Injection Framework for Demand-Controlled Ventilation and Heating Systems Based on Wireless Sensor and Actuator Networks," *In Proc. of the 9th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON 2018),* ***BEST PAPER AWARD***, University of British Columbia, Vancouver, Canada, 2018.

5. **Ali Behravan**, Roman Obermaisser, Deepak Hanike Basavegowda, Simon Meckel, "Automatic Model-Based Fault Detection and Diagnosis Using Diagnostic Directed Acyclic Graph for a Demand-Controlled Ventilation and Heating System in Simulink," *In Proc. of the 12th Annual IEEE International Systems Conference (SYSCON 2018)*, Vancouver, Canada, 2018.

6. Ahlam Mallak, **Ali Behravan**, Christian Weber, Madjid Fathi, and Roman Obermaisser, "A Graph-based Sensor Fault Detection and Diagnosis for Demand-Controlled Ventilation Systems Extracted from a Semantic Ontology," *In Proc. of the 22nd IEEE International Conference on Intelligent Engineering Systems 2018 (INES 2018)*, Las Palmas de Gran Canaria, Spain, Spain, 2018.

7. **Ali Behravan**, Roman Obermaisser, Amirbahador Nasari, "Thermal Dynamic Modeling and Simulation of a Heating System for a Multi-Zone Office Building Equipped with Demand Controlled Ventilation Using MATLAB/Simulink," *In Proc. of the IEEE 2017 International Conference on Circuits, System and Simulation (ICCS 2017)*, University of Greenwich, London, United Kingdom, 2017.

8. **Ali Behravan**, Roman Obermaisser, Deepak Hanike Basavegowda, Ahlam Mallak, Christian Weber, Madjid Fathi, "Fault Injection Framework for Fault Diagnosis based on Machine Learning in Heating and Demand-Controlled Ventilation Systems," *In Proc. of the IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI-2017)*, Tehran, Iran, 2017.

1. INTRODUCTION

The effects of climate change on our earth have become a significant concern for human beings in the last decades. In the U.S., the building sector is responsible for almost 40% of greenhouse gas emissions [1]. This issue motivates academia to seek strategies that aim at a decrease in global warming effects. All the efforts alongside the reduction of energy consumption account for this challenge, e.g., less fuel consumption of the central heating system of a building leads to less exhaust gas of the furnace which means less carbon dioxide (CO_2) emission to the atmosphere. For example, the air exchange in the ventilation system during the cold seasons is considered a heating load for the heating system that causes a great increase in the heating system's energy consumption. The energy consumption of buildings is almost 18%-30% of global energy consumption [2, 3]. The building sector in the European Union (EU) is one of the biggest energy consumers, which consumes around 40% of the total energy in the union [4], [5], and [6]. The U.S. Department of Energy denotes that Heating, Ventilation, and Air-Conditioning (HVAC) systems consume 42%-48% of the energy consumed in residential buildings[1], and from this amount, 31%-33% is the share for space heating and ventilation, which confirms the potential of energy saving [7].

On the other hand, the air exchange based on natural ventilation is a cost-efficient method to improve Indoor Air Quality (IAQ) and to dilute indoor CO_2 concentration and odors, or to remove other gaseous pollutants or airborne virus particles, e.g., different variants of the Covid-19, from the building zones. As part of the operating and control systems in buildings, the HVAC systems keep thermal conditions in a comfort zone IAQ in an acceptable range. The CO_2 generated by human metabolism, if it is not correctly being controlled to some limits, can decrease the indoor air quality, deteriorate the occupants' efficiency, create severe mental problems, or cause considerable excess effort for the thinking ability, especially in cold seasons that openings are kept closed most of the time. Thus, implementing a robust ventilation control system for the buildings particularly crowded office buildings is momentous. These statements describe the importance of energy savings and IAQ in the building sector. Therefore, governments allocate funds, and scholars invest massive efforts in this area.

Embedded systems are frequently used as networked components in the HVAC system. Numerous embedded devices with sensors, actuators, Real-Time Operating Systems (RTOS), and microcontrollers are required to implement HVAC systems. The energy consumption of buildings is highly related to the occupancy pattern, the operating and control systems, the outdoor environment, the structure specifications, and the materials. Recent research trends and technologies emerged based on advanced control strategies in building energy systems, which indicate potential energy-saving up to 30% of total energy consumed in a building [8]. Among the diverse HVAC technologies in literature, the adaptive Demand-Controlled Ventilation (DCV) system is a solution with a high potential in energy savings and the provision of a comfortable indoor climate, especially for retrofit purposes. Merema et al. in their results of case studies show that there are significant reductions in energy consumption are achieved by DCV for both the fans (50–55%) and ventilation heat losses (34–47%) as a high energy-saving potential of the DCV systems in office and school buildings [9]. Decreasing the energy demand through energy-efficient solutions in buildings contributes to major sustainability in our society. Therefore, exploiting innovative and adaptive solutions that optimize the system gives the best results. DCV and heating systems are solutions establishing the appropriate balance between reducing energy consumption and supplying improved and comfortable indoor conditions in buildings consider natural ventilation with its unique role in improving the IAQ by dilution of air pollutants such as CO_2 or excess humidity. The DCV is an automatic adjustment of ventilation that modifies the amount of air exchange according to the

[1] https://css.umich.edu/factsheets/residential-buildings-factsheet

demand for fresh air based on sensor measurements of air quality parameters (CO_2 measurements, occupancy, or in some cases, humidity measurements) and need in every zone at any time by the automatic adjustment of damper actuators. A DCV system based on the natural ventilation principle profits the potential energy saving in heating systems by preventing excess outside low-temperature air from entering the building spaces according to demand based on occupancy rates and activities in balance with energy conservation requirements. The DCV and heating system is a promising solution that controls and optimizes the ventilation rates based on thermal comfort and indoor air quality demands with a high potential in energy saving. Natural ventilation is a cost-efficient method to improve IAQ and dilute indoor CO_2 concentration and odors in offices. DCV technology is opposed to traditional codes and standards, which are based on constant values of required air change volume per person or per area for different places, thereby increasing the potential energy saving in heating systems by decreasing the heating load of the system. Studies demonstrate that 15% to 25% of the HVAC system's energy can be saved by setting the ventilation rates based on the occupancy's fresh air requirement [10]. The research shows that the constant values in codes and standards can lead to over ventilation and increased energy consumption [11]. For example, The American Society for Heating, Refrigeration, and Air-Conditioning Engineers (ASHRAE) standard 62 recommended the outdoor air requirements in *cfm per person* for different applications, e.g., 20 cfm/person for office spaces or 0.35 cfm/ft^2 floor area [12]. However, the DCV system aims to keep the indoor CO_2 concentration in a fault-free (healthy) range below 1000 ppm for office rooms [13]. There is a research problem in the study of the DCV and heating systems. Many types of research in literature study the DCV system or the adaptive thermal control separately, or they assumed some recommended values for the air exchange based on the standards, e.g., ASHRAE. However, there is no comprehensive framework to study this complex system dynamically due to its numerous functions, inputs, and outputs for an in-depth assessment of components' functionality and effective parameters besides the adaptive thermal control. The interactions among heating systems, ventilation demand, and thermal demand are complex due to several important parameters, e.g., outside temperature pattern, occupants' behaviors, physical and thermodynamic properties of building components, and the heating system specifications. Improper design of the DCV system accelerates discomfort conditions and air quality degradation by causing undesirable infiltration and energy loss. Simulation tools abstract from reality to track the behavior of the modeled systems over time with advantages against experimental setups, e.g., more efficiency in development time and cost. The models are more complex with new components added into the system or in reconfigurable system architectures. This thesis introduces the simulation framework with modular, composable models of DCV and heating systems for arbitrary architectures based on a high-level specification of the building characteristics.

This DCV and heating system, with its numerous components such as sensors, actuators, and processing units connected in a network, is an example of Cyber-Physical Systems (CPSs). The integration of the continuous time-driven nature of the DCV and heating system with the discrete event-driven nature of Wireless Sensors and Actuators Networks (WSANs) shows the nature of the CPS. These systems include many parameters and input/output signals with a wide range of values. Also, the electronic systems for DCV and heating systems make a level of complexity due to their numerous components such as sensors and actuators. These complex structures cause the systems error-prone, more susceptible to faults and the following failures can affect the efficiency severely. Faults, if they are not detected and diagnosed early, can lead to problems including discomfort, performance degradation, uneasiness for the occupants, and shorten the equipment's life, which might lead to permanent damages to its components and high maintenance costs. Also, faults in HVAC systems can cause a waste of energy and reduce the productivity of the occupants in the building. From the indoor air quality aspect, CO_2 proliferation in office spaces due to human respiration causes adverse effects for occupants' comfort, e.g., drowsiness, increased heart rate,

headache, feeling unwell, lack of concentration, and deterioration in efficiency. Significantly, the occupants' capacity to concentrate declines, and the mental tasks require much effort when exposed to a high concentration of airborne CO_2 for 2 to 3 hours [14]. Lazarova-Molnar et al. depict the annual impact of faults in energy consumption with impacts from HVAC and air distribution faults describing unoptimized HVAC systems with occupancy issues and improper work of dampers in Table 1.1 [15].

Table 1.1 Impact of faults in terms of energy consumption [15].

Fault	Fault type	By percentage	Annual energy consumption [quads]
Duct leakage	Air Distribution	30%	0.3
HVAC left on when space unoccupied	HVAC	20%	0.2
Lights left on when space un occupied	Lighting	18%	0.18
Airflow not balanced	Air Distribution	7%	0.07
Improper refrigerant change	Refrigeration Circuits	7%	0.07
Dampers not working properly	Air Distribution	6%	0.055
Insufficient evaporator airflow	Air Distribution	4%	0.035
Improper controls setup/commissioning	Controls	2%	0.023
Control component failure or degradation	Controls	2%	0.023
Software programing error	Controls	1%	0.012
Improper controls hardware installation	Controls	1%	0.010
Air-cooled condenser fouling	Refrigeration Circuits	1%	0.008
Valve leakage	Waterside Issues	1%	0.007
Total	-	100%	1.0

As large buildings are deployed with complex electronic systems and divided into many sectors, zones, and zones with multiple sensors to manage their operations, faults are not easily detectable. Therefore, tracing the behavior of components and systems to reach caused faults is a challenging task. The services of these systems, such as decision-making, learning, optimization, and control, need dependability to be trustworthy. The probable results due to faults are wrong decisions or catastrophic failures, making the system unreliable and untrustworthy, especially in critical infrastructures such as airports or hospitals. Here, HVAC systems are safety-critical such as in case of fire or toxic gas leakage. Therefore, system monitoring, data collection, and fault tolerance are inseparable parts of HVAC systems. Explicit redundancy can guarantee vital services in case of faults, but it is not a cost-effective solution. Automatic failure detection and fault diagnosis techniques are cost-effective solutions to unwanted faults.

Failure detection and fault diagnosis is the combination of failure detection which is the implication of the fault in a component or system with fault diagnosis that is finding the type, severity, time of occurrence, and locality of faults. The state-of-the-art of fault diagnosis methods for building energy systems, e.g., HVAC systems, reviewed data-driven and knowledge-driven diagnostic methods with their strengths and shortcomings [16].

The knowledge-driven methods are mainly based on expert knowledge and simulate the diagnostic thoughts of domain experts with the argumentation of uncertainties, diagnostic of different fault severities, and understandability; although, with a higher and time-consuming effort that need a deep understanding of the causal relations among system inputs, faults, and symptoms. Moreover, the knowledge-based methods still lack automatic strategies to improve efficiency. The data-driven methods depend on similarities and patterns, with high sensitivity to any change of patterns and more accuracy than the knowledge-driven methods. Expert knowledge combines the knowledge from first principles with causal modeling based on the system description to create

qualitative models based on rules as the central core of the knowledge-based techniques [17]. Expert systems are categorized into a shallow-knowledge expert system with IF-THEN rules of rule-based methods, deep-knowledge expert systems based on functional reasoning, first-principles expert systems, and machine learning. Knowledge-based techniques provide a technique to solve a problem based on data.

However, the data-driven methods mainly rely on similarities and patterns. They are very sensitive to changes in patterns and have more accuracy than the other knowledge-driven based methods. But, they require huge data for training; they cannot provide the reason behind the results, and their strategies indicate black boxes with low understandability. Zhao et al. classified the 135 artificial intelligence-based research papers in the field of FDD in HVAC systems from 1998 to 2018 and the result shows that most papers were about the data-driven-based methods (79%), however, a few papers were about the knowledge driven-based methods (21%) [16]. This shows the relative expertise of researchers in the data-driven-based methods when knowledge-driven-based methods need special attention and new AI-based methods are in demand that can combine the advantages of knowledge-driven and data-driven methods in the future [16].

The research gap filled in this thesis is therefore the combination of knowledge-driven and data-driven diagnosis in DCV and heating systems to gain advantages from both categories. The diagnostic method presented in this thesis initiates low experts' effort without necessitation of in-depth understanding of the causal relations compared to existing knowledge-driven methods with high understandability and high accuracy. The fault diagnosis strategy provided in this thesis is an AI-based method that combines a data-driven classifier with knowledge-driven inference, e.g., fuzzy logic and a Bayesian Belief Network (BBN). The combination of BBN and fuzzy logic that is Fuzzy Bayesian Belief Network (FBBN) analyzes the dependencies of the system signals based on Mutual Information (MI) theory.

In the offline mode, a library of trends and statuses based on training fault cases is established. The conditional probabilities are calculated based on fuzzy weights of signal values and statuses that are used to obtain mutual information. The positive MI values show the dependencies of the subdomains in a pair of measurement signals or statuses for each fault case (pairwise dependency) and negative MI values show that there is no dependency. Then, the conditional probabilities of the subdomains in a pair with positive MI values are calculated and the conditional probability with the higher value indicates the direction of the dependency in each pair of nodes. Then, these dependencies are stored in the offline library as the RDPs. In the online diagnosis mode, our strategy compares the trends and statuses of the real scenario, which can be a fault scenario, with the trends and statuses that are stored in the offline library to find the most similar trends and statuses of signals of the fault case to the trends and statuses of signals for example scenarios in the library. For this comparison, the RDP of the real or faulty scenario is compared to the RDPs in the offline library and the percentages of similarities are calculated. Then, the evaluation step determines and sorts the likely fault causes based on the comparison results from the highest degree of similarity to the lowest as the composed diagnostic classifier result. These percentages of similarities are beliefs sorted from the highest value to the lowest value where larger values imply a higher probability of the corresponding fault. The overall benefits are more understandability, less effort for experts, and higher diagnostic accuracy. Our strategy only needs expert knowledge to define fuzzy sets and the whole process can intelligently classify the faults compared to the other knowledge-based strategies. The evaluation results show that our strategy can accurately map a fault case to the predefined fault in the library.

On the other hand, a data-driven strategy is specifically established using deep neural networks to compare and evaluate the performance of the presented composed diagnostic classifier.

The data-driven classifier uses observed signals from faulty and healthy operations of the system to train and evaluate the performance of the designed network model.

Both strategies are tested on a demand-controlled ventilation and heating system. For the evaluation, four types of stuck-at faults at different components such as temperature sensor, CO_2 sensor, heater actuator, and damper actuator with various fault values at different instants of time were investigated. A fault injection framework artificially injects the faults to serve the diagnostic classifiers, e.g., training the models and evaluations. The results show the composed classifier introduced in this thesis has comparable performance to the data-driven method while advantaging the strengths of knowledge-driven methods. The evaluation results show that 97.22% of faults were truly diagnosed in the composed diagnostic classifier with better precision, F-score and accuracy compared to a deep neural network as a data-driven diagnostic method.

All the contributions mentioned above in this thesis are designed based on a modular composability scheme. Therefore, the introduced model in this thesis can combine modules and understand the composition of diverse subsystems by having a whole perception of its components and combinations. Composability is applicable in many fields, such as cyber-physical systems that include Systems of Systems (SOS) and compositions of diverse subsystems. Modularity is a design technique for subsystems (modules) with well-defined interfaces to create scalable models that can be used in various contexts. The unique modules are used to create various complex building models. This thesis introduces the modular composability scheme using generic diagnostic components at the system level applicable in many areas. That means the model, thanks to the modular composability algorithm, is scalable. The user can generate different types of building with various architectures with many rooms and floors. The system model, fault injection blocks, and diagnostic modules will be automatically extended.

1.1. THESIS OBJECTIVES AND CONTRIBUTION

Nowadays, fault diagnosis in a large-scale system is a crucial challenge. This thesis aims to provide a novel fault diagnosis framework based on composed diagnostic classification for specific fault types based on the causal relations in an FBBN using RDPs beside another fault diagnosis framework based on the data-driven classification with a deep neural network to use for the diagnosis of a range of faults and to be used to evaluate the composed diagnostic classifier results for the DCV and heating system model. The main reasons behind the selection of the DCV and heating system in this thesis are the high potential in energy-saving, the decrease of CO_2 patterns that are detrimental to the occupants, and the guarantee of the indoor air quality and thermal comfort of occupants. This system with its numerous components such as sensors, actuators, and processing units connected in a network integrates the continuous time-driven nature of the environment with the discrete event-driven nature of the computational world. The implementation part of this thesis presents a framework that was analytically modeled at first and then simulated in MATLAB/Simulink using the Simscape toolbox to monitor and control IAQ and comfort parameters in an office building in real-time and study the fault diagnosis methods. Also, this framework supports the modular composability scheme. The proposed framework offers a modular adaptive thermal model of the heating system for a multi-zone office building equipped with the DCV system and diagnostic modules. The modular composability scheme uses generic diagnostic components developed in this thesis for each zone and fault injection blocks for testing and evaluation.

The contributions of this thesis can be summarized as follows:

1. ***Modeling and implementing the DCV and heating system:***

   The thesis models and simulates a DCV and heating system applicable in many kinds of research. This adaptive thermal control system considers the airborne CO_2 emissions due to the

occupants or any other sources in building spaces with minimum energy consumption while optimizing the utmost indoor air quality, occupants' thermal comfort, and energy demands and prevents overheating and overventilation that can be the reason for energy waste and occupants discomfort. Many types of research in literature study the DCV system or the adaptive thermal control separately, or they assumed some recommended pre-set values for the air exchange based on the standards, e.g., ASHRAE. From this perspective, there is no framework to study this complex system including DCV and heating control dynamically with its numerous functions, inputs, and outputs for an in-depth assessment of components' functionality and effective parameters besides the adaptive thermal control. The interactions among heating systems, ventilation demand, and thermal demand are complex due to several important parameters, e.g., outside temperature pattern, occupants' behaviors and CO_2 generation, physical and thermodynamic properties of building components, and the heating system specifications. The model developed in this thesis is useful in sensitive applications especially during the pandemics, e.g., office buildings, schools, hospitals, pharmaceutical factories, manufacturers of electronic components, or poultry farms, with some modifications in the type of sensors and actuators. Also, this framework can be extended to other gaseous pollutants, e.g., Carbon Monoxide (CO), Nitrogen Dioxide (NO_2), Ozone (O_3), Methane (CH_4), and Particle Pollution (PM). The model is analytically modeled at first and then simulated in MATLAB/Simulink.

2. ***Developing an automatic composed diagnostic classifier based on the combination of data-driven and knowledge-driven diagnosis methods:***

   A new automatic fault diagnosis method is developed in this thesis as a composed diagnostic classifier to solve the fault diagnosis problem of the DCV and heating systems by involving a knowledge-driven based diagnostic system as a white-box approach when maintaining the advantages of the inference-based methods, e.g., Bayesian inference-based and fuzzy inference-based methods, in combination with the automatic data-driven-based fault diagnosis using multiclass classifiers for stuck-at or constant fault types. The composed diagnostic classifier relies on the causal relations in a fuzzy Bayesian belief network using relation direction probabilities to lower the fault diagnosis efforts and energy waste as well as to increase the operational efficiency. The method in this thesis is based on the integration of causal relations using the relation direction probabilities and fuzzy theory into the Bayesian belief networks. The RDPs were used to update the fault library. In offline mode, for each fault class, a Relation-Direction Probability (RDP) table is computed and stored in a fault library. In online mode, we determine the similarities between the actual RDP and the offline precomputed RDPs. The combination of BBN and fuzzy logic in our introduced method analyzes the dependencies of the signals using MI theory. The method creates a unique RDP table for each class of faults and data sets. This method can also be extended to additional faults by adding RDPs of new fault classes to the offline library. This method provides more understandability, less effort for experts, and higher diagnostic accuracy while it is independent of the historical data, independent of the expert knowledge, and computing-resource efficient. Our strategy is less dependent on expert knowledge and only requires the expert to define fuzzy sets and the whole process can intelligently and automatically classify the faults compared to the other knowledge-based strategies. The evaluation results show that our strategy can accurately map a fault case to the predefined fault in the library.

   This diagnostic method automatically reveals the hidden and intrinsic dependencies of trends (for sensors) or statuses (for actuators) in signals that change concurrently over time in case of faults. For example, if a damper sticks at open status, the room temperature decreases and makes the heater stick at ON status indirectly because the heater wants to compensate for the heating load due to the damper, which is a hidden dependency between damper and heater

status signal. Also, this diagnostic method finds fault-symptoms dependencies and diagnoses faults in complex systems with many signals.

3. ***Developing a failure detection and fault diagnosis framework based on the data-driven classification using the deep neural networks:***

   This thesis developed a pure data-driven based fault diagnosis method to solve the fault classification problem in a multi-class model to evaluate the performance of the introduced automatic composed diagnostic classifier (contribution No. 3) in this thesis. This system relies on a big amount of data based on a black-box approach. This system includes different steps of data acquisition, data preprocessing, network model design, model optimization, and network model evaluation. The evaluation results from this method will be compared with the other diagnosis method described in this thesis.

4. ***Development of fault injection blocks for fault injection and GUIs for fault activation and system monitoring:***

   The building blocks and GUIs are developed for the users in this thesis for evaluating the robustness, e.g., against various faults and failures, by artificially injecting different types of faults based on the time of fault injection, severity, type, and domain. The data produced in this framework is used to test various diagnostic classifiers. The simulated model can be connected to the data server using the structured query language for the data logging.

5. ***Modular composability scheme:***

   All the contributions mentioned above in this thesis are designed based on a modular composability scheme. Composability is applicable in many fields such as cyber-physical systems that include systems of systems and compositions of diverse subsystems. Modularity creates scalable subsystems (modules) with well-defined interfaces. The unique modules are used to create various complex building models. This thesis introduces the modular composability scheme using generic diagnostic components at the system level applicable in many areas. Therefore, the introduced model in this thesis can combine modules and understand the composition of diverse subsystems by having an entire perception of its components and combinations. The user can generate different types of building with various architectures with many rooms and floors. The system model, fault injection blocks, and diagnostic modules can be automatically extended.

1.2. STRUCTURE OF THE THESIS

This report is organized into eight chapters. Chapter one provides the introduction focusing on the motivation, challenges, requirements, thesis objectives, and contribution. The contents of the remaining chapters are summarized below.

Chapter two provides the related works and basic concepts discussing the state-of-the-art of the main parts of this thesis for DCV and heating systems simulation and failure detection and fault diagnosis in DCV and heating systems containing knowledge-driven and data-driven diagnostic methods.

Chapter three formulates the theory of the system model, including three parts: physical model, DCV, and modular composability.

Chapter four describes the implementation part that consists of modeling and simulation of the system model and the modular composability scheme.

Chapter five explains faults and fault injection framework implementation, including modeling and simulation.

Chapter six describes the failure detection and fault diagnosis techniques used in this thesis, including a novel automatic composed fault diagnosis classifier that combines the knowledge-driven inference and data-driven automatic classifier based on the causal relations in FBBNs using RDPs. Also, a pure data-driven fault diagnosis method is introduced based on the deep neural networks to evaluate the automatic composed diagnostic classifier introduced in this thesis.

Chapter seven explains the evaluation of both pure data-driven and composed diagnostic classifiers using the fault injection framework by comparing the performance metrics.

The thesis ends with chapter eight that is the conclusion and further research works.

2. BASIC CONCEPTS AND RELATED WORK

This chapter presents the basic concepts and introduces the related work relevant to this thesis.

2.1. BASIC CONCEPTS OF COMPONENT-BASED SYSTEMS

A **system** is an entity that interacts with other entities, i.e., other systems such as hardware, software, and humans [18]. The **behavior** of a system is what the system does to implement its function [18]. The **system's function** is what the system is intended to do [13], and **the system's behavior** perceived by the users describes its **service** [19]. The **user** is a physical or human system interacting with another system at the service interface [19].

Two terms of compositionality and composability are the most used ones for component-based systems. **Compositionality** shows the overall system properties from the properties of its components [20]. Compositionality refers to combining modules and understanding the composition of diverse subsystems by having a whole perception of its components and combinations. **Composability** is a system design principle with a focus on the inter-relationships of components. Composability describes that the component property will be valid under some local conditions after composition, which guarantees the preservation of previously established component properties[20]. Composability maintains the stability of component properties across merging in an environment (when its environment changes by adding or removing components)[20]. Composability is a technique that enables the selection and assembling of components in the flexible forms of several combinations while keeping the specific user requirements [21]. Composability aims to increase the reliability and cost-effectivity of a system [22].

There are terms to show the trustworthiness of the system that are verification and validation. **Verification** is the process of checking whether the system adheres to given properties or the verification conditions [19]. **Validation** means checking the specification of a system [19].

2.2. BASIC CONCEPTS FOR DEPENDABILITY

The failure detection and fault diagnosis concepts are related to the dependability measure and its attributes. **Dependability** is defined as the ability to deliver a service that can justifiably be trusted [14]. Dependability has attributes including **availability** that is defined as readiness for usage, **reliability** as continuity of correct service, **safety** as the absence of catastrophic consequences on the user(s) and the environment, **integrity** as an absence of improper system state alterations, and **maintainability** as the ability to undergo repairs and modifications. **Robustness** is maintaining the dependability of the system despite erroneous inputs.

Fault diagnosis is a prerequisite of a reliable system as it is essential for creating a fault-tolerant system. Figure 2.1 shows the chain from cause to effect view in the physical system and from observation to diagnosis in an Automatic Fault Detection and Diagnosis (AFDD) system [23].

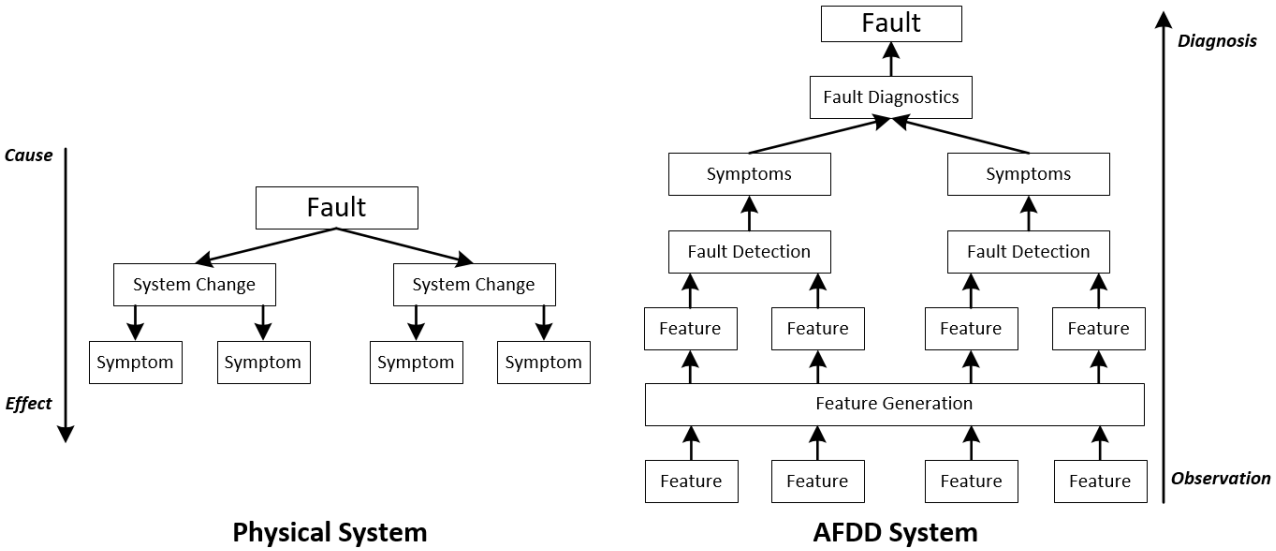


Figure 2.1 Overview of an AFDD system – from observation to diagnosis [23].

Fault Detection is a term used to "determine the occurrence of faults in the functional units of the process", which leads to the incorrect, unacceptable, or abnormal operation of a system in some respects by continuous monitoring of the behavior of the system [24], [25], [26]. In most failure detection techniques, discrepancies between measured variables and nominal values (features) indicate failures at the system level. Detecting such a failure can be made using different techniques ranging from simple static or dynamic thresholds to complicated decision-making algorithms.

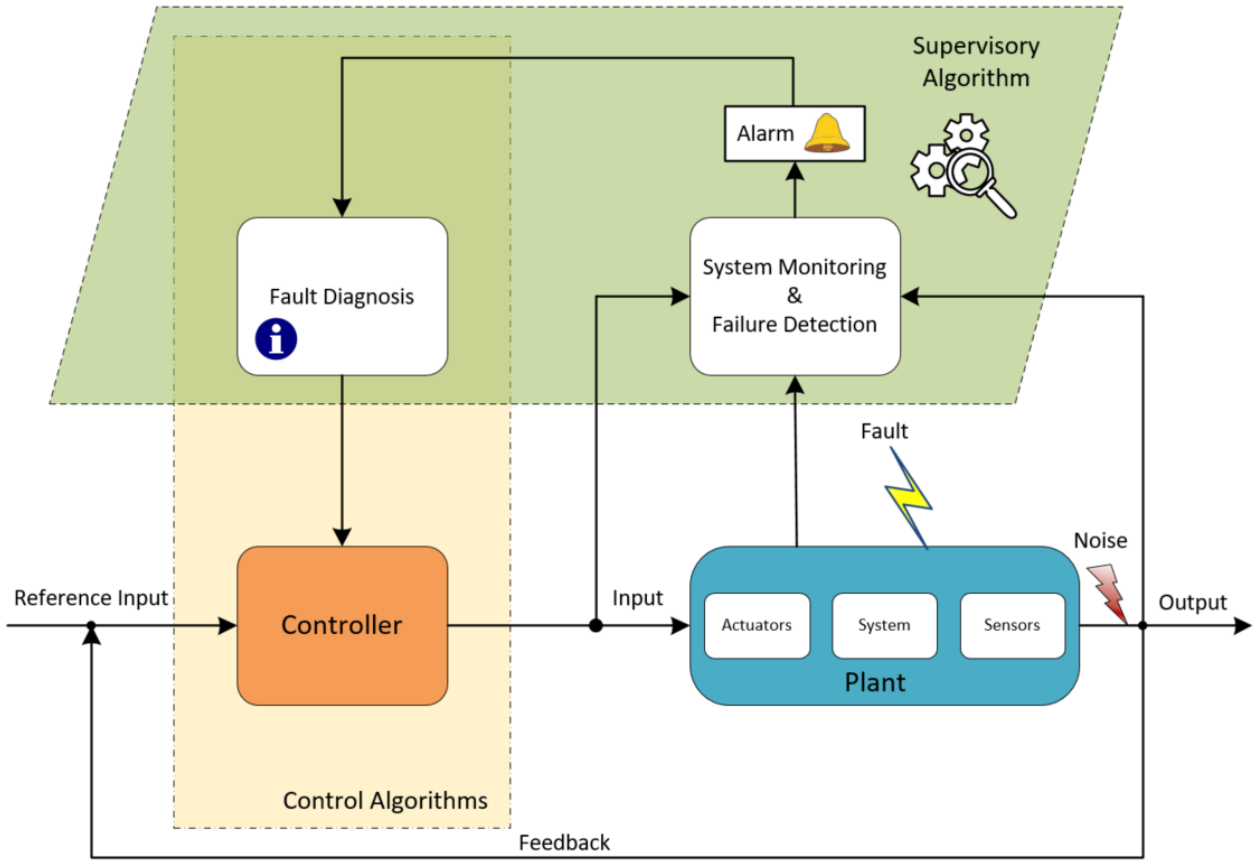


Figure 2.2 Failure detection and fault diagnosis architecture.

Figure 2.2 provides a global view of architecture for failure detection and fault. The supervisory-level algorithm that includes system monitoring, failure detection, and fault diagnosis enables the system to overcome the faulty plant and handle the controller.

Failure Detection and Fault Diagnosis (FDFD) is the investigation of detecting the failure in a system and diagnosing relevant fault(s) that is the root cause of the aroused failure in the aspect of the type, size (intensity), locality, and time. The FDFD in this thesis has the same meaning as FDD in other research studies because this thesis considers the fault at a system level as a failure at a component level. Then, the diagnostic information can be used for taking corrective actions. The terms "fault identification," "fault isolation," and "fault analysis" are also referred to as fault diagnosis [23]. Diagnosis is a forerunner to fault-tolerant control. Fault diagnosis uses the symptoms generated from the failure detection stage to make the diagnostic process more manageable. These symptoms are categorized into *analytical* symptoms from FDFD algorithms or *heuristic* symptoms generated by human observation based on expert knowledge. In a successful FDFD process, component failures are detected early before a severe system failure, damage, or loss of the service, and their fault causes are diagnosed [10].

Establishing a precise diagnostic model is a challenging task accompanied by try and error efforts considering many diagnostic methods [24]. An ideal FDFD system should have the following characteristics that are extended for this thesis [23]:

- **Low cost:** An ideal FDFD system should be economical, which is now possible thanks to computation and sensing resources. However, the expert knowledge required in the setup stage of the FDFD system is still costly. Therefore, low dependence on the expert-knowledge means decreasing the cost.
- **Reliable:** An ideal FDFD system must retain its reliability.
- **High truly diagnosis rates:** The performance characteristics of an FDFD system must be evaluated, such as accuracy, precision, recall, F-score, true positives, false positives, true negatives, and false negatives. The higher diagnosed fault rates are associated with lower costs and maintenance efforts, high trustworthiness, in addition to saving in operation resources.
- **Composable:** An ideal FDFD system should overcome the composability limitations due to system characteristic changes to automatically adapt and eliminate updating efforts of diagnostic algorithms.
- **Automatic configuration:** it is desirable to have an automatically configurable FDFD system that adaptively selects and fits many building architectures considering identical or similar systems.

The complex nature of HVAC systems due to interconnected components makes the FDFD approach in this context highly challenging. The fault-error-failure propagation model is used to picture the relationship between the interconnected components' faults, errors, and failures. Figure 2.3 illustrates how an activated error due to fault with internal or external resources propagates either at the component level into one or more components or at a higher level into one or more systems based on the process hierarchy [27].

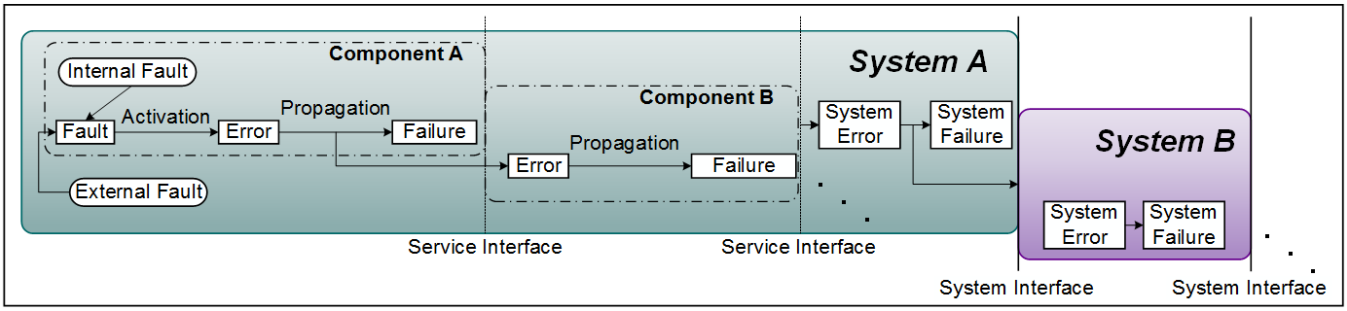


Figure 2.3 Fault-Error-Failure Propagation Model at Component Level and System Level [27].

The internal fault causes are located inside the system, while the external causes come from the system's environment. The activated fault causes the error in the component, and the error can propagate either internally in the component to cause a component failure or externally to another component of the same system via the service interface or another system via the system interface. In the former case, the service delivered to the other component becomes incorrect, e.g., an incorrect service produced by component A is delivered to component B. Henceforth, the error in component A causes an error due to an external fault in component B. This error can again propagate into that component by creating a component failure or moving to another component of the same system and creating system failure in the same system or another system as its final destination.

"A **fault** is an unpermitted deviation of at least one characteristic property of the system from its normal, acceptable, usual and expected behavior that leads to an error and symptoms" [28]. Himmelblau defines the term fault as "a departure from an acceptable range of an observed variable or a calculated parameter associated with a process" [29]. Faults may result in binary variables, e.g., OK/failed, or to some extent in values depending on the severity of the fault. A **symptom** is an observed event or value from the input/system/output signal that is needed in the fault detection process. An **error** is generally an incorrect or undefined system state that may later manifest its effect in the shape of failure [30]. "A **failure** is a permanent interruption of a system's ability to perform a required function under specified operating conditions" that result from one or more faults [28]. The failure is either at the component or the system level. The fault is the reason for the failure. In case of a failure, the system functionality deviates from its specification [27]. The components may encounter abrupt failure or degradation failure.

Fault Injection (FI) is the technique of inducing artificial fault into the system to trace the system behavior in case of occurring or absence of fault [27]. Fault injection techniques are commonly used to test the diagnostic techniques. The faults defined during the fault injection are intentionally injected, developed, and identified using a suitable fault injection framework with reproducing ability. The simulation result and behavior of different systems and components can be observed in faulty and non-faulty (healthy) conditions.

The delivery of the correct service in the presence of faults is guaranteed by **fault tolerance**. It describes how to ensure service up to fulfilling the system's function in the presence of faults [31]. It is generally implemented using error detection, recovery, and fault handling, including fault diagnosis, fault isolation, system reconfiguration, and system reinitialization. **Fault diagnosis** identifies and records the cause(s) of error(s) in terms of both location and type [19]. **Fault isolation** performs physical or logical exclusion of the faulty components from further participation in service delivery, i.e., it makes the fault dormant [19]. **Fault prevention** describes how to prevent fault occurrence or introduction [31]. **Fault removal** describes how to reduce the presence (number, seriousness) of faults [31]. **Fault forecasting** describes how to estimate the present number, the future incidence, and the consequences of faults [31].

2.3. BASIC CONCEPTS FOR MUTUAL INFORMATION THEORY

Mutual Information is a concept rooted in information and probability theory. MI of two random variables is a statistical measurement of the mutual dependency of two random variables. MI also measures the amount of information about one random variable by observing the other random variable [32]. There are many definitions of random variables. For example, G. Zeng [33] has classified MI definitions into two categories: 1) definitions with random variables and 2) ensembles. The original definition is appointed to Shannon [34] that is described in the equation below.

$$H(x) = -\sum_{i=1}^{n} p_i \log p_i \tag{2.1}$$

In this equation, x is a chance variable with a probability sequence of $p_1, p_2, \ldots p_n$ and $H(x)$ is the entropy of x. The joint entropy of x and y is defined as:

$$H(x,y) = -\sum_{i,j} P(i,j) \log P(i,j) \tag{2.2}$$

Entropy in information theory is a logarithmic measure for the rate of information transfer in a particular message. This information rate (R) in Shannon's definition is defined as:

$$R = \sum_{j,j} P(i,j) \log \frac{P(i,j)}{P(i)p(j)} \tag{2.3}$$

In this equation, $P(i,j)$ is the joint probability density function. As mentioned, the first category of mutual information depends on the joint distribution of two random variables. Kullback [35] defines MI, assuming that two random variables have different probability spaces. Ash [15] assumed that the x and y variables have the same probability space. Cover and Thomas [36] have a modern definition of MI in which they have presented MI without probability spaces as shown:

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{2.4}$$

Random variables are replaced with ensembles in the second category of MI definitions, and an average of MI will be calculated. The mutual information of two continuous random variables X and Y called differential mutual information and is defined as:

$$I(X,Y) = \iint P(x,y) \log \frac{p(x,y)}{P(x)P(y)} dxdy \tag{2.5}$$

Where $P(x,y)$ is the joint probability function of X and Y . $P(x)$ and $P(y)$ are density functions for each x and y. Defining the density function of continuous variables is a critical issue. The fuzzy theory can be used an appropriate likelihood density function [37]. Therefore, the MI can be defined for two fuzzy subsets of A and B as follows:

$$MI(A,B) = P(A,B) \log_2 \left(\frac{P(A,B)}{P(A) \times P(B)} \right) \tag{2.6}$$

The probability values of $P(A)$, $P(B)$, and the joint probability value of $P(A,B)$ are calculated in this thesis in sections 6.1.1.4 and 6.1.1.5.

2.4. STATE-OF-THE-ART IN HVAC SIMULATION

The energy demand in the building sector reports that the share of the residential buildings in energy use is almost 21% of the total energy consumption on earth [38]. The buildings' energy consumption is about 40% of the total energy consumed in the European Union (EU) [4], [5]. The office buildings own almost 18% of this energy consumption among the whole building sector [2]. The heating and cooling systems consume approximately 68% of the energy used in buildings in Europe, which is almost 5% of the total energy consumption in the EU [39]. In the United Kingdom, space heating exclusively consumes 61.3% of the energy in the household sector [40]. In the last decade, the new advanced control strategies and their associated approaches for the building energy systems are proposed because they potentially save around 30% of the energy consumed in buildings [8]. Therefore, the study of HVAC and its control strategies is very significant.

Building models highlight new control approaches as they can be modeled and simulated to predict their behavior in the buildings for development purposes. The other benefit of the modeling and simulation is the testing and optimizing of the systems in their design phase [41]. Also, simulation improves weaknesses and consequences before experimental implementations.

MathWorks Inc. [42] has created a simple example of a "House Heating System[1]" in MATLAB/Simulink, which only includes a heater, a controller, and a house structure with four radiators and four rooms, and the heat is transferred within the environment through its exterior walls, roof, and windows. The control function only considers the overall average temperature of rooms, and if it falls below 18 degrees in centigrade, it turns the heater of the entire building on, and if the temperature exceeds 23 degrees, it is turned off. This example model has many shortcomings as it is assumed that heat is not transferred internally between rooms which makes the model far from reality as the adjacent zones have significant thermodynamical impacts on each other, which cannot be ignored. The other shortage of this model is that there is no control in each specific room, which means the waste of energy because some rooms may be empty of occupants, whereas the heater of that room produces useless heat. Moreover, the model does not investigate the ventilation effect, which significantly affects the heating load of the heating system in winter times.

It is essential to learn about system behavior by analyzing the system model to optimize a complex building automation model because it specifies what a system does, including the grasp of energy generation and losses dynamics. Lapusan et al. illustrated a multi-room building thermodynamic model based on the lumped capacitance method with the 3R-2C network, including three resistors and two capacitors using the Simscape library in MATLAB/Simulink [43]. The developed model includes four rooms with an ON/OFF heating control system with a significant limitation when the building model must be extended. Because the connections between zones are the heat flows which makes the model very complex. However, the calculations of the heat transfers in this thesis are based on the calculated temperatures in each room.

Karmacharya et al. developed a simplified building heating system using the lumped node method [44]. In the lumped node method, the construction elements such as walls, windows, and floors are divided into different numbers of elements having uniform temperatures about which an energy balance can be represented. Karmacharya et al. considered the heating system's type of variable flow with a constant temperature. That means the heater flow rate is continuously changing. However, this study considers the heaters as the constant power source. Therefore, the output of the heater, once it is on, is a constant value. Also, in the validation part, Karmacharya et al. use comparative testing where the results from one program are compared to another program. They used the results gathered from the model in MATLAB and compared them with the results acquired from another program named the Design-Builder Software. Gouda et al. [45] describe an optimized second-order lumped parameter method that is used as a thermal building model. This model can represent the wall, floor, or roof of multiple layers. They used a nonlinearly constrained optimization method for the reduction of the model order of building elements. The result of their work shows that a reduced model based on a 2^{nd}-order building element is accompanied by a minimal loss of accuracy but considerable improvements in computational effort. Crabb et al. [46] developed a model in which a room air capacitance and a single lumped construction capacitance were linked in a conduction path with three resistances. The inputs are ventilation rate and zone heat inputs. Sukla and Jenkins [47] have used the lumped node method to model a house, taking heat loss and thermal capacity into account without considering solar radiation, ventilation, and gains.

Mendes et al. [48] focused only on the mathematical model of building thermal analysis and control system design. They have considered a building envelope with three envelope layers, room air, sensor, and heater, and the lumped approach to model the room air temperature and a multi-layer model for the building envelope. The lumped capacitance method enables the study of the transient analysis of room air temperature in case of environmental temperature variations. MATLAB/Simulink is used for evaluating system performance. For this, the energy conservation and energy balance equations and the heater and the temperature sensor models were described, and then, the building model in the format of a state-space problem was implemented in

[1]https://de.mathworks.com/help/physmod/simscape/examples/house-heating-system.html

MATLAB/Simulink. Therefore, the state model consists of 21 state variables. They also studied the parameter of heater performance coefficient (η) in two cases of no radiation of the heater. The first case (emissivity 0) shows that all the heater energy will be consumed to warm up the room air, and the second case is for the emissivity of 0.8. The results show that the heater performance coefficient is 99.99% in the no radiation case and 99.97% with the emissivity of 0.8. Bertagnolio et al. used the 1st order lumped node method to model building elements [49]. Embaye et al. [50] focused on the effect of changing the radiator inlet flow strategy from constant flow to pulsed flow on energy consumption of a radiator in a hydronic centrally heated building in MATLAB/Simulink. The mathematical models of the single heater (radiator) and the room's envelope for this study include walls and windows. The Logarithmic Mean Temperature Difference (LMTD) method was used to calculate the output energy of a hydronic radiator. Then, the energy balance equation was described. However, the constant rates for air changes per hour and the volume flow rate of air infiltration were considered for the ventilation and infiltration parts. The other shortage of this model is modeling and simulation of just one room instead of considering a multi-room building with interconnections. Also, using the Proportional Integral Differential (PID) controller was studied, which causes a decrease in the fluctuation of the room temperature signal.

Thavlov et al. [51] indicated a dynamic heat model that predicts the indoor air temperature and power consumption of electrical space heating in an office building using stochastic differential equations. This model was developed based on the lumped method in the simulation environment of SYSLAB. However, the heat transfer between different building zones is neglected in this model, and only the overall indoor temperature of the building can be calculated. Thavlov showed that due to the high amount of natural ventilation in FlexHouse, especially the nonlinear properties of wind, conditions should be integrated into the model due to their influence on the indoor temperature. Danza et al. [52] focused on developing a Resistance-Capacitance (RC) thermal model using the coupling of the 3R-2C modules for a test cell and the simulation of the model in a Modelica environment. The Differential/Algebraic Equations (DAE) developed by Danza et al. have been solved by the DASSL step-variable solver, which uses backward differential formulas to integrate DAE [53]. Bastida et al. [54] developed a dynamic thermal model of a building with multi-layer walls based on this thesis using state-space and transfer functions. However, they studied just a single-zone building. The type of controller is PI. Mai et al. [55] studied a three-zone building and developed the thermal R-C network, the authors of that study took into account the input heating/cooling rate to each zone calculated by knowing the coefficient of performance (COP) of the HVAC equipment. The research describes that the lumped capacitance method is a good candidate for experimental implementations and studies. Dimitriou et al. [40] experimentally developed a suitable thermal model for domestic buildings which fits with intelligent home equipment such as wireless control products or sensors to monitor and predict the energy demand. The grey-box model was used based on the lumped parameters.

However, none of the mentioned works studied the indoor air quality parameters and pollutants that negatively affect the occupants' life and work quality, such as CO_2 concentration in the building zones. Natural ventilation is considered an efficient technique to improve IAQ by diluting indoor air pollutants such as CO_2 emissions. Therefore, some researchers used mathematical modeling to investigate the ventilation system, which can be simulated as a dilution technique. In this context, Li et al. [56] focused on modeling indoor CO_2 concentration based on real-time indoor occupant prediction and CO_2 generation rates for an air-based HVAC system using ducts connected to a boiler (or chiller) as a central plant. However, their control approach was different as they assumed a constant volumetric airflow rate, a constant supply of CO_2, and constant indoor CO_2 generation. Also, they used the experimental data first to predict the indoor CO_2, and then the (forced) ventilation rate was continuously changed to keep the indoor CO_2 concentration at a constant value, e.g., 600 ppm.

There are some standards as references for an acceptable indoor air quality, such as ASHRAE Standard 62-1999 that provides some fixed values for the ventilation rate based on the building's occupants. However, the number of inhabitants differs continuously in reality [57]. In contrast, demand-controlled ventilation brings its energy-saving potential to the application by preventing excess fresh air exchange that accompanies more heating load. Brandenmuehl and Braun show 15% to 25% energy saving potential of the DCV system using ventilation rates setting based on the occupancy requirements in their study [11].

There are many kinds of research in the literature of Wireless Sensors Network (WSN) simulation frameworks without considering the actuation on the control plant [58], [59], and [60]. Guinard et al. describe a scalable single-hop WSN tool using Borland C++ for the sensor information collection in building energy management applications. However, their study did not include an actuation control system [61]. Song et al. studied the DC servo control plant and provided a real-time simulation model for WSAN based on the IEEE 802.15.4 communication protocol using TrueTime [62]. Callaway et al. described the role of the ZigBee/IEEE 802.15.4 protocol in intelligent homes [63]. Wang et al. mentioned the usability and necessity of using ZigBee technology in intelligent home energy management [64]. Aziz et al. researched a wireless system to monitor and process the room temperature and humidity using ZigBee [65]. Hyncica et al. presented the performance of ZigBee in a small office building and described important features of ZigBee communication [66]. Therefore, this thesis focuses on developing a DCV and heating system framework based on Wireless Sensors and Actuators Network (WSAN) architecture.

2.5. State-of-the-art in Fault Detection and Diagnosis for HVAC Systems

Fault Detection and Diagnosis (FDD) for building systems ensures dependability, efficiency, and economy. Dependability avoids service failures that would be more frequent and severe than the acceptable range [18]. For this reason, dependability attributes should be maintained, i.e., availability (readiness of correct services), reliability (continuity of correct services), safety, integrity, and maintainability. The rise of microcomputers and digital control is considered the earliest effort on automatic FDD for buildings in the 1980s [67], [68].

Isermann [28] categorized the fault diagnosis approaches into inference methods based on causalities between faults and respective symptoms, e.g., fault trees based on decision trees and IF-THEN statements using expert knowledge, and classification methods based on training when fault-symptom causalities are not prominent. The inference techniques are helpful when the causal relations between faults and symptoms are known. Otherwise, classification techniques such as artificial intelligence black-box models can be applied based on experimental data collected from the system to extract features without expert knowledge and find causal relationships. Gao et al. proposed other FDD methods in three categories of model-based methods, data-driven methods, and rule-based methods [69]. Zhao et al. [16] review 135 artificial intelligence-based papers from 1998 to 2018 and classified fault diagnosis methods into two subcategories of data-driven-based and knowledge-driven-based methods. Methods in the knowledge-driven-based category are based on the domain experts' thinking power and depend on the domain knowledge's and inference approaches' nature in FDD methods. Against, methods in data-driven-based subcategory typically are based on the patterns' similarities. Figure 2.4 shows this classification. This thesis focuses on both data-driven-based and knowledge-driven-based diagnostic methods. From the knowledge-driven-based method, a combination of the Bayesian inference-based and fuzzy-inference-based diagnosis methods, and from the data-driven-based methods, the multiclass-classification-based method are used to develop our strategy.

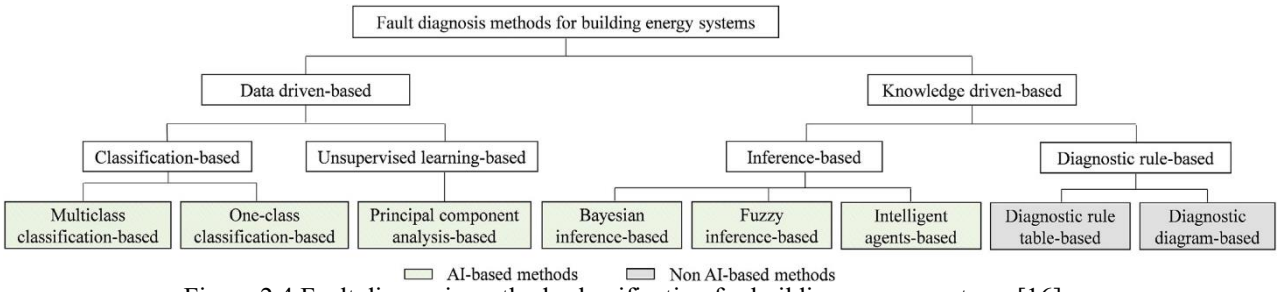


Figure 2.4 Fault diagnosis methods classification for building energy systems [16].

FDD in HVAC systems obtained the attention of many researchers as improper control strategies can come from poor maintenance or system malfunction and malfunctions cause an estimated energy loss between 15% and 30% [26], [70]. The National Institute of Standards and Technology (NIST) illustrates that there is an energy-saving from 10% to 40% with successful fault FDD techniques when a fault is detected at an early stage before causing an unbearable break down [71]. For example, if a damper is stuck at a constant position, significant energy can be wasted depending on the system's functionality or cause comfort issues. Lee et al. studied the Variable Air Volume (VAV) box stuck-at fault of a damper in cooling mode accompanied by 36% more cooling energy consumption [72].

Therefore, reliable FDFD techniques are necessary for an energy-efficient and reliable operation of DCV and heating systems. Implementing FDD promises a better system lifetime and increased indoor air quality [70]. Various FDD techniques are deployed in different applications to detect and diagnose associated faults in embedded components such as sensors and actuators or their network fabric.

The simulation environments are valuable platforms to study and test or compare the effectiveness of different diagnostic methods for different system-level or component-level faults. Sterling et al. analyze failures by comparing qualitative and quantitative model-based diagnostics using Modelica [73]. Basarkar et al. identified and detected some common faults in HVAC systems by simulating the faults in the EnergyPlus tool. However, there is a limited capability of modeling faults in EnergyPlus [74].

Recent advances in Information and Communications Technology (ICT), especially in embedded systems, enable the development of Cyber-physical Systems (CPS) that profoundly couple our physical world to the computation world. The term CPS refers to the integration of the computation world, performed by embedded computers, with physical processes aiming at monitoring and control [75]. Physical processes affect computations and vice versa via feedback loops. The CPSs include components ranging from various sensors to measure the environmental and system parameters as the input signals to the cyber domain to different types of actuators that insert the processed command to the physical process or environment. The common characteristic of the CPSs is the tight integration of hardware and software capabilities and constraints, including advanced sensing and actuation, as well as the combination of heterogeneous components in several subsystems [76]. The CPSs can include many variables, signals, look-up tables, components with continuous and discrete dynamics [77]. The complexity of a CPS increases when it becomes more extensive with more components and equipment that cooperate simultaneously. HVAC systems, e.g. DCV and heating systems, are examples of CPSs. This level of complexity in HVAC systems makes them error-prone, more susceptible to faults, that may lead to waste of energy, e.g., continuous heating in case of a stuck-at damper, poor thermal comfort, and unacceptable indoor air quality. Therefore, the occurrence of faults is unavoidable, and faults, e.g., stuck-at or constant faults, occur inevitably.

Despite the inherent complexity of DCV and heating systems, their applications require them to be fault and failure-tolerant. A fault and failure-tolerant design of DCV and heating systems

require developments in failure detection and fault diagnosis techniques. Detection and diagnostic techniques' testing and evaluation activities are thus of critical importance for the early detection of problems in the models in the design phase before they propagate to the actual DCV and heating systems. The failure detection and fault diagnosis in early stages of occurrence avoid threatening situations, degradation of system performance, energy loss, or discomfort conditions [69]. In HVAC systems, for instance, the faults can be the reason for energy waste up to 20% of total energy consumed, excess pollutant emissions, and decremented comfort for occupants [78], [79]. Basarkar et al. describe that faults based on the type and their severity can be the reason for up to 22% of the total energy consumption of HVAC systems [74]. In another study, Wu shows that faults in HVAC systems account for 20% of total energy consumption [78]. In HVAC systems, probable faults in building systems are various. For example, ASHRAE Project 1043-RP shows that a typical water-cooled centrifugal chiller can face more than twenty types of common faults [80]. ASHRAE Project 1312-RP indicated 68 types of common faults for a typical air handling unit [81]. Therefore, it is costly to capture sufficient training data for every fault, and most of the research projects consider only a part of these faults in most data-driven-based chiller FDD methods.

Many fault diagnosis methods have been mentioned widely in many studies. Steinder et al. [82] have focused explicitly on fault localization techniques in complex communication systems to find the exact source of a failure from a set of failure indications. They have classified the fault localization techniques into three categories of (1) Artificial Intelligence (AI) techniques, including rule-based, model-based, and case-based systems, (2) model traversing techniques. And (3) fault propagation models, including code-based techniques, dependency graphs, Bayesian networks, causality graphs, and phrase structured grammars.

Artificial Neural Network (ANN) is the basis of the data-driven diagnostic method in this thesis. ANNs are introduced in 1943 by Warren McCulloch and Walter Pits as computational models of reasoning based on a vast collection of interconnected processing elements called neural units (artificial neurons). These are imitators of the observed behavior of the biological brain's cells because the ANNs can learn and implement in many applications. The ANN was modeled on the human brain and nervous systems with internal base units like neurons, axons, dendrites, and synapses. The network acquires knowledge through the learning process, and the internal connection strength between neurons is used to record information. One does not need to know their internal structure as they work as a black box. They learn meaningful relationships between the input and output by processing previously gathered data. The basic structure of a neural network consists of three layers: input, hidden, and output layers. The complexity of these networks increases with increasing hidden layers. Each neuron corresponds to an input parameter at the input layer and vice versa for the output layer and is connected to every neuron of the previous layer through variable synaptic weights.

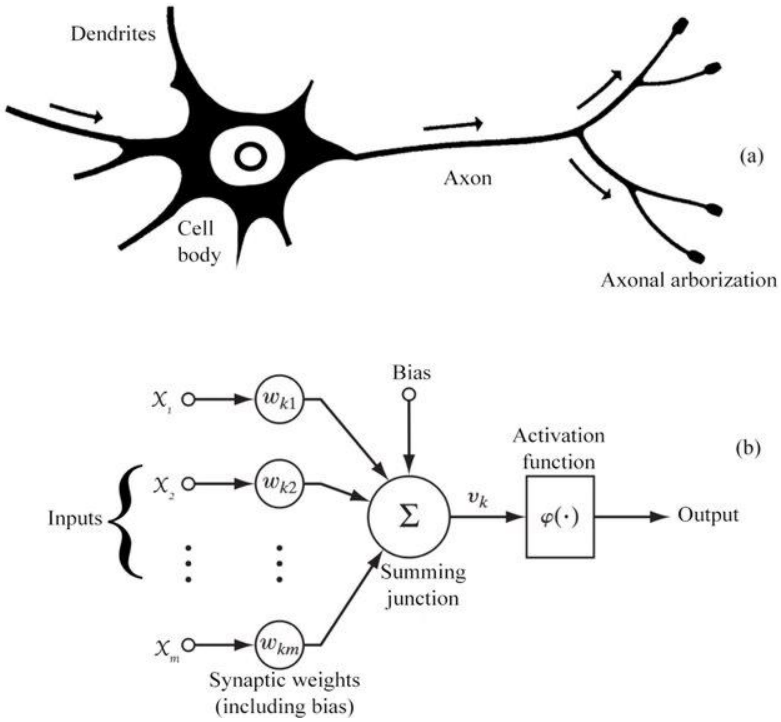


Figure 2.5 The similarity between biological and artificial neural networks [83].

Observing Figure 2.5 gets the analogy between biological and artificial neural networks describes neuron, input, output, and weight similar to soma, dendrite, axon, and synapse, respectively. The mathematical model of neurons is:

$$Output = f(\sum_{i=1}^{n} W_i X_i + b_i) \tag{2.7}$$

Where:

$W_i = \text{weight},$

$X_i = \text{inputs},$

$b_i = \text{bias},$

$f = \text{activation function, e.g., a sigmoid function given by } \sigma(x) = (1+e^{-x})^{-1}.$

Neural Networks (NNs) consist of basic parallel working components that are inspired by the biological nervous systems. The training process trains the ANN by adjusting the connection weights to produce the required outputs. By changing the values of the links (weights) between components, we can train a neural network to conduct a specific function [84]. Commonly, neural networks are adapted or trained to result in a specific target output resulting from one particular input. Based on a comparison of the output and target, the network is adapted until its output matches the target. Therefore, many such input and target pairs are typically used to train a network. There are two training methods for ANN's. Supervised and Unsupervised Learning. The first method is applied when the network knows the input and outputs and is trained to minimize the error between the desired outputs. For the latter, the inputs are only known in a structure, and different learning techniques are used to find a hidden structure in an unlabeled dataset.

Deep Neural Network (DNN) is a Machine Learning (ML) method that uses a multi-layer Neural Network (NN) including more than two hidden layers and it can learn a long chain of causal links [85]. DNN is also known as Deep Learning (DL). Lower layers can, for instance, recognize

edges in image processing, while higher layers can recognize human-sensitive objects such as digits/letters or faces [86]. The main difference between DL against ML is that the domain expert is not needed for the feature extraction from the raw data. The other difference of deep neural networks is their better performance with an increase in data [87].

The NNs and more specifically DNNs are successfully used in the building energy systems, e.g., HVAC systems, for various applications, e.g., optimization, fault diagnosis.

In optimization application, several variables need to be considered including controllable and uncontrollable variables. In this way, the NNs can discover the connection between all variables, e.g., the effect of one parameter on the energy consumption of the system. The NNs can also forecast the air conditioning system's future trend to develop its effectiveness [88]. In another study [89], optimization techniques and NN modeling tools for HVAC systems are researched. The artificial NN is applied to the HVAC scheme to improve energy consumption and indoor air quality. It can also be used to forecast the future trend of the air conditioning scheme.

In fault diagnosis application, Lee et al. used Deep Belief Network (DBN) in combination with Restricted Boltzmann Machine (RBM) for the detection and diagnostic of three abnormal states of the stuck fan, cooling coil valve leakage, and low efficiency of heat exchanger in the Air Handling Unit (AHU) that results shows the diagnosis accuracy above 95% [90]. Overall, the modern management systems of buildings are pretty helpful in capturing and analyzing the system's failures, while manual detection and diagnosis of these faults are pretty time-consuming and effortful tasks [91]. Dehestani et al. [92] proposed an ANN and an online Support Vector Machine (SVM) for FDD and the ANN was used to generate a reference model for the SVM, which they used to detect faults in the HVAC system in both online and offline states. However, in their method, they have a knowledge-driven label generator that combines all labels in a proper logic to generate one label for each fault as a fault needs just one label for training. In our approach, fuzzy theory helps us to automatically classify the system signals and variables.

One focus of FDD lies in the sensors that is known as Sensor Fault Detection and Diagnosis (SFDD). In most diagnostic systems, if a noticeable sensor fault occurs, it raises the failure alarm or the occurrence of a failure. The NN detects failures in the system that are the implication of faults. When a fault is diagnosed, appropriate removal action can be taken to remove the fault from the system.

Figure 2.6 shows an example of a heating and demand-controlled ventilation system that is simulated, and faults and anomalies are inserted into the system to see its performance by Wang et al. [93]. The first step in the system model is programed under normal conditions of the air conditioning system, and the results are obtained. In the second step, random noise is added to the values to imitate the normal measured data. Next, the neural network is trained on the imitated measured data. After training the neural network, a controller is designed which implements FDD strategies. So, if the fault detection unit detects a fault in the system, the neural network is used to regain the value of the faulty sensor. After that, the system is validated using test data on both normal and faulty conditions. The faults that have been discussed, e.g., stuck damper, failure of outdoor airflow sensor, failure of supply airflow sensor, failure of return airflow sensor, soft faults in sensors, and failure of CO_2 sensor, in the reference article [93] are the most common types of faults in an HVAC system. The first technique to diagnose faults is to preprocess the data. For that, filters are applied to the data, which helps in reducing the random noise in the readings. These filters have been applied to the CO_2, outdoor, supply, and return sensors.

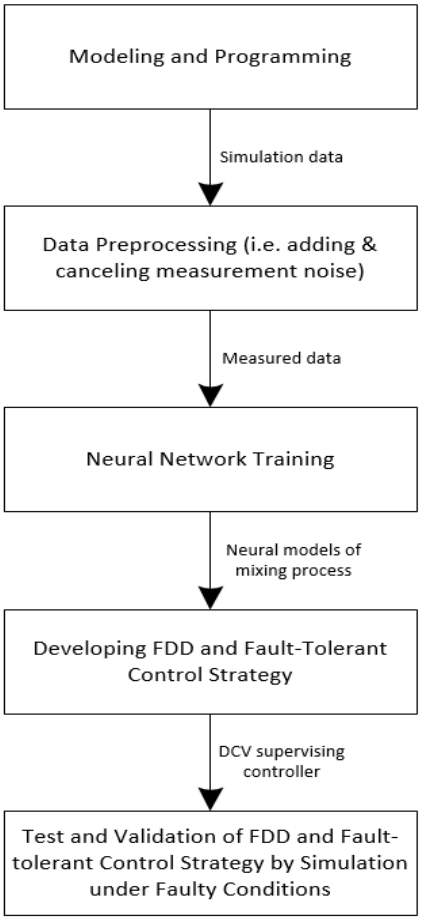


Figure 2.6. State Flow of an FDD System [93]

Each parameter as the neural network's output is a function (e.g., *f*, *g*, or *h*) of other factors, e.g., the outdoor airflow v_O, the damper position u_D, supply air v_S, and the return air v_{Rtn}. For example, if the damper position u_D, supply air v_S, and return air v_{Rtn} are fed to the neural network, the neural network will calculate the outdoor airflow $v_{O,NN}$. The following three equations can visualize the interrelations and one of them is shown in the figure below.

$$v_{O,NN} = f_{NN}(u_D, v_S, v_{Rtn}) \tag{2.8}$$

$$v_{S,NN} = g_{NN}(u_D, v_O, v_{Rtn}) \tag{2.9}$$

$$v_{Rtn,NN} = h_{NN}(u_D, v_O, v_S) \tag{2.10}$$

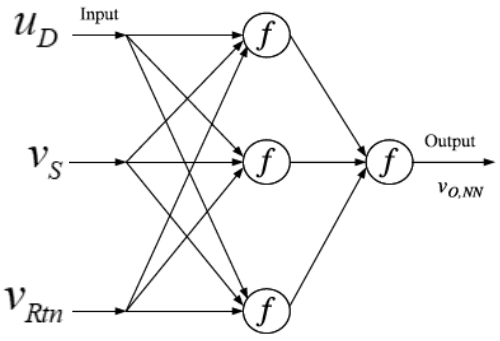


Figure 2.7 An example neural network architecture to calculate the outdoor airflow [93].

There are two types of data used in this neural network, i.e., sensor measured data and the control signals. It is shown that the relationship among sensors needs to be derived for the complete recovery of the system. For example, if the fault of the air supply sensor has been removed, then the outdoor air supply rate values are used in the feedback loop as a corrected reading [93]. The neural network is trained on the non-erroneous data, and its output is also expected to be non-erroneous. However, if a fault occurs in the system, e.g., a stuck damper, the neural network is not obliged to give the correct values for the outdoor airflow rate [93]. The data-driven systems are powerful in training the patterns but if the mean square error rate drops below a certain level during the training, the training procedure stops. Also, these systems need a massive amount of data to be well trained and they are based on the black-box approach. The other shortcoming of the AI method is that they lack the power of expert knowledge. Figure 2.8 indicates a four-layer ANN classifier for FDD [16].

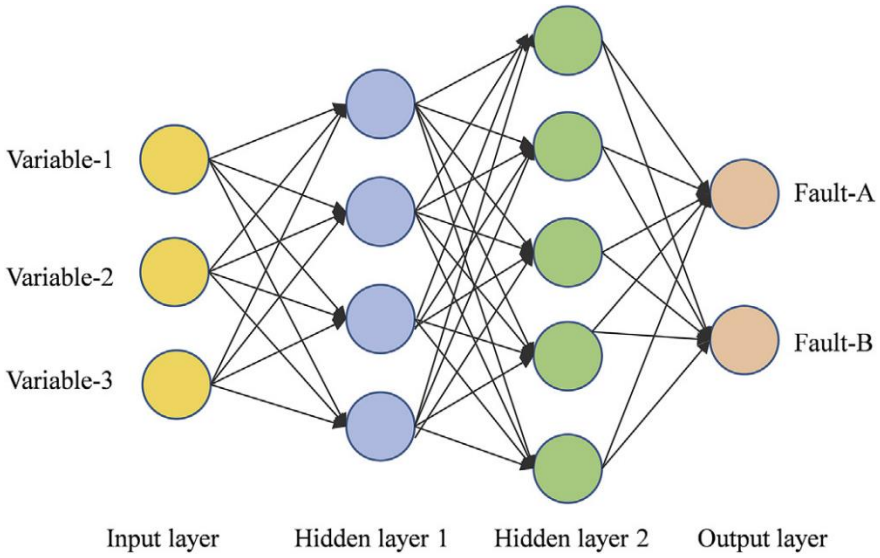


Figure 2.8 An ANN-based fault diagnosis model [16].

Yang et al. [94] explain the phases of the fault diagnosis methods, including information gathering, information analyzing, diagnosing, and resolving. They also discuss the diagnosis methods, including knowledge-based, analytical-model-based, and signal-processing-based methods. Knowledge-based fault diagnosis methods include fault-tree, expert system, fuzzy theory, artificial neural network, grey theory, Bayesian networks, and Petri networks.

The BBN is one of the fault diagnosis methods based on probabilistic theory. Qiu et al. explain that BBNs have effectively modeled probabilistic relationships in complex diagnostic situations by providing a framework to identify critical probabilistic mappings [95]. Probabilistic methods can link symptoms to failures by calculation of failure probabilities [95]. The probability values can be captured from operating data such as all prior probabilities and fault mappings over a sufficient time range or through experts' request of subjective probabilities. CPSs interact with the environment, and the sensory data and signals are measured continually. Therefore, defining an appropriate conditional likelihood density function in BBNs for continuous attributes is critical. BBNs are potent tools for modeling uncertain knowledge and reasoning based on conditions of uncertainty, probabilities, and graph theory [96]. BBNs were introduced by J. Pearl in the 1980s [97], and they were deployed successfully for data mining, financial investment, fault diagnosis, and industrial control. BBNs can be combined with other approaches such as machine learning techniques, signed directed graphs, probabilistic ensemble learning, fuzzy theory, fault-trees, and genetic algorithms. For ensemble learning, Yu et al. [98] have proposed an online fault diagnosis technique for industrial processes with a Bayesian network-based probabilistic ensemble learning strategy (PEL-BN). In this method, they have used an ensemble index to evaluate the diagnosis model in a probabilistic manner. This ensemble acts as a classifier, and the Bayesian network is constructed with three topologies. All these three topologies are integrated to detect the faults. The

experimental results in this work illustrate that the PEL-BN strategy has improved the diagnosis performance of different kinds of single-mode faults and is a feasible solution for the multi-mode faults by integrating decisions derived from different diagnosis BN topologies. Qiu et al. explain that BBNs have effectively modeled probabilistic relationships in diagnostic situations by providing a framework to identify critical probabilistic mappings [95]. Their Probabilistic method can link symptoms to failures by calculation of prior probabilities of faults. They defined the symptoms based on pure expert knowledge, e.g., repair data log and consulting with experts in printers [95]. However, they have not implemented any data-driven method and used a single BBN for the FDD process, but our approach creates a unique BBN for each RDP data set. Their method also needs historical data and a system log for constructing the BBN.

Embedded control systems interact with the environment, and the sensory data and signals are measured continuously. Therefore, defining an appropriate conditional likelihood density function in BBNs for continuous attributes is critical. Tang et al. [96] have presented a Fuzzy Bayesian Network (FBN) for machinery fault diagnosis demanding intensive experience and expert knowledge described by the natural language, such as large, high, or fast. They used fuzzy logic to define the fuzzy events, mapped the system to those uncertain ones, and then produced the BBN. However, they have used a single BBN for the fault detection process, but our approach creates unique BBNs for each output RDP data set. Also, their approach is limited to expert knowledge. In machine learning techniques from the data-driven-based methods, Hu et al. [99] have proposed an intelligent fault diagnosis network for refrigerant charge faults of a variable-refrigerant-flow air-conditioning system. This network is developed under the BBN theory. However, this method as a data-driven-based fault diagnosis is very costly to get sufficient training data for every fault [16]. In signed directed graphs, Peng et al. [100] have proposed a Multi-logic Probabilistic Signed Directed Graph (MPSDG) fault diagnosis approach in chemical processes based on the Bayesian inference. They show that the signed-directed graphs cannot be applied for complicated logic relations, but the authors have shown that the BNs can solve this complexity. They have two offline modeling and online diagnosis phases. In the offline mode, they have analyzed the historical data and deviation values and evaluated the priori probabilities of reason nodes and directed edges. However, constructing the SDG or MPSDG in systems with no historical data is not possible.

On the other hand, the FBBN combines the Bayesian (belief) networks with the fuzzy theory. Chiu et al. have proposed a fuzzy Bayesian classifier with Case-based Reasoning (CBR) for improving car diagnosis problems [101]. In this study, they have used fuzzy theory to define conditional density functions in BBNs to cope with the problem caused by continuous attributes. The accuracy and efficiency of this approach for decision-making applications have been proved by many studies [102]. The FBBN is often used as an effective method of uncertain knowledge representation and reasoning. Fuzzy sets are mathematical sets whose elements have degrees of membership derived from the concept of fuzzy logic, which was introduced by Lotfi A. Zadeh and Dieter Klaua in 1965 [103], [104]. Several examples show the effectiveness of FBBNs in solving uncertain problems, applying Fuzzy sets, and calculating the probabilities of BBNs based on them [105]. Yao et al. [105] have modeled a Fuzzy Dynamic Bayesian Network FDBN for fault diagnosis and reliability prediction in complex systems using various test information. They have presented a model in a fault diagnosis model with uncertain and dynamic information. Their work introduces a dynamic process to the Bayesian network to model a dynamic system, and it includes modeling the BN, the fuzzy set theory applied to BN, and static BN (SBN) that can be converted to dynamic Bayesian network models by introducing time dependency. They have used fuzzy theory to evaluate the reliability of the system with different language variables “very high,” “high,” “intermediate high,” “medium,” “intermediate low,” “low,” and “very low.” However, they generated the BN based on the fault statistics and the fuzzy failure probabilities of root nodes, but in our approach, the starting point is the observations of the system attributes. Intan et al. [106] have applied a fuzzy Bayesian network for analyzing medial tracks. In this paper, they have extended the MI concept

using fuzzy theory to construct an FBBN based on learning Bayesian network structures using an information-theoretic approach introduced by Cheng et al. [107]. They used fuzzy labels to determine the relation between two fuzzy nodes. For example, they have found the relationship of the different disease labels with other factors in a record of data for different patients, e.g., age, degree, and other types of diseases. However, they have used this method for analyzing a medical data set for different patients and diseases; therefore, the application is different. Further, their approach only measured the causal relations in FBBN using the relation direction probabilities between different pairs of parents and children. However, in our approach, we used the causal relations in FBBNs based on relation direction probabilities for fault diagnosis, and our approach includes two modes, an offline training mode, and an online diagnosis phase, to classify the fault cases.

A suitable FDFD system ensures the HVAC systems' proper operation as these systems are subject to various errors, which can lead to malfunctions. HVAC systems fail typically when the actuators "stick" and no longer change their set point, despite the commands. This actuator failure can arise in several parts. For instance, a valve may stick at fully-open, fully-closed, or any intermediate setpoints. If an actuator sticks in an open or closed position for a specific period, some concerns are expected, e.g., the energy waste or uncomfortable environment [108]. The pure knowledge-based diagnosis models are also developed, e.g., a real-time white-box tool for a VAV AHU was developed by Shiazoki and Miyasaka using a signed directed graph. The signed directed graph model is a minimized rules-based model to lower the effort that can detect the symptoms of the faults to find the root cause [109]. However, the performance of the method depended on the thresholds set. The wrong thresholds setting can cause incorrect diagnosis. Also, threshold settings are laborious and time-consuming. Shi et al. introduced a model using probabilistic representations for dependencies of faults and symptoms in a VAV AHU. The fault diagnostic model is developed based on a DBN to diagnose persistent and transient faults while maintaining the FDD system's good performance [110]. However, calculation of the conditional probability values between the faults and symptoms is yet manual and depends on expert knowledge. In addition, for large systems, the amount of data might be unbearable. Therefore, many complicated faults may not be evaluated unless advanced modeling or sensing methods are used.

Zhao et al. [111] have proposed a three-layer Diagnostic BN for chiller faults diagnosis based on the BBNs using a graphical and qualitative illustration of the intrinsic causal relations among three layers of causal factors, faults, and fault symptoms, and this Diagnostic BN can be constructed based on the probability analysis and graph theory. The prior probabilities of root nodes and prior probabilities of faults are the normalized frequencies of faults, and conditional probabilities show the relations of the nodes in three layers. With observed pieces of evidence, posterior probabilities for fault diagnosis can be calculated. This framework uses all beneficial information of the chiller concerned and chiller experts' knowledge, the quantitative and qualitative knowledge from diverse sources is merged and has a strong ability in dealing with incomplete or even conflicting information. However, there are major differences to our proposed method. They calculated the conditional probabilities using statistical or machine learning algorithms, while we have calculated them based on fuzzy weights. Further, our approach is constructed based on MI theory and the dependencies of the system attributes (i.e., signals) for each fault case using BBN theory which only needs expert knowledge to define fuzzy sets. Also, Zhao et al.'s mentioned approach is highly dependent on expert knowledge, especially in calculating prior probabilities. In addition, they have defined rules to conclude the posterior probabilities, but we used the sorting technique showing that the top ranks show reasonable results with high accuracy.

Xiao et al. [112] describe a diagnostic Bayesian network for FDD of VAV terminals. In this method, the parameters of the diagnostic Bayesian network describe the probabilistic dependencies between faults and evidence. The inputs of the diagnostic Bayesian network are the evidence that can be obtained from the measurements in Building Management Systems (BMSs) and manual tests.

The outputs of the diagnostic Bayesian network are the probabilities of target faults. The structure of the diagnostic Bayesian network is a graphical illustration of experts' diagnostic thinking, which can illustrate the relationships among faults and symptoms qualitatively. They have defined a table including the fault nodes, states of the system, rules for defining the states, and prior probabilities for each state. A fault node may have several states that help in estimating the conditional probabilities of the fault evidence given the fault. The rules in the defined table can determine the corresponding states. There are also specific tables and rules to define evidence nodes. The whole structure of this diagnostic Bayesian network depends on expert knowledge and the rules to define the system states. In another paper from these authors, Zhao et al. [113] have also developed a second study on diagnostic Bayesian networks for diagnosing faults in AHU in buildings. This paper developed four diagnostic Bayesian networks to diagnose heating/cooling coils faults, sensors, and faults in a secondary chilled water/heating water supply. However, establishing the FDD strategy and diagnostic Bayesian network nodes is highly dependent on expert rules. The same issue of the dependency to the expert rules is described by Taal et al. [114] in FDD of DCV systems using the 4S3F method based on effects of the diagnosis Bayesian networks probabilities. In this article, a generic set of symptoms and faults has been proposed and symptom rules and their thresholds were estimated by the HVAC design expert.

Cai et al. [115] have used two BNs for multiple-simultaneous faults with a multi-source information fusion-based fault diagnosis methodology. These BNs are established based on sensor data and observed information. The basis of Bayesian network architecture is the cause-and-effect sequence of faults and symptoms. One BN is made based on sensor data, and the other is based on sensor data and observed information; however, the relationship between faults and symptoms is based on expert reasoning and purely knowledge-based.

The data-driven-based methods, e.g., classification-based methods, are numerous but powerful in learning patterns from training data and further need massive training data, and have problems in reliability and robustness [16]. The knowledge-driven-based methods, e.g., Bayesian inference-based or Fuzzy inference-based methods, are powerful in simulating the diagnostic thinking of experts but, they are highly dependent on expert knowledge [16]. Therefore, this thesis uses the capability of the probabilistic methods and fuzzy theory to facilitate automatic calculating the probabilities of continuously measured signals to integrate them into the BBNs to combine both data driven-based and knowledge driven-based diagnostic methods to introduce a generic fault diagnosis classifier as a methodology based on the causal relations in fuzzy Bayesian belief networks using relation direction probabilities. This thesis benefits from the advantages of both knowledge-driven and data-driven domains. The proposed fault diagnosis method in this thesis aims to automatically detect the dependencies of continuous attributes using an example scenario that is DCV and heating system model based on MI theory and applying the fuzzy theory for labeling the system attributes to calculate the conditional probabilities in continuous signal attributes and solve the fault classification problem. This diagnosis is introduced as a classifier and the method has been tested and evaluated to diagnose permanent faults with constant values or fault values in a specific range or a subdomain from an enormous signal domain. The proposed classifier diagnoses faults that occurred in different components and subsystems of the system, including various sensors and actuators, according to the dependencies of signal measurements and attributes of the system.

Table 2.1 indicates an overview of the related works in fault diagnosis and includes the techniques and application domains of different studies and demonstrates whether a technique is categorized as a knowledge-driven or data-driven diagnosis method or covers both. Further, this table highlights the novelty of this thesis from different aspects, i.e., the target research is independent of the historical data, independent of expert knowledge, or computing-resource efficient. This overview indicates some of the studies cover both diagnostic techniques such as this thesis, but they are neither independent of the historical data, independent of the expert knowledge, nor computing-resource efficient.

Table 2.1. Overview of the related works in fault diagnosis.

Author(s)	Technique and Application Domain	Knowledge-driven diagnosis	Data-driven diagnosis	Independent of historical data	Independent of expert knowledge	Computing Resource efficient
Taal et al. **2020** [114]	**BBN based** FDD in DCV	✓	✓	**NO**	**NO**	**NO**
Dehestani et al. **2011** [92]	FDD in HVAC	✓	✓	YES	**NO**	**NO**
Shi et al. **2018** [110]	**BBN based** FDD in VAV AHU	✓	✓	**NO**	**NO**	**NO**
Tianyun Gao *(Dissertation)* **2020** [116]	**BBN based** Building FDD using data modeling	✓	------	NO	NO	YES
Md. Tanjin Amin *(Dissertation)* **2018** [117]	**Dynamic BN based** FDD in HVAC	✓	------	NO	NO	YES
Wang et al. **2002** [93]	FDD in DCV	------	✓	NO	YES	NO
Luo et al. **2019** [118]	FDD in chilled water system	------	✓	NO	YES	NO
Qiu et al. **2001** [95]	**FBBN based** FDD in printers	✓	------	NO	NO	YES
Tang et al. **2007** [96]	**FBBN based** FDD in machinery	✓	------	NO	NO	YES
Hu et al. **2018** [99]	**BBN based** FDD in VRF	✓	✓	**NO**	**NO**	**NO**
Peng et al. **2014** [100]	FDD in production process	✓	------	NO	NO	YES
Chiu et al. **2007** [101]	**FBBN based** FDD in cars	✓	✓	**NO**	**NO**	**NO**
Yao et al. **2015** [105]	**FBBN based** FDD in aircrafts	✓	------	NO	NO	YES
Zhao et al. **2013** [111]	**BBN based** FDD in chillers	✓	------	NO	NO	YES
Xiao et al. **2013** [112]	**BBN based** FDD in VAV	✓	------	NO	NO	YES
Cai et al. **2014** [115]	**BBN based** FDD in heat pumps	✓	------	NO	NO	YES
Zhao et al. **2017** [113]	**BBN based** FDD in AHU	✓	------	NO	NO	YES
Intan et al. **2010** [106]	**FBBN based analysis of medical data records**	✓	------	NO	NO	YES
This Dissertation (Behravan) 2021	**Generic FBBN based** FDD (example scenario DCV)	✓	✓	**YES**	**YES**	**YES**

In specific, the novelties of the introduced fault diagnosis method in this thesis are based on the following points:

- ✓ **Integration of data-driven classifier, fuzzy logic, and Bayesian belief network for the combination of data-driven and knowledge-driven diagnosis:** The composed diagnostic classifier in this thesis includes the knowledge-driven diagnosis theories, i.e., fuzzy and Bayesian theories, and data-driven diagnosis strategy based on the intelligent diagnostic classification algorithm that can automatically diagnose the faults. In offline mode, for each fault class, an RDP table is computed and stored in a fault library. In online mode, we

determine the similarities between the actual RDP and the offline precomputed RDPs. The combination of BBN and fuzzy logic in our introduced method analyzes the dependencies of the signals using MI theory. The method creates a unique RDP table for each class of faults and data sets. This method can also be extended to additional faults by adding RDPs of new fault classes to the offline library. This method provides more understandability, less effort for experts, and higher diagnostic accuracy. Our strategy is less dependent on expert knowledge and only requires the expert to define fuzzy sets and the whole process can intelligently and automatically classify the faults compared to the other knowledge-based strategies. The evaluation results show that our strategy can accurately map a fault case to the predefined fault in the library.

- ✓ **Reveal of hidden and intrinsic dependencies of trends or statuses in signals over time in case of faults:** In our diagnostic method, a novel strategy is introduced based on the dependency of trends (for sensors) or statuses (for actuators) in different subdomains over time. Therefore, our automatic diagnostic method can find the intrinsic and hidden dependencies of measurement signals and statuses that change concurrently over time in the case of a specific fault based on MI and fuzzy theory. For example, if a damper sticks at open status, the room temperature decreases and makes the heater stick at ON status indirectly because the heater wants to compensate for the heating load due to the damper, which is a hidden dependency between damper and heater status signal.
- ✓ **Extendibility of the strategy in this thesis to complex systems:** Finding fault-symptoms dependencies and fault diagnosis in other knowledge-based strategies in the literature are purely based on the expert knowledge, which can be very hard or impossible if the target system is complex with many measurement signals and statuses to the limit that even the experts cannot find the exact and hidden dependencies. However, our approach can automatically find these dependencies and faults in complex systems.
- ✓ **Mapping and evaluation of the novel diagnostic method for DCV and heating systems:** The presented diagnostic fault model covers sensor and actuator faults to map and evaluate the integrated diagnostic method to DCV and heating systems, as an example use case.
- ✓ **Experimental evaluation of the introduced diagnostic method based on FBBNs compared to deep neural network method using simulation framework:** Manufacturers typically are reluctant to provide the full-set fault data. Therefore, the diagnostic method in this thesis is implemented in a simulation framework that can inject any desired faults into the system [27]. The evaluation results show a convincing performance of the introduced composed method (knowledge-driven and data-driven) in fault diagnosis in this thesis compared to a deep neural network (data-driven method) [85]. The review paper in state-of-the-art [16] shows the lack of accuracy of the knowledge-driven methods.
- ✓ **Accurate fault diagnosis independent of prior knowledge and historical data:** The other strategies use the BBN theory, but they use historical data, repair logs, or experimental data to calculate the prior conditional probabilities. In the strategy introduced in this thesis, the signals only need to be defined as continuous or discrete variables and use the fuzzy theory to categorize the signal values to create the Bayesian network.

According to the state-of-the-art mentioned above, the limited accuracy and performance of the expensive qualitative diagnosis models with their manual setting of the thresholds is still a problem. Also, another problem still exists, which is the construction of the conditional probabilities between the faults and symptoms that are still manual and rely on expert knowledge, e.g., the construction of the conditional probabilities between the faults and symptoms in the fault diagnosis methods based on Bayesian networks. Furthermore, the complexity problem, energy waste,

discomfort conditions, and the challenging nature of fault diagnosis tasks in HVAC systems are examples of the CPSs area. These problems will be more prominent if the system gets more complex.

3. SYSTEM MODEL

A complete understanding of the system's overall behavior is required to optimize a system or technique through system model analysis. Analysis of the system model leads to knowledge about the system's functionality. The function of a system is what the system is intended to do [18]. The model specifies what a system does [75]. This chapter describes the modeling techniques and the theoretical principles of the system models for HVAC systems.

Model: A model is a physical, mathematical, or logical description of a system entity, phenomenon, or process [119]. The model should be a simple but sufficiently detailed representation of a system to study and show valid results in analogy to the existing system.

Analytical model: An analytical model is a mathematical description of a solution for a problem used for simulating, explaining, and making predictions of the mechanisms inside complex physical processes and the controlling computer systems [120]. A system's analytical solution can help construct the simulation platform when it describes the system in a simple but sufficiently detailed form.

Simulation: The simulation executes a model over time. A simulation brings a model to action and shows how a particular object or phenomenon will behave. The simulation is the art of imitating a real-world system over time to estimate and visualize the measures of performance of the system with the simulation-generated data. The simulation can tell us what may happen with any change in designs or scenarios before implementation, giving insight into the model capabilities without implementation cost and comparatively in a short time.

The goal of the simulation is to provide an analysis tool that can run the model and investigate the effects of changing a design parameter on the existing system to predict the performance and infer the new system's behavior. The design parameters are properties of the models that can be changed before and during the simulation. Valuable insight can be obtained by changing simulation inputs and observing how variables interact during the simulation, what are their changes and effects on the system performance, and the resulting outputs, without disrupting ongoing operations. In this way, the answer to “what-if” questions can be straightforward. The simulation tools give the freedom and possibility to the user to design the system and set the parameters based on the desire to make the simulation comparable to the real scenarios or even create scenarios facing limitations, risky situations, or would involve expensive practical experiments.

The models can be classified into three types: white-box models such as quantitative analytical models based on first principles and physical process [29], black-box models such as statistical modeling, and grey-box models, which are a hybrid of the two foremost. The grey-box models use deterministic differential equations [121] and continuous-time modeling, calibrating the model parameters over time [122]. The differential equations can be solved by implementing the dynamic models in modeling languages such as MATLAB/Simulink or Dymola.

The lumped capacitance model is considered a highly efficient grey-box model, which means it is low-cost and straightforward from the computational expense aspect [52]. This method is used frequently to assemble simple thermal models [123].

Foucquier et al. illustrate a complete comparison between white, black, and grey-box techniques proposed in Table 3.1 [124].

Table 3.1 Comparison between white, black, and grey box techniques [124].

Methods	Building Geometry	Training Data	Physical Interpretation
White Box (Physical)	A detailed description of the building geometry is required	No training data are required	Results can be interpreted in physical terms
Black Box (Statistical)	A detailed description of the building geometry is not required	A large amount of training data collected over an exhaustive period is required	There are several difficulties to interpret results
Grey Box (Hybrid)	A rough description of the building geometry is enough	A small amount of training data collected over a short period is required	Results can be interpreted in physical terms

The models in the white box are prepared based on well-known physical relations and thermal properties. Therefore the implementation needs complete prior knowledge regarding all the processes and the parameters without observing the study's objective. This kind of detailed description of the building geometry can be assumed a drawback as all the details may not always be available, or sometimes the data collection processes using various sensors are expensive and time-consuming. White-box models are based on static or dynamic models, linear or nonlinear models, and continuous or non-continuous differentiable models. In contrast to the physical model (white-box model), the black-box model mostly uses data-driven techniques such as statistical methods or machine learning to show the system behavior. This model is only based on input and outputs with observed measurements. The black-box approach is applied when the phenomena of the objects are too complex to be described by fundamental principles. The black-box model estimates the relationships between input and output variables, independent of the system phenomena or its variables. The output from the black-box method is not always easy for interpretation of the system behavior; thus, the hybrid of these methods is here surviving method. "Grey-box models are analytical models loosely based on first principles, and the model parameters can still be traced to the process's physical response" [23], [125]. Grey-box models are faster in processing and easier to formulate and establish than the white-box models and more robust than the black box models [23]. A number of grey-box models are found in the literature [126], [127], [128].

3.1. PHYSICAL MODEL

As described, the physical model framework is established in this thesis. In the Simulink environment, there is a helpful library for thermal modeling called Simscape. The Simscape schematic components of Simulink demonstrate physical phenomena or elements. The signal lines between these components are considered physical connections of the existing system, which transmit power. Each Simscape domain uses a distinct color and line style for the connection lines and block icons. The established models by Simulink blocks show a physical network approach to enable the system designer to analyze the behavior of the system by its physical structure and mathematical equations will help to get a better understanding [129]. Figure 3.1 describes the steps of a simulation flow. The physical model that is the analytical form of the physical system based on differential equations is a part of the simulation step flow. This simulation flow (in blue-colored blocks) starts with input data such as weather data, building elements specifications, dimensions, and materials and collects set parameters, internal and external gains using GUI (in green-colored blocks). Then, the data from the first stage will be applied to the physical model, which can be simulated in a simulation tool to result in simulation output and time-series scopes.

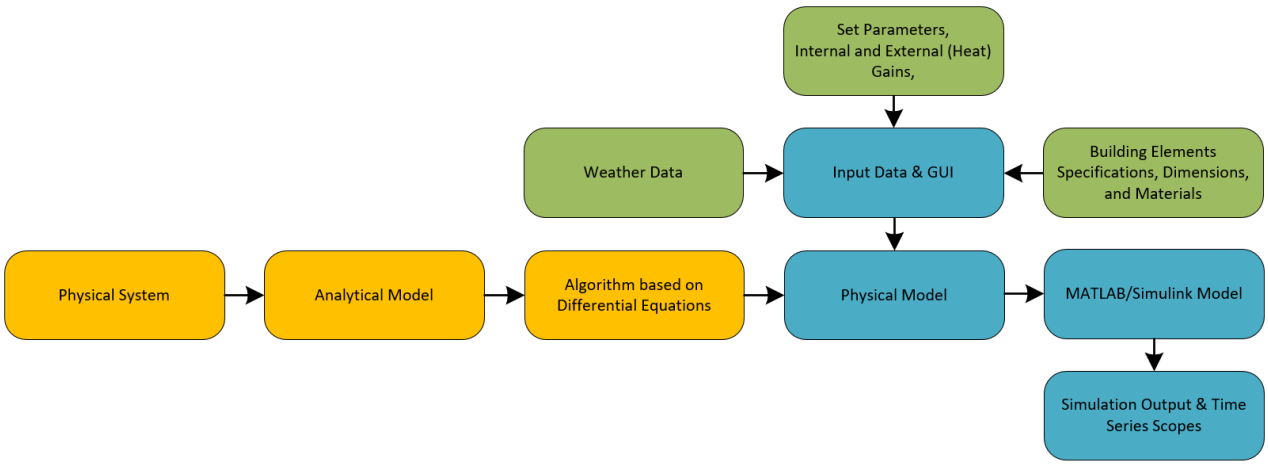


Figure 3.1 Simulation flow.

3.1.1 Principles of Thermal Models and Space Heating in Multi-zone Buildings

As part of the HVAC system, the central space heating system supplies the heat (H for Heating) to the entire building, including its internal spaces, in a space-to-space manner. The term central describes the heat generated at a point of the building, and the thermal energy is delivered to all the zones to compensate the heat loss and to maintain internal air conditions at the thermal comfort level using a medium which is usually water or steam at a specific temperature and pressure considering its design condition. The other solution to warm up the building zones is using electrical heaters powered by electricity to produce heat from the electrical resistors.

The models are identified by two thermal behaviors of static and dynamic. In static models, the output of the system does not depend on time. The static behavior simplifies the thermal model and dominates the restrictions of computing resources. In dynamic models, the output is time-varying due to evolution over time. These dynamic models are typically represented by ordinary, partial, linear, or non-linear differential equations. The dynamic behavior helps the comprehension of the thermal exchange fundamentals in simulation. The static behavior is helpful in comprehension of steady-state conditions of buildings when the internal and external inputs and outputs are available to be controlled [130]. The thermal phenomena, including heat transmission, heat storage, fluid flow, heat flux, and thermal properties of building elements, are highly time-sensitive [131].

The design conditions of thermal models and space heating are based on several factors, including physical and thermodynamic specification of the building, e.g., U-values and thermal capacity values of the building materials, e.g., walls, ceilings, floors, and the air inside enclosed spaces, internal heat transfer including conduction, convection, and radiation, ventilation rate, environment weather condition, e.g., temperature, radiation measure, wind speed, pressure difference, specification of the heating system, e.g., system type (hot water, steam, or electricity) and efficiency, control technique, user requirements, e.g., desired temperature, indoor air quality, and internal condition, e.g., heat gain due to home appliances and occupant's behavior.

"*Heat transfer (or heat)* is thermal energy in transit due to a spatial temperature difference" [132]. The heat transfer will happen once a temperature difference exists in a medium or between media. The heat can be stored in the thermal heat capacities and transmitted through these elements via heat transfer methods. Mainly, three types of heat transfer account for almost any type of heat transfer, *conduction*, *convection*, and *radiation*.

3.1.1.1 CONDUCTION

The heat transfer because of a temperature gradient in a static medium (e.g., a solid) is called ***conduction***. The higher temperature is interpreted as higher molecular energy. The molecules with a high level of energy transfer their energy to the less energetic molecules via collision. Therefore, the energy is transferred in the direction of decreasing temperature. In other words, conduction is the diffusion of energy due to random molecular motion.

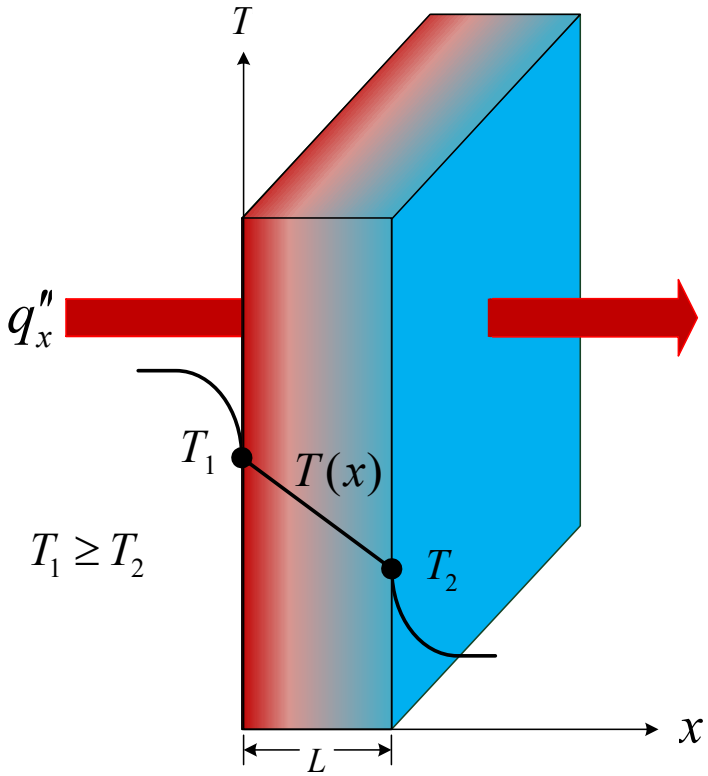


Figure 3.2 One-dimensional heat transfer by conduction through the wall.

Figure 3.2 shows the conduction heat transfer through a wall, its direction from the high-temperature side to the low-temperature side, and its temperature profiles. The heat transfer is the calculation of the rate equation. The rate equation calculates the amount of energy transferred per unit of time, known as *Fourier's law*. For the one-dimensional heat transfer

$$q'' = -k\frac{dT}{dx} \tag{3.1}$$

Where,

q'' = The heat flux per unit area perpendicular to the direction of transfer in [W/m^2],

$\frac{dT}{dx}$ = The temperature gradient in the direction of transfer in [K/m],

k = The *thermal conductivity* constant in [$W/m \cdot K$].

Thermal conductivity is a transport property, and its value depends on each material's atomic and molecular structure. The minus sign in equation 3.1 shows that the heat transfer occurs in the direction of decreasing temperature. The heat flux can be rewritten as below, assuming a linear temperature distribution inside a medium.

$$q''_x = k\frac{\Delta T}{L} \tag{3.2}$$

Where,

ΔT = The temperature difference in [K],

L = The path length of thermal conduction in [m].

The heat rate through an area can be calculated:

$$q = q'' \cdot A = k\, A \frac{\Delta T}{L} \tag{3.3}$$

Where,

q = The heat rate in [W],

A = The area perpendicular to the heat transfer direction in [m^2].

3.1.1.2 CONVECTION

The temperature gradient between moving fluid and a bounding surface causes a particular form of heat transfer called ***convection*** [132]. The energy transfer in convection is mainly due to random molecular motion (diffusion) and macroscopic (bulk) motion mechanisms. The bulk motion represents the fluid motion associated with large numbers of molecules moving altogether due to the temperature gradient. Figure 3.3 illustrates the boundary layer development in one-dimensional heat transfer by convection beside the wall edges and temperature profiles.

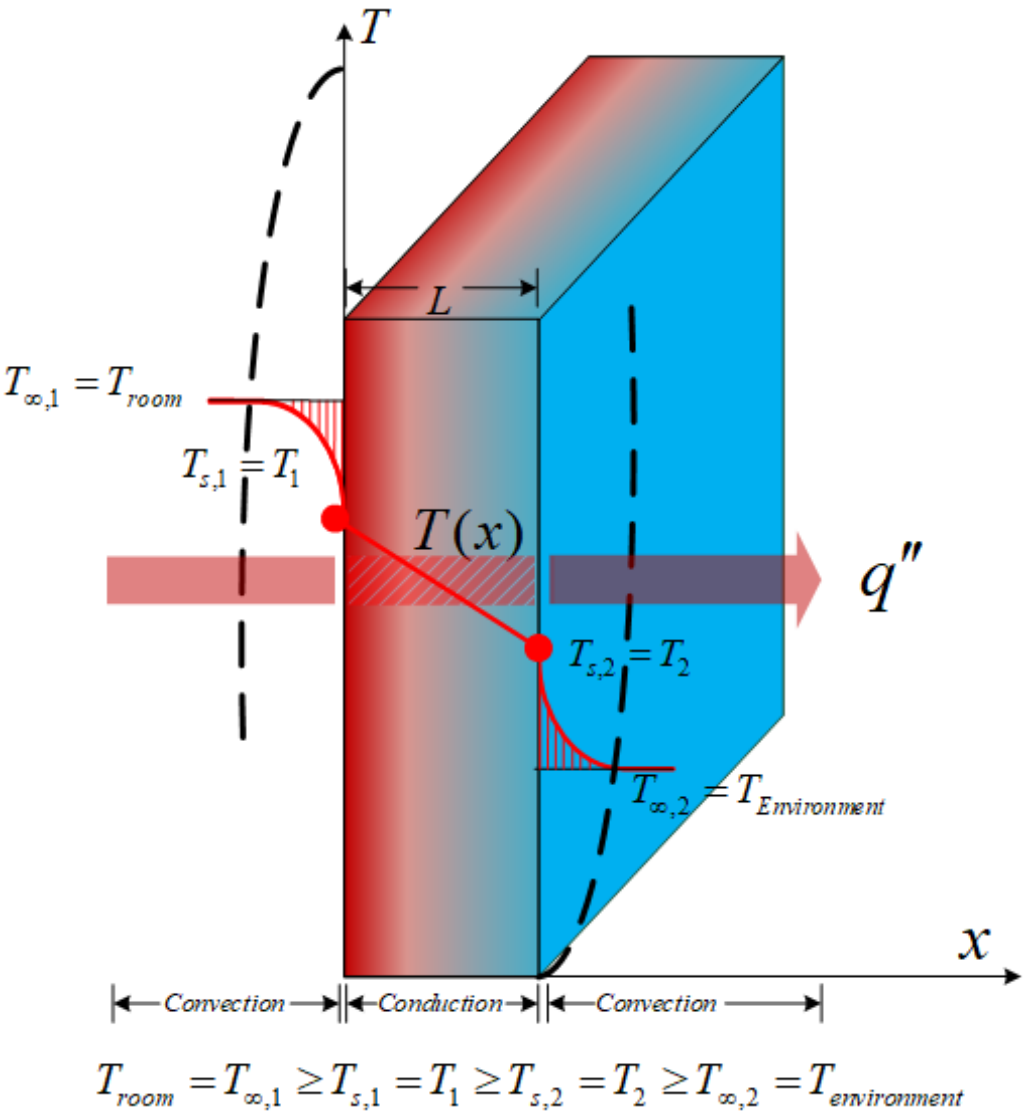


Figure 3.3 Boundary layer development in one-dimensional heat transfer.

Interaction of the fluid flow with the surface develops a region called hydrodynamic or velocity boundary layer in the fluid. The flow velocity in this layer changes from the value of zero at the surface to a finite value u_∞. Likewise, if there is a temperature difference between the surface and the flow, the interaction of the flow with the surface develops a region called the thermal boundary layer in the flow. The temperature in this layer varies from T_s at $x = 0$ to T_∞ of the flow, which in this thesis is the inside air temperature of a zone.

Newton's law of cooling calculates the convective heat flux q'' , which is proportional to the temperature difference between fluid and surface:

$$q'' = h(T_s - T_\infty) \tag{3.4}$$

Where,

h = The convection heat transfer coefficient in [$W/m^2 \cdot K$],

T_s = The temperature of the body at the surface ($y = 0$),

T_∞ = The temperature of the environment flow.

The convection heat transfer depends on conditions in the boundary layer, surface geometry, the nature of the fluid motion, and an assortment of fluid thermodynamic and transport properties. The total heat transfer through an area is:

$$q = q'' \cdot A = hA(T_s - T_\infty) \tag{3.5}$$

3.1.1.3 RADIATION

All surfaces of finite temperature emit energy in the form of electromagnetic waves. Hence, there is also net heat transfer by radiation between two surfaces at different temperatures in the absence of an intervening medium. The energy emitted in this way is named thermal radiation. The *blackbody* emissive power E is the rate of energy released per unit area as described by the *Stefan-Boltzmann law*:

$$E_b = \sigma T_s^4 \tag{3.6}$$

Where,

σ = The Stefan-Boltzmann constant ($\sigma = 5.67 \times 10^{-8} \quad W/m^2 \cdot K$),

T_s = The absolute temperature in [K]

The heat flux emitted in the real surfaces is less than the relative blackbody's surface:

$$E_b = \varepsilon \sigma T_s^4 \tag{3.7}$$

Where,

ε = The emissivity, $0 \le \varepsilon \le 1$.

The *emissivity* value describes the efficiency of a real surface in emitting energy relative to the blackbody and is highly dependent on the surface material and finishing.

The *irradiation*, G (W/m2), defines "the rate at which radiation is incident upon the surface per unit surface area, over all wavelengths and from all directions" [132].

$$G_{abs} = \alpha G \tag{3.8}$$

Where,

G_{abs} =The absorbed irradiation,

α =The absorptivity, $0 \le \alpha \le 1$.

The *absorptivity* is the rate of radiation energy absorbed per unit area of the surface depending on the nature of the irradiation and the surface. If the surface is opaque, some portions of the irradiation will be reflected, and if it is semitransparent, some portions can be transmitted through the surface and figure below shows a visualization of the radiation heat transfer.

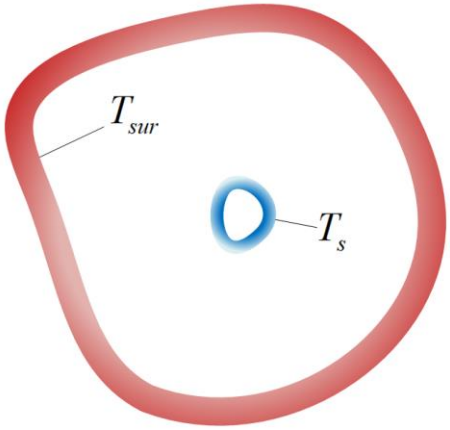


Figure 3.4 Radiation heat transfer from the surrounding to the surface with lower temperature.

For the particular case of the small convex object in a large cavity, it is assumed that such a small surface is surrounded by an isothermal surface, e.g., the sun. The net radiation heat transfer rates the difference between thermal energy released due to radiation emission and thermal energy gained from radiation absorption described by the equation below [132].

$$q''_{rad} = \frac{q}{A} = \varepsilon\sigma(T_s^4 - T_{sur}^4) \tag{3.9}$$

In this study, the radiative heat gain is relative to the outside temperature $T_{out}(t)$ for simplification, which is a sinusoidal function [1]:

$$q''_{rad} = \tau(T_{out}(t) - \zeta) \tag{3.10}$$

Where,

τ =100 and

ζ =7, are the constant values.

3.1.2 Conservation of Energy: Relationship to the 1st Law of Thermodynamics

The 1st law of thermodynamics describes that "the change of stored energy in a bounded control volume must equal the amount of energy that enters the control volume minus the amount of energy that leaves the control volume" [132]. In other words, the total energy of a system is conserved until energy crosses its boundaries. This energy can be in the form of heat transfer over the boundaries or work applied on the system or by the system:

$$\Delta E_{st}^{tot} = Q - W \tag{3.11}$$

Where,

ΔE_{st}^{tot} = The total energy change of the system (subscript *st* denotes the stored energy),

Q = The *net* heat transfer to the system,

W = The *net* work done by the system.

The stored energy over time, which can be mechanical and thermal, is an equilibrium among the thermal and mechanical energy transport across the control surface:

$$E_{st} = E_{in} - E_{out} + E_g \tag{3.12}$$

Where,

E_{st} = The stored energy inside the control volume,

E_{in} = The inflow term of energy transport,

E_{out} = The outflow term of energy transport,

E_g = The energy generation inside the control volume.

The heat stored in an enclosed volume causes a temperature change of the medium. The equation below explains that the heat required for a temperature change is related to the mass (m) of the medium enclosed in the control volume and the specific heat capacity c_p, which is a physical quantity of that medium.

$$q = mc_p \frac{dT}{dt} \tag{3.13}$$

Where,

q = The heat load in watts [W],

c_p = The specific heat capacity in [$kJ/kg\,K$],

ρ = The density of outdoor air in [kg/m^3],

$\frac{dT}{dt}$ =The rate of temperature change corresponds to the heat flow in [$K/$s].

3.1.3 THE THERMAL RESISTANCE CONCEPT

Circuit expressions provide a suitable tool for simple visualization of the heat transfer concept. There is an analogy between heat diffusion and electrical charge diffusion if there is no heat generation inside the wall. There can be a thermal resistance against heat conduction as there is an equivalent electrical resistance against electricity conduction. Similar to Ohm's law which describes an electrical resistance as the ratio of a driving potential to the corresponding transfer rate, there is an equation that gives the thermal resistance for conduction in a plane wall:

$$R_e = \frac{E_{s,1} - E_{s,2}}{I} = \frac{L}{\sigma A} \tag{3.14}$$

$$R_{t,cond} = \frac{T_{s,1} - T_{s,2}}{q_x} = \frac{L}{kA} \tag{3.15}$$

Where,

R_e =The electrical resistance in [Ω]

$E_{s,1}, E_{s,2}$ =The electrical voltages at the 1st and 2nd surface in [V]

I =The electrical current in [A]

σ = Stefan-Boltzmann Constant in [W/m^2K^4]

$R_{t,cond}$ = The thermal resistance in [$K/W \cdot m^2$]

$T_{s,1}, T_{s,2}$ =The thermal temperatures at the 1st and 2nd surface in [K].

Same as the thermal-electrical analogy in conduction, there can be thermal resistance against heat convection. Newton's law of cooling is an empirical relationship discovered by Sir Isaac Newton (1642-127), describing that "the rate of heat loss of a body is proportional to the temperature difference between the body and its surroundings." Therefore, keeping the air inside the home warm on a cold day involves a higher cost because more temperature difference with the outside needs much heat. Newton's cooling law can be mathematically exposed as the following equation:

$$q = hA(T_s - T_\infty) = hA\,\Delta T\left(t\right) \tag{3.16}$$

Where,

h =The convective heat transfer coefficient in [W/m^2K]

Then, the convective thermal resistance is calculated by

$$R_{t,conv} = \frac{T_s - T_\infty}{q} = \frac{1}{hA} \tag{3.17}$$

Therefore, an equivalent thermal circuit can be proposed for every heat transfer problem.

Assuming that steady-state conditions characterize the system if the temperature at each point is independent of time, based on the thermal resistance concept, the heat transfer rate might be shown in the following form

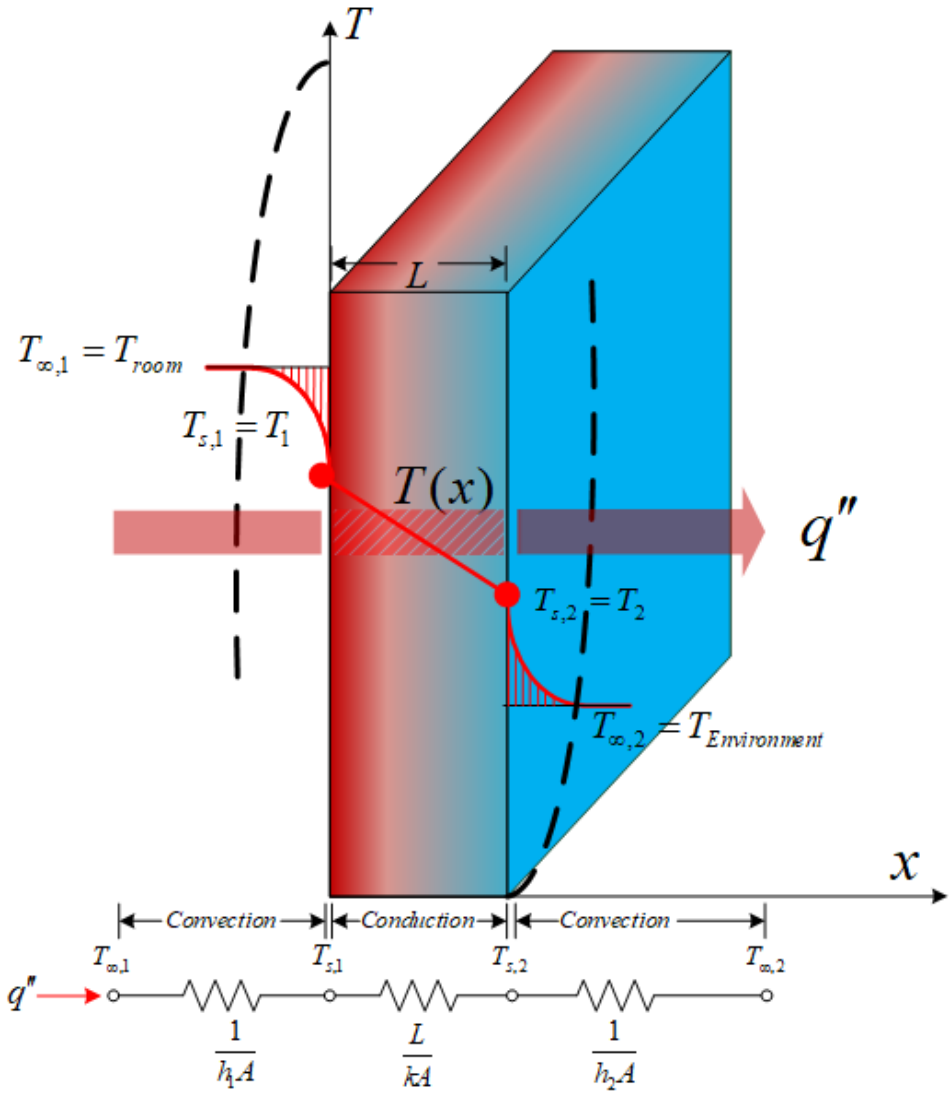
$$q = q'' \cdot A = \frac{\Delta T}{R_t} \tag{3.18}$$

Where,

R_t = *Thermal Resistance* in [$K/W \cdot m^2$].

The thermal resistance is variously parametrized for each of the three different types of heat transfer (conduction, convection, and radiation).

When a wall separates two fluids of different temperatures, there is a convective heat transfer from the hot fluid at $T_{\infty,1}$ to the first surface of the wall at $T_{s,1}$. There is a conduction heat transfer inside the wall from the first surface (with higher temperature) to the second surface of the wall at $T_{s,2}$, and there is a convective heat transfer from the second surface of the wall at $T_{s,2}$ to the cold fluid at $T_{\infty,2}$. The temperature distribution is measured by solving the heat transfer equation assuming the proper boundary conditions.



$$T_{room} = T_{\infty,1} \geq T_{s,1} = T_1 \geq T_{s,2} = T_2 \geq T_{\infty,2} = T_{environment}$$

Figure 3.5 Equivalent thermal circuit.

Also, in analogy to the electrical current, the heat transfer rate is the same for the series elements that are tandem. As an example, for Figure 3.5 the heat transfer rate is as below:

$$q_x = \frac{T_{\infty,1} - T_{s,1}}{1/h_1 A} = \frac{T_{s,1} - T_{s,2}}{L/kA} = \frac{T_{s,2} - T_{\infty,2}}{1/h_2 A} \tag{3.19}$$

this equation is valid also for the overall temperature difference of $T_{\infty,1} - T_{\infty,2}$

$$q_x = \frac{T_{\infty,1} - T_{\infty,2}}{R_{tot}} \tag{3.20}$$

The total thermal resistance is calculated as

$$R_{tot} = \frac{1}{h_1 A} + \frac{L}{kA} + \frac{1}{h_2 A} \tag{3.21}$$

For a series composite wall with more layers, the calculations are the same as with a series electrical circuit, and it is easier to work with an *overall heat transfer coefficient U*:

$$q_x = UA\,\Delta T \tag{3.22}$$

where ΔT is the overall temperature difference. If there is heat transfer through a composite material, the equivalent resistance is the summation of the resistances of each part of the composite. Likewise, if different heat transfer modes exist, then the total resistance is the summation of all the resistances respective to different modes, as follows:

$$U = \frac{1}{R_{tot} A} = \frac{1}{\left[\left(1/h_0\right) + \left(L_1/k_1\right) + \ldots + \left(L_{n-1}/k_{n-1}\right) + \left(1/h_n\right)\right]} \tag{3.23}$$

or

$$R_{tot} = \Sigma R_n = \frac{\Delta T}{q} = \frac{1}{UA} \tag{3.24}$$

3.1.4 THE LUMPED CAPACITANCE METHOD

Hudson and Underwood (1999) described the mathematical model for building the system's operation simulation using an RC electric circuit model.

The lumped capacitance method is a well-known method based on the analogy between thermal systems and electric circuits [133], [134]. It is easier, in this case, to describe these phenomena by the grey-box modeling approach [135]. This approach is based on the semi-physical laws analog-circuit model such as the equivalent electric RC circuit. The RC model can be presented as mathematical expressions (algebraic, differential equations), distinguishing between deterministic and stochastic parameters. These parameters cannot be directly observed (indirect measurement). They need to be estimated [136], [137] depending on the prior and posterior information of the building system.

The lumped capacitance model or lumped parameter method, or the lumped system analysis, is the simplest method of building thermal response models. This method simplifies the description of a physical system with a given topology, including the building elements' abstraction to a discrete number of spatially uniform-temperature elements ("lumps") used in network analysis. Nevertheless, spatially uniform-temperature value changes over time. Then, the energy balance equation of transient conduction can be proposed based on these construction elements.

The RC approach is derived from the lumped capacitance solution of the transient heat transfer equations, well known as the thermal-electric analogy described in Table 3.2. This approach consists of spatially uniform temperature elements that construct an energy balance equation. Therefore, this method cannot show the temperature distribution inside elements, but it perfectly describes the overall thermal behavior of building zones. The model consists of several lumped thermal resistances ($R_1, R_2, ..., R_n$) and some thermal capacitances ($C_1, C_2, ..., C_n$), idealized into electrical components such as resistors and capacitors in an electrical network with conducting wires. The thermal resistance is analogous to the electrical resistor and the heat transferred is analogous to the electrical current. In this model, the capacitance reservoir absorbs heat until its thermal steady state. The thermal resistances also play different roles in the different modes of heat transfer, e.g., conduction, convection, and radiation. The order of this model depends on the number of thermal capacitance layers (thermal mass). A 1st order lumped parameter in analogy to the construction elements can be seen in figure 3.6 (a typical wall view of this thesis).

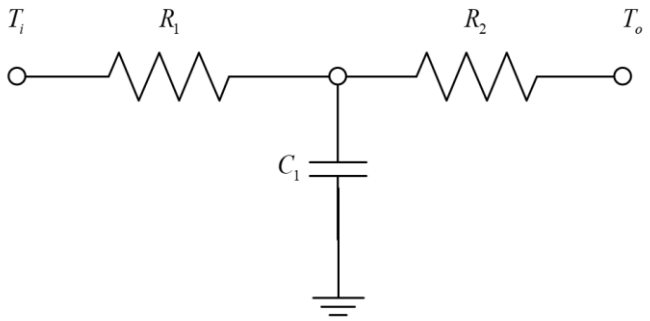


Figure 3.6 The 1st order lumped elements in the RC approach.

The 2nd order or more lumped parameter construction element can be used if the walls between the zones include more layers, as shown in Figure 3.7.

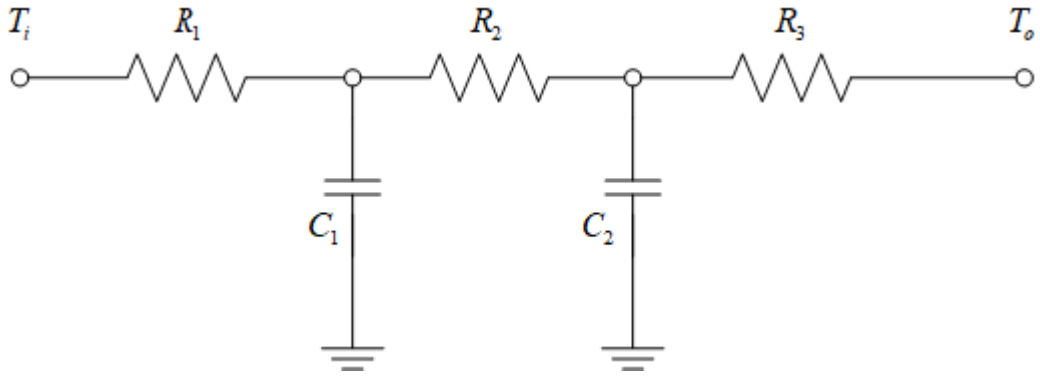


Figure 3.7 The 2nd order lumped elements in the RC approach.

Table 3.2 Thermal-Electric Analogy.

Quantity	**Across Variable**	**Through Variable**	**Dissipation Element**	**Storage Element**
Electrical Circuit Parameter [Unit]	Voltage – V [V]	Current – I [A]	Electrical Resistance – R [Ω]	Electrical Capacitance – C [F]
Thermal Equivalent Parameter [Unit]	Temperature – T [K]	Heat Flow Rate - q [W]	Thermal Resistance – R [K/W]	Thermal Capacitance – C [J/K]

The usability of the lumped capacitance method is high whenever the heat conduction inside the system boundary is much larger and faster than the heat transfer across the system boundary into or out of it. With this condition, the object temperature at a specific time is meaningful as the entire element has the same temperature. The Biot number (Bi) is a dimensionless parameter as the ratio of the conductive heat resistance within the element body to the convective heat transfer resistance across the system boundary to check this condition. If the Biot number is less than 0.1 for solids, the entire element has the same temperature. If the Biot number is greater than 0.1, the system's length can be divided into several sections or lumps with a Biot number of less than 0.1 for that section to satisfy the condition.

As discussed in this section earlier, the lumped-capacitance method has been used in this study to model the thermal dynamics of the use-case office building (thermal network model) where the heat transfer is illustrated by thermal resistance and heat storage (thermal capacitance) implemented in chapter four. A thermal node represents each zone and each wall. The nodes are connected via thermal capacitors to the ground reference and thermal resistors to the adjacent nodes. In the designed model of this study, for every zone, there is a central node that is later connected to central nodes of other zones via thermal paths across the walls and windows. A schematic of this node and its connections can be seen in figures 3.8 and 3.9.

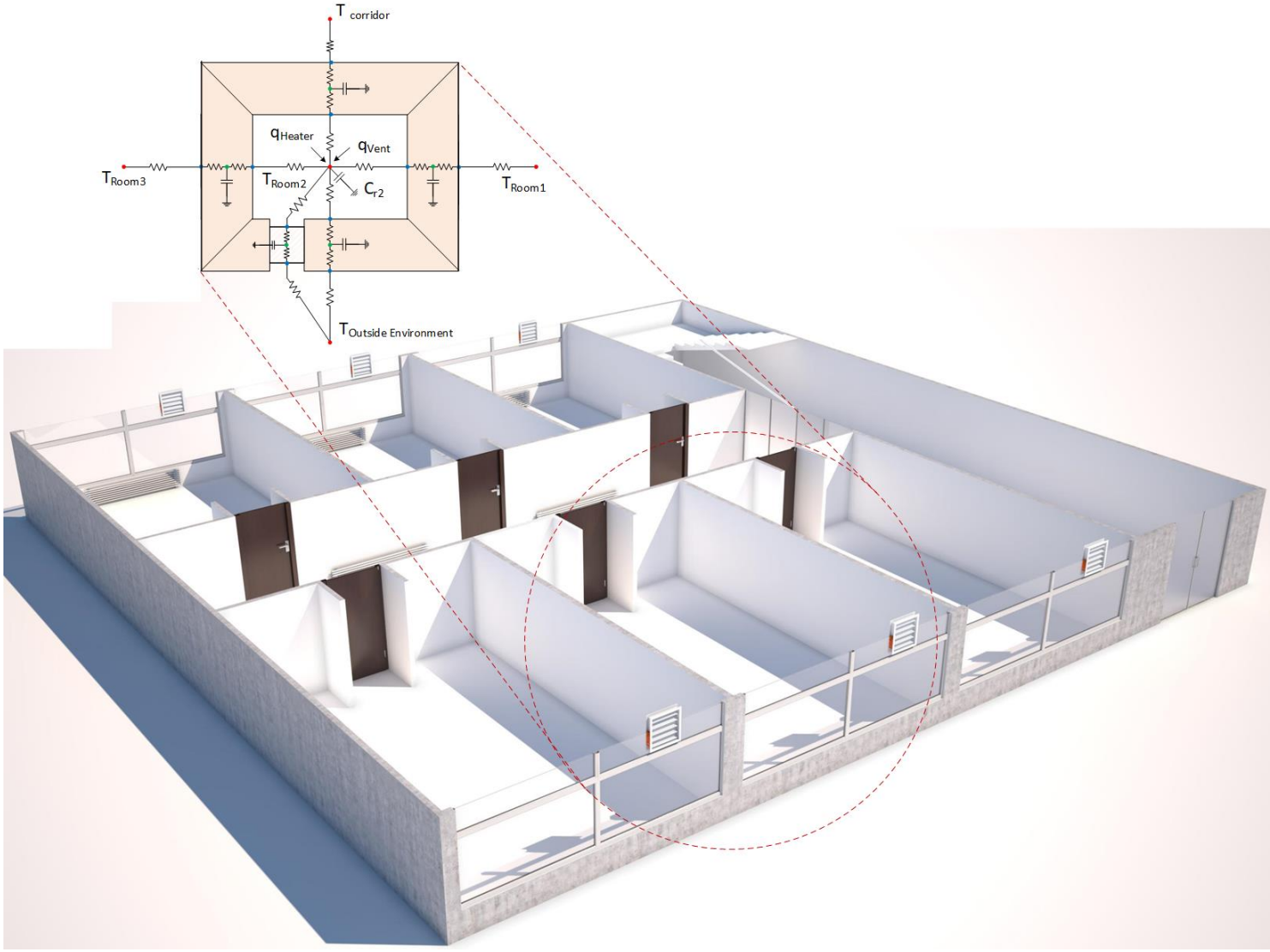


Figure 3.8 Schematic of the equivalent lumped capacitance model for a typical room.

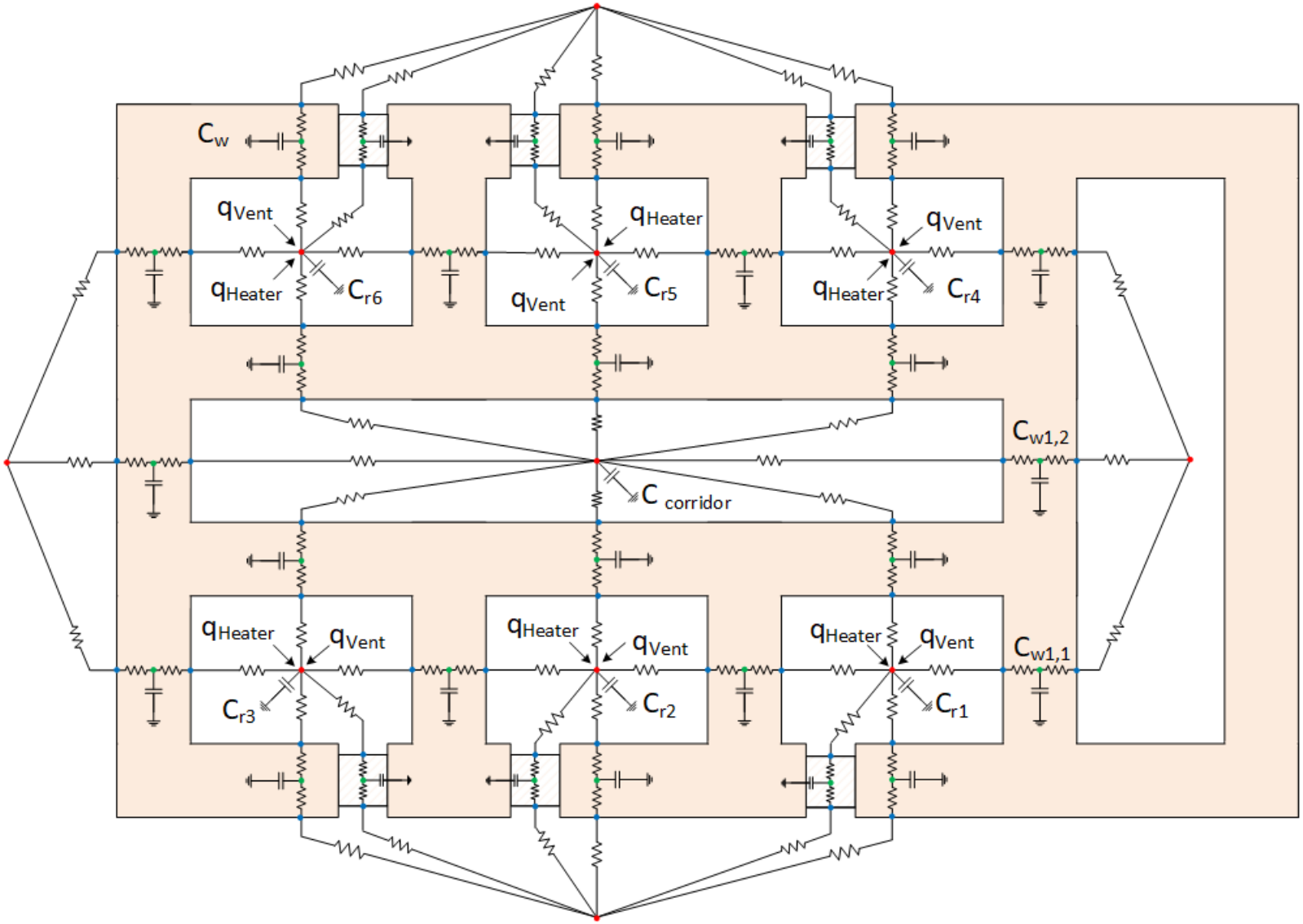


Figure 3.9 Equivalent lumped-capacitance model for example scenario office building.

3.2. DEMAND-CONTROLLED VENTILATION SYSTEM

A DCV system is a control strategy based on the ventilation concept. This system modifies the amount of fresh air to improve the indoor air quality while increasing the potential energy saving by automatic adjustment of the volume of air exchange using damper actuators based on sensor values received from air quality sensors, e.g., CO_2 concentration sensors, temperature sensors, occupancy sensors, heater status, and control theory. Figure 3.10 shows the role of the DCV and heating system in optimizing the system considering different parameters.

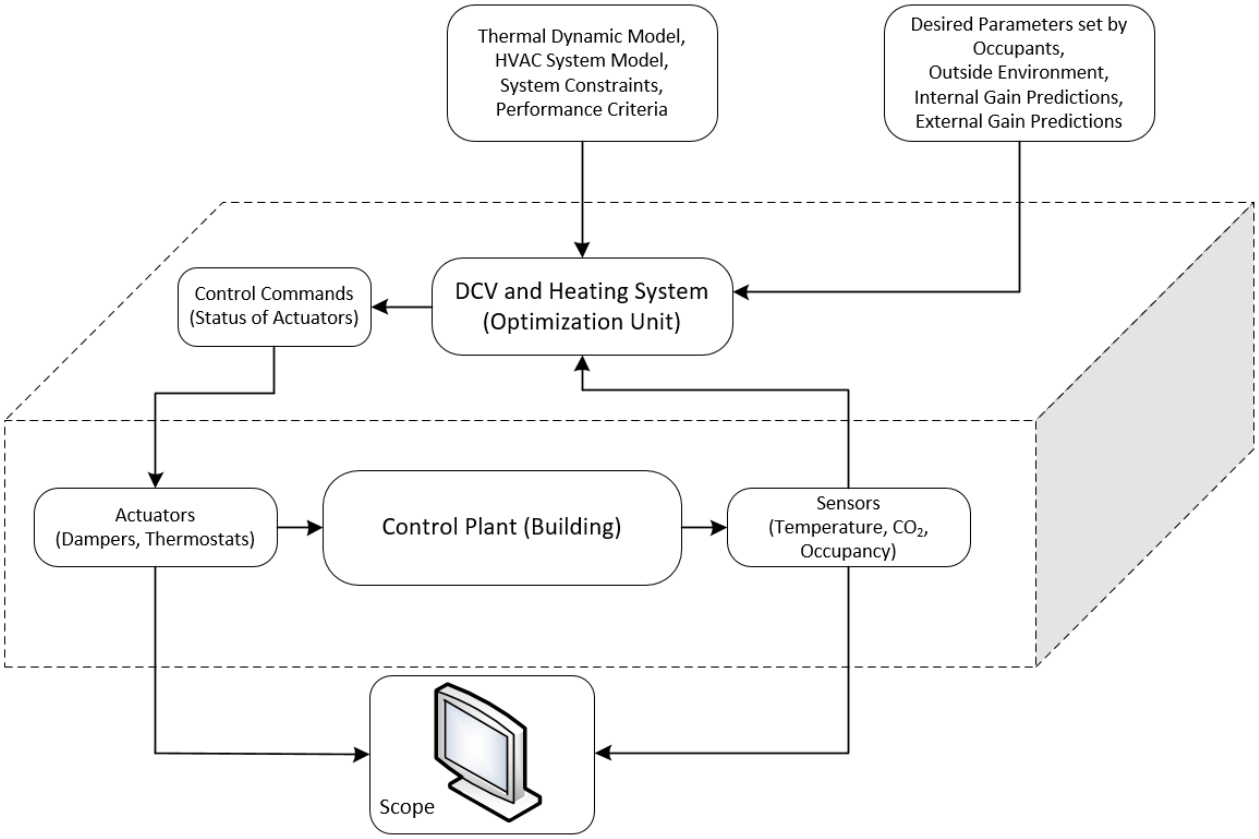


Figure 3.10 Role of DCV and Heating system in optimizing the system.

The air exchange includes the inward flow of the fresh air into the zone and the outward flow of polluted air from the zone. Natural ventilation is the oldest and prevalent method used for air exchange throughout history, which is also popular in European countries. The idea came up in the ancient era when people started using construction to capture the wind for the aeration to their living spaces. *Windcatcher,* "Baadgir" (*bâdgir*: *bâd* "wind" + *gir* "catcher") in Persian, is a famous example in history as a traditional Iranian architectural element which was constructed in many hot-arid-climate regions of Iran, especially in the city Yazd, to produce natural ventilation [138]. Dolatabaad garden is a prominent place because of its famous windcatcher (Figure 3.11), which is open on eight sides. These wind catchers were constructed open at one, two, four, or eight sides at the top of the tower (Roaf, 1989) that the wind forces the air into the building on the windward side and removes the high-temperature air from indoor based on the Buoyancy force (stack effect) to the outside (the leeward side). Therefore, the breeze could be captured from any direction [139].

Figure 3.11 A traditional windcatcher in Dolatabaad garden, Iran[3].

The indoor air quality depends on various parameters, such as the concentration of hovering air pollutants in the gas form and the dynamic performance of the HVAC system. Human metabolism needs fresh air, including oxygen, for the oxidation process of carbohydrates, fats, and proteins that produces mainly carbon dioxide and vapor. Natural ventilation can meet this metabolic requirement besides the dilution of the indoor air containment, odor, and pollutants for an acceptable IAQ and to supply the oxygen needed for the combustion process of the laboratories and exchange of the exhaust gas with the outside environment plus its capability of energy saving [140], [141]. The injection of fresh outdoor air with lower pollutants into the building zones is always a promising way to dilute the gaseous indoor air contaminants. Therefore, indoor air quality measures are a function of the mentioned parameters with the indoor and outdoor sources. Natural ventilation is the air movement through building openings due to wind, stack effect, or static pressure created by differences in temperature between the interior and exterior of the building, known as the stack effect [142]. Building openings used for natural ventilation are typically windows, doors, stacks, vertical flues, roof ventilators, wind catchers, and other-purpose-designed openings. Natural ventilation due to its simple design, installation, operation, and maintenance is ubiquitous. The ventilation rate indicates the dependency of the outdoor air quality to the indoor air quality and is the amount of outdoor air delivered into the building space. A precise control approach on the ventilation rate is vital as insufficient air exchange causes poor IAQ or children's health issues such as allergic diseases.

On the other hand, the heating and cooling loads increase during natural ventilation on cold and hot days, respectively. Significantly, too much fresh air exchange accounts for more waste of

[3] Photo credit to Bernard Gagnon, Pavilion in the Dolat Abad Garden, Yazd, Iran, 23 October 2016,
Link: https://commons.wikimedia.org/wiki/File:Dolat_Abad_Garden_-_Pavilion_02.jpg

energy. Therefore, reliable HVAC control is essential to maintain the utmost indoor air quality for occupants while balancing the energy consumption of the HVAC system.

Ventilation design determines how much ventilated air is required and why. In North America, the ASHRAE Standard 62-2007 standard is generally accepted to reach the best IAQ in buildings [143, 144]. For reaching an acceptable IAQ, a ventilation rate of 15 cfm per person is required and must be kept into consideration in winter and summer. This section describes the basic principles of natural ventilation. Three ventilation fundamentals are prevalently applied in the calculation of the natural driving forces in buildings that are illustrated in Figure 3.12:

1. Single-sided ventilation
2. Cross ventilation
3. Stack ventilation

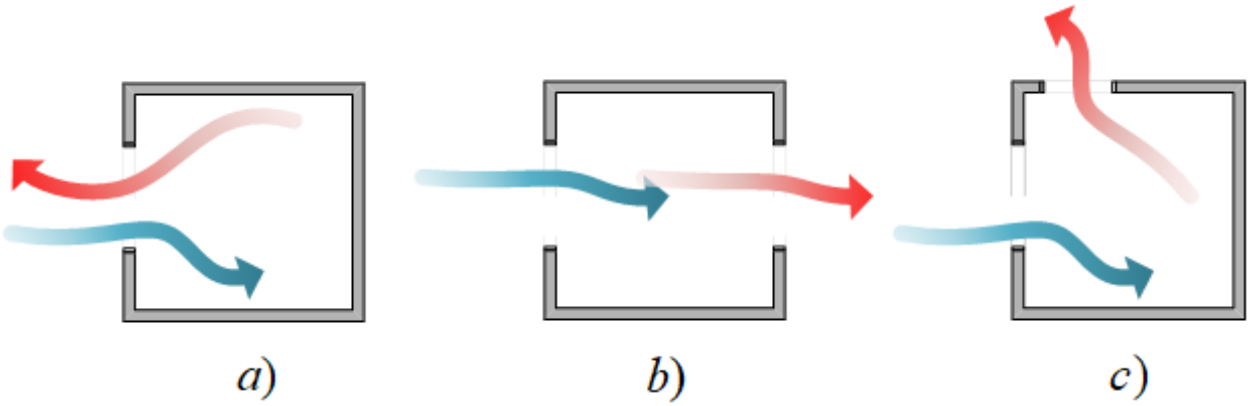


Figure 3.12 Natural ventilation types: a) Single-sided, b) Cross, c) Stack.

The ventilation fundamental shows in what way the exterior and interior airflows are linked. Furthermore, the ventilation principle indicates how the air is introduced into the building, and how it is exhausted out of it.

3.2.1. SINGLE-SIDED NATURAL VENTILATION

Single-sided ventilation is defined based on airflow through building openings on only one side of a room. The airflow enters and leaves from the opening(s) on the same wall. However, the ventilation capacity of single-sided ventilation is lower than cross ventilation, but it is more common and straightforward to implement based on architectural design limitations, especially in cellular office buildings with openings only on one side. Moreover, the privacy and security concerns, fire safety requirements, and control of air velocity issues result in a higher tendency towards using single-sided ventilation despite the higher efficiency of cross ventilation [145]. Some references refer to the building height/width (H/W) ratio as a crucial parameter. BRE digest suggests that to have effective single-sided ventilation, a window area must be at least 1/20 floor area and maximum room depth must be up to 2.5 times of the ceiling height (H) [146]. The same reference recommends cross ventilation for the depth of more than 2.5 H and up to 5 H because wind pressure can provide more effective flow rates [140].

The main driving forces of natural ventilation are the Buoyancy force (stack effect) and the wind (wind effect).

The Buoyancy effect: The air pressure and temperature difference across the opening causes a variation in density, and therefore the mass of the air flows inside the room; therefore, the low-density air goes upward, and the high-density air goes downward in a zone known as Buoyancy force. Awbi described stack ventilation as mostly suitable for cold and windy conditions with a form of single-sided ventilation, such as windows, for providing enough airflow rates [140, 147].

The wind effect: The wind forces generate an air exchange mainly due to the wind velocity depending on the incidence angle of the wind to the opening. When the wind effect is combined with the Buoyancy effect, a complicated relationship between the affecting parameters and the ventilation rate where either wind or buoyancy can be dominant.

The single-sided natural ventilation has advantages, e.g., most of the time is available, its usage is free without extra costs of ductwork and air handling units, it is accompanied by the energy-saving and silent operation as there is no fan. There are also disadvantages along with the single-sided natural ventilation, e.g., it highly depends on external climate conditions compared to the indoor spaces, lack of control over its magnitude, no defined exit route for air, poor ventilation is probable if the driving forces are not calculated correctly; and not suitable for deep plan spaces. The depth of penetration is limited to a maximum of approximately 2.5 times ceiling height, according to Awbi [140]. Practical issues and architectural design matters often limit the opening size. Single-sided natural ventilation also has limitations. Though natural ventilation is a reliable and cost-effective solution for air exchange, some cases necessitate the usage of mechanical ventilation, e.g., factories or industrial premises with a relatively high amount of dust or toxic containments, or hospitals where it is essential to control the airflow precisely besides its crossing through filters to remove airborne bacteria. Pavelchak et al. studied 140 specified airborne infection isolation rooms in 38 buildings from 1992 to 1998 [148]. The result shows unwanted directional airflow out of the patient room in around 38% of the buildings. Other examples are enclosed parking lots and the tunnel roads where it is essential to remove exhaust gases and their containments such as carbon monoxide (CO) or fuel vapors from the air. Commercial kitchens and lecture halls with lots of occupants.

3.2.1.1. VENTILATION PRINCIPLES

Fundamentals for the prediction of the airflow through a small and sharp-edged opening are described in this section. Each human breathes around 12000 liters of air per day, on average. Thus, indoor air quality has a significant effect on the occupant's health and efficiency. The recent building constructions are tighter to prevent energy loss through building envelopes. Therefore, pollutants emitted from occupants or building equipment are more likely to be trapped inside the occupied zones, which can cause severe health impacts on the individuals or dissatisfaction at least. Carbon dioxide (CO_2) is one of the most known odorless and colorless air pollutants. Carbon dioxide results from the oxidation of carbohydrates, proteins, and fats consumed by humans besides the energy release. Also, it can be produced from the fuel gas burning in the kitchen. There are two main techniques to remove pollutants from indoor spaces. One solution is crossing the air through filters. The other solution is to swap air with the outside environment. The first solution needs a considerable maintenance effort to change the filters or to clean them, but the latter seems more promising and is inexpensive and often available. However, the lack of control over natural ventilation is a significant issue that causes uncontrollable airflow rates due to high dependency on outdoor conditions. During the manual opening or closing of windows, there is no control over the outside weather and inside-zone conditions.

Recent progress in sensor technologies, besides their mass production, makes the sensor-based control systems noteworthy solutions for increasing the IAQ and decreasing energy usage and costs. A CO_2 sensor detects and shows the amount of carbon dioxide inside the air. The airflow and its direction through the building openings are time-dependent, accompanied by time-dependent temperature and pressure differences that cause the airflow rate to be unpredictable in magnitude and direction. Therefore, DCV tries to create a more predictable control approach for natural ventilation. A typical CO_2 sensor ranges from 0 to 9999 ppm for its measurements with an accuracy of 50 ppm $\pm 5\%$ [149]. Also, because of the rough relationship between the amounts of generated CO_2 by the occupants with CO_2 available in the air of a specific closed zone, a CO_2 sensor is helpful to determine the number of occupants in a closed zone. However, due to the air exchange in the

ventilated zone, the CO_2 sensor measurement is not a trustful value for estimating the number of occupants. The model of the CO_2 concentration sensor output signal is based on the calculation of CO_2 concentration shown by the following equation and frequently can be found in different references [11]:

$$Vol\frac{dC(t)}{dt} = G(t) + Q(C_0(t) - C(t)) \tag{3.25}$$

Where,

C = The indoor CO_2 concentration in [ppm],

t = The time in [s],

Vol = The building or space volume in [m^3],

G = The indoor CO_2 generation rate in [m^3/s].

Q = The building or space ventilation rate in [m^3/s],

C_0 = The outdoor CO_2 concentration in [ppm],

However, equation 3.25 needs correction in its unit aspect. All the terms are described in $m^3 \cdot ppm/s$ except the first term at the right side of the equation, which is not described in $m^3 \cdot ppm/s$ and is only in m^3/s. Therefore, to show this term in $m^3 \cdot ppm/s$, this must be multiplied by 10^6 that gives:

$$Vol\frac{dC(t)}{dt} = (G(t)\times 10^6) + Q(C_0(t) - C(t)) \tag{3.26}$$

In some references, the building or space ventilation rate (Q) is based on a rough calculation via some constant value N taken from fresh air requirement in the table of the standards, e.g., ASHRAE [12, 143, 150] by the equation below:

$$Q = N \cdot Vol \tag{3.27}$$

Where,

N = air change rate in [$1/h$].

In this study, a precise calculation is a goal to make the CO_2 concentration model more realistic. Therefore, the building or space ventilation rate is calculated that is the summation of the ventilation due to the wind and stack-flow (Buoyancy). Li et al. [56] showed a good match between the predicted results gathered from this equation and the experimental data on the modeling of indoor CO_2 concentration based on real-time indoor occupant prediction and CO_2 generation rates for an air-based HVAC system using ducts connected to a boiler (or chiller) as a central plant. The time constant for the system (V/Q) in a steady-state can also be calculated. The indoor CO_2 concentration rate is the summation of CO_2 generated by the occupants and the room CO_2 concentration. Referring to references such as Lu et al. [151], ASTM standard [152], and ANSI/ASHRAE standard 62.1-2016 [150], the CO_2 generation rate was considered 0.0052 L/s or 0.31 L/min for an adult sedentary employee at the activity level of 1.2 met units. Alternatively, the CO_2 generation rate can be calculated using the equation [56] below:

$$G = \frac{0.00276 A_D M}{(0.23 R_Q + 0.77)} \times R_Q \tag{3.28}$$

Where,

R_Q =The respiratory quotient (R_Q = $CO_{2\ eliminated}$ / $O_{2\ consumed}$),

M =The metabolic rate in [met],

A_D =The DuBois surface area in [m^2]

The respiratory quotient equals the volumetric rates of exhaled CO_2 divided by the consumed O_2 and is related to the diet type and physical activity. In this study, the respiratory quotient is considered 0.83 for a regular diet and sitting activities [153]. *M* is a parameter that shows the level of physical activity or indicates the metabolic rate per unit of surface area (MET), which is considered 1 for the seated office activity.

$$1 met = 58.2 W/m^2 \tag{3.29}$$

However, Persily et al. [154] recommend that 1 met equals 49.8 W/m^2 . He also suggests a new formula for the CO_2 generation rate (*G*), considering the effects of air pressure and temperature.:

$$G = R_Q \, BMR \, M \, (T/P) 0.000211 \tag{3.30}$$

Where,

BMR=The basal metabolic rate in [*MJ/day*],

P =The pressure in [Pa],

T =The temperature in [K].

The DuBois surface area with the unit m^2 and can be calculated by

$$A_D = 0.203 H^{0.725} W^{0.425} \tag{3.31}$$

Where,

H =The height of the person in [m],

W =The body mass in [kg].

Based on the equation above, this study assumes A_D as 1.83 m^2.

In natural ventilation systems, the fresh air is supplied through window openings. The basic principle of natural ventilation is based on thermal buoyancy and wind pressure on the building envelope. All the environmental factors such as temperature and pressure difference and architectural properties such as window sizes and types substantially affect ventilation rates. Different parameters affect the natural ventilation flow rates, e.g., building shape, height, location and the surrounding terrain, opening size and its position, airtightness and the leakage distribution, outside weather conditions, and wind characteristics such as air velocity and its direction.

Single-sided ventilation is prevalent in office buildings. In this method, all the window openings are placed on one side of the room. The window type in this thesis is the bottom-hung inward opening. Ventilation induced by buoyancy due to indoor/outdoor temperature difference (stack ventilation) is effective in cold conditions but not in warm conditions because outdoor is warmer than indoors.

A typical orifice model is used to model a typical sharp-edged window. Figure 3.13 describes a typical orifice with section areas A_1 and A_2:

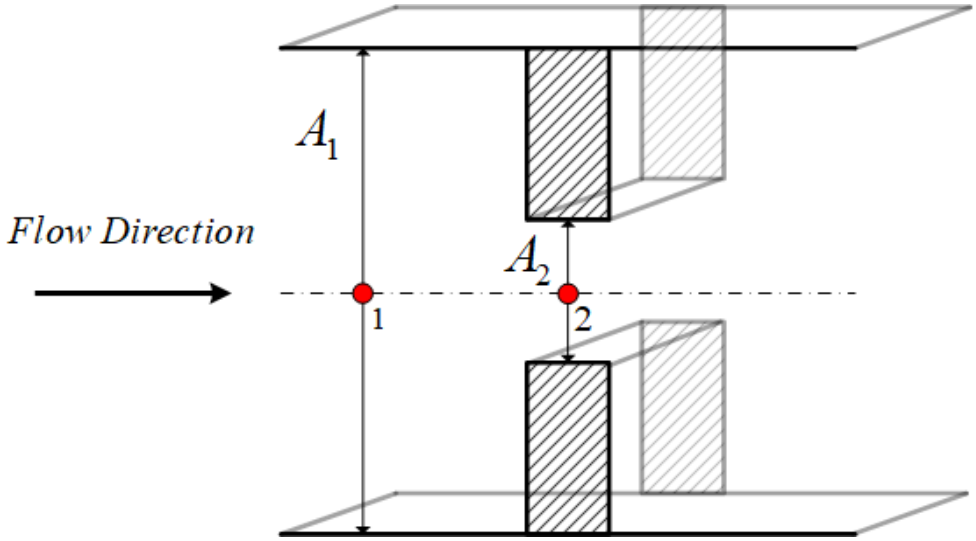


Figure 3.13 Schematic of an orifice plate.

When the conservation of energy is applied to the Bernoulli equation, then we have the following equations.

$$\frac{P_1}{\rho g}+\frac{V_1^2}{2g}+Z_1=\frac{P_2}{\rho g}+\frac{V_2^2}{2g}+Z_2+loss_{1-2} \tag{3.32}$$

Where,

Z = The height of measure point in [m]

V = The flow speed in [m/s]

P = The pressure in [Pa]

g = The gravitational acceleration in [m/s^2]

ρ = fluid (air) density in [kg/m^3]

Where $loss_{1-2}$ is the summation of the head losses over upstream, downstream, and across the orifice plate. Assuming that Z_1 is the same as Z_2 and $loss_{1-2}=0$, we have

$$\frac{P_1}{\rho}+\frac{V_1^2}{2}=\frac{P_2}{\rho}+\frac{V_2^2}{2} \tag{3.33}$$

The pressure drops across the orifice plate (ΔP) can be calculated by rearranging the equation above

$$P_1-P_2=\frac{\rho}{2}(V_2^2-V_1^1)=\frac{\rho V_2^2}{2}\left[1-(\frac{V_1}{V_2})^2\right] \tag{3.34}$$

The law for the conservation of mass, which is also known as the equation of continuity, describes that the volume flow rate (Q) across the orifice remains the same, in mathematical form.

$$Q_1=Q_2 \tag{3.35}$$

Or

$$V_1 A_1 = V_2 A_2 \tag{3.36}$$

Substitution of equation 3.36 in equation 3.34 gives:

$$V_2 = \sqrt{\frac{2(P_1 - P_2)}{\rho\left[1 - (\frac{A_2}{A_1})^2\right]}} \tag{3.37}$$

Where in the natural ventilation concept, the outside environment area can be estimated to an infinity value ($A_1 \cong \infty$), and this gives

$$V_2 = \sqrt{\frac{2(P_1 - P_2)}{\rho}} \tag{3.38}$$

Knowing that the average air velocity through the opening is $V_{avg} = V_2$, $A = A_2$, $Q_{theoretical} = Q_2$, and $\Delta P = P_1 - P_2$ gives:

$$V_{avg} = \sqrt{\frac{2\Delta p}{\rho}} \tag{3.39}$$

and

$$Q_{theoretical} = A \cdot V_{avg} \tag{3.40}$$

Substituting equation 3.39 in equation 3.40 gives:

$$Q_{theoretical} = A\sqrt{\frac{2\Delta p}{\rho}} \tag{3.41}$$

The actual volume flow rate crossing a sharp-edged window is always less than the theoretical volume flow rate due to the sharp-edge resisting effect. The discharge coefficient (C_d) is an essential parameter in the sharp-edged orifice plate design, which is defined as the ratio of the actual flow rate (Q_{actual}) to the maximum theoretical volume flow rate ($Q_{theoretical}$) [155]. The discharge coefficient value of 0.61 for the sharp-edged rectangular opening is used [156].

$$C_d = \frac{Q_{actual}}{Q_{theoretical}} \tag{3.42}$$

Furthermore, Q_{actual} can be calculated using equation 3.42 by:

$$Q_{actual} = C_d A\sqrt{\frac{2\Delta p}{\rho}} \tag{3.43}$$

The inward airflow due to wind and the temperature difference between the indoor and the outdoor environments through a window opening is calculated using equation 3.43, also demonstrated by H. Awbi [140]. This method is known as The British Standard Method [142]. The wind pressure and the temperature difference (stack effect) between indoor and outdoor create pressure differences across the building envelope. This model did not include mechanical ventilation. Therefore, the effective cross-section area of the flow through the opening (A_c) which is shown in Figure 3.14 can be calculated as:

$$A_c = C_d A \tag{3.44}$$

where

A_c = effective damper area in m^2

C_d = discharge coefficient

A = damper (opening) area in m^2

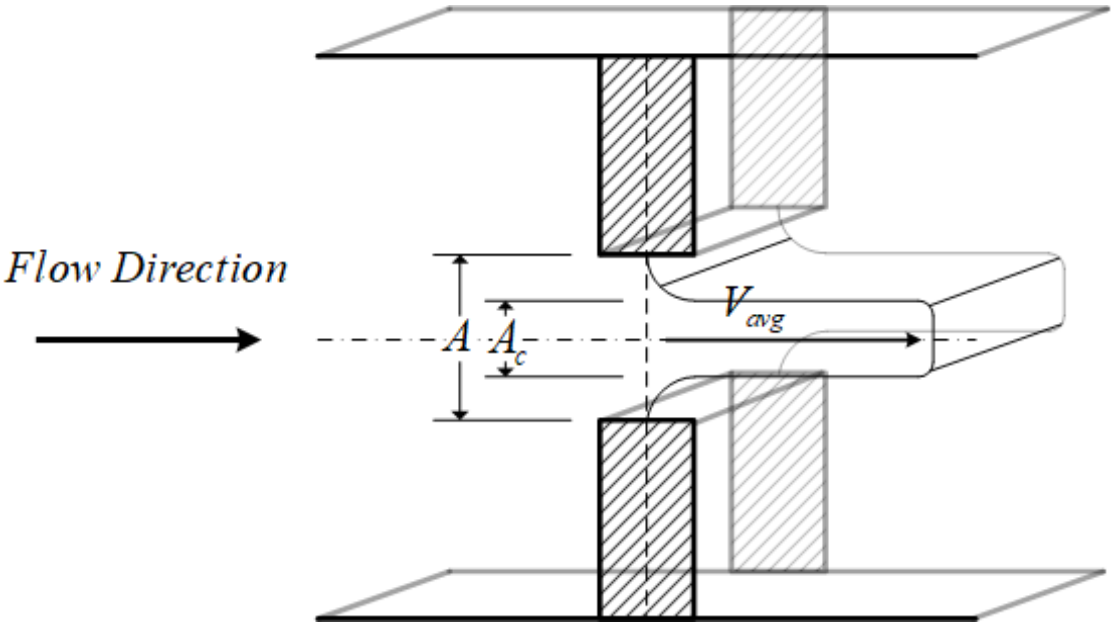


Figure 3.14 The effective area of an opening.

The effective area of multiple openings in a series of openings in one facade can be calculated by

$$A_c = C_d A = C_{d1} A_1 + C_{d2} A_2 + \cdots \tag{3.45}$$

and in a parallel layout of the openings on opposite sides, it can be calculated by

$$\frac{1}{A_c^2} = \frac{1}{(C_d A)^2} = \frac{1}{(C_{d1} A_1)^2} + \frac{1}{(C_{d2} A_2)^2} + \cdots \tag{3.46}$$

The building ventilation rate in this thesis is calculated by accumulating flows through an opening, including Buoyancy-driven flow and wind-driven flow. For Buoyancy-driven flow (stack flow) through an opening, as the height effect on the pressure difference is insignificant, the overall pressure difference can be calculated based on only the density difference between indoor and outdoor. Thus, by substitution for the term pressure difference from the Bernoulli equation 3.32 on a window on a wall, therefore, $V_2 = V_1$ and $loss_{1-2} = 0$, the equation 3.43 can be written as:

$$Q_{Buoyancy-driven} = C_d A \sqrt{\frac{2(\rho_o - \rho_i) g (h - H_{NPL})}{\rho_o}} \tag{3.47}$$

The Boussinesq model [157] is used for the approximation to eliminate density terms (ρ):

$$(\rho_o - \rho_i) g \cong -\rho_o \beta (T_o - T_i) g \tag{3.48}$$

Where,

β =The coefficient of thermal expansion.

For ideal gases, there is the expression below:

$$\beta = \frac{1}{T_i} \tag{3.49}$$

Therefore:

$$Q_{Buoyancy-driven} = C_d A \sqrt{\frac{2(T_i - T_o) g (h - H_{NPL})}{\rho_i}} \tag{3.50}$$

Where,

H_{NPL} = The neutral pressure layer height

h = The certain vertical height along with the opening

g = The gravitational acceleration in [m / s^2]

T_i = The inside temperature in [K]

T_o =The outside temperature in [K]

For single-sided natural ventilation of a large opening with a maximum height of H, if H_{NPL} is half of the opening height ($H_{NPL} = H/2$) then figure 3.15 shows a better view of the airflow through the opening:

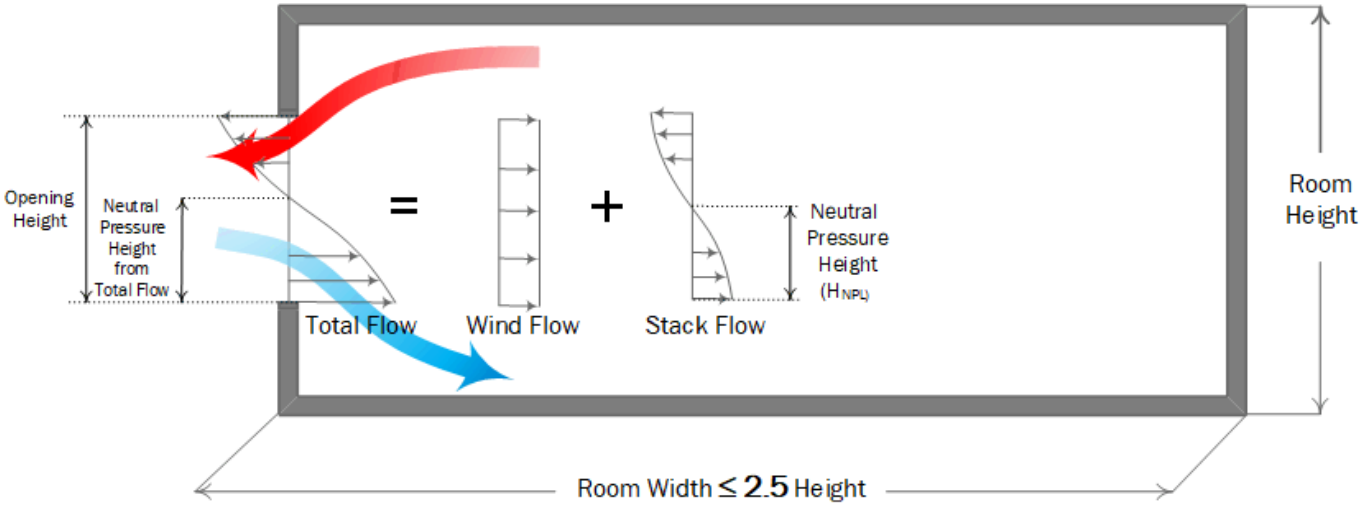


Figure 3.15 Schematic of the single-sided ventilation.

By substituting equations 3.48 and 3.49 in equation 3.50, the Buoyancy-driven flow is calculated by equation 3.51:

$$Q_{Buoyancy-driven} = C_d A \sqrt{\frac{gH\left(T_i - T_o\right)}{T_i}} \tag{3.51}$$

For wind-driven flow through an opening, the pressure difference is calculated based on the pressure coefficient (C_p), which is discussed in many references [158], [159], and [147]. The pressure coefficient is calculated by dividing the difference between the pressure on a surface and the static pressure of the collided wind to the design wind pressure. The static and design wind pressures can be calculated at the height of the target point considering the geographical and environmental properties:

$$C_p = \frac{P_w - P_o}{\frac{1}{2}\rho V_r^2} \tag{3.52}$$

Where,

C_p = The pressure coefficient at the opening [-],

P_w = The static pressure at the opening surface on the building in [Pa],

P_o = Static pressure of the free stream (ambient or atmospheric pressure) in [Pa],

ρ = The air density of free stream in [kg/m^3],

V_r = The reference wind velocity in the free stream in [m/s].

By substitution of equation 3.52 in equation 3.43, equation 3.53 gives the wind-driven flow:

$$Q_{wind-driven} = C_d \cdot A \cdot V_r \sqrt{C_p} \tag{3.53}$$

For the combined flow through an opening, including wind and Buoyancy (stack) effects, it is possible to calculate the flow rate respective to each mode and combine them in various superposition methods. There are many methods known. However, the simplest one illustrates that the total ventilation flow is the linear addition of each flow and the respective formula is demonstrated by

$$Q_{total} = Q_{wind-driven} + Q_{Buoyamcy-driven} \tag{3.54}$$

Awbi et al. (2010) [160] and Shaw (1985) [161] introduced another superposition method which is known as the simple pressure method, as the following:

$$Q_{total} = \left(Q_{wind-driven}^{1/n} + Q_{Buoyancy-driven}^{1/n} \right)^n \tag{3.55}$$

Where $0.5 < n < 1.0$ depends on the opening dimensions and type of flow, e.g., 0.5 for fully turbulent flow or 1 for fully laminar flow, a value of 0.67 has been suggested for a small opening, and for the large opening, its value is 0.5 [160]. Therefore, the established model of this study can calculate the results based on different superposition methods.

From the heat transfer, the heat rate is defined by equation 3.13:

$$q = mc_p \frac{dT}{dt} \tag{3.13}$$

Moreover, from the density definition, it is known that:

$$m = \rho V \tag{3.56}$$

Substituting gives the heat load due to the natural ventilation inside the building:

$$q_{vent} = \rho Q_{total} c_p \Delta T \tag{3.57}$$

Where,

q_{vent} = The heat load due to the natural ventilation in watts [W],

c_p = The specific heat capacity in [$kJ/kg\,K$],

ρ = The density of outdoor air in [kg/m^3],

ΔT =The temperature difference between inside and outside of the room in [K].

Hall et al. recommend a value of 1.2 kg/m^3 for the outdoor air density and 1 $kJ/kg\,K$ for the specific heat capacity of the air [162]. This thesis considers a value of 1.237 kg/m^3 for the outdoor air density and 1.006 $kJ/kg\,K$ for the specific heat capacity of the air.

3.3. MODULAR COMPOSABILITY OF HVAC MODELS

This thesis introduces the modular composability scheme using generic diagnostic components at the system level applicable in many areas, especially in cyber-physical systems. This scheme is described in this section and implemented in chapter four using a DCV and heating system for the office buildings. The methodology of composability was used in the field of computer security that describes constructing an extensive complex system from a standard set of smaller systems that are independently verified [22]. Non-interference property prevents influencing high-level and low-level behaviors on each other when either composable or a method of composition exists, which is a security property [22]. Implementing the composability technique is not easy as individual components have different specifications compared to their combined form [163]. Composability intensifies the quality of modeling and simulation in system design [164], [165]. Composability is applicable in many fields, such as cyber-physical systems that include systems of systems (SOS) and compositions of diverse subsystems. Modularity is a design technique for subsystems (modules) with well-defined interfaces to create scalable models that can be used in various contexts. The unique modules are used to create various complex building models. The main aspects of composability modeling in the system design phase are integrating models in different disciplines and fields, the abstraction of the model description, restrictions and constraints of the development tools, and the level of composability of the model components [166]. Abstraction is a simplified representation of the model that eliminates some characteristics to keep only a set of essential characteristics, and programmers hide some unwanted details to reduce the complexity and increase efficiency.

In the composability, the properties are preserved by the composition of architectures [167]. From this point of view, a composability method prevents the risk of maintaining their characteristic properties while combining them. Considering the composition operator $\oplus$, architectures A_1 and A_2, each with a set of components B that make composite components $A_1(B)$ and $A_2(B)$, each architecture satisfies a characteristic property Φ_1 and Φ_2, the composition of A_1 and A_2 that is written as $A_1 \oplus A_2$ preserves characteristic properties of both or meets $\Phi_1 \wedge \Phi_2$. Safety and fault tolerance are examples of the characteristics of the property. As the composition operator $\oplus$ is associative and commutative, we will have:

$$\oplus_{i=1}^{m} A_i = \oplus_{i\in[1,m]} A_i = A_1 \oplus \cdots \oplus A_m \tag{3.58}$$

"Considering the interaction model γ which is a set of interactions, each interaction a is a set of actions ($a \in \gamma$) by the synchronously executed composed components" [167]. If C is a set of coordinating components, we have:

$$A(B) = \gamma(C, B) \tag{3.59}$$

In this thesis, each type of building architecture A is a solution to a specific coordination problem with a specific characteristic property. Definitions of components and interaction model are as follows:

A *component* is a Labelled Transition System $B=\left(Q,\ q^0,P,\rightarrow\right)$, where Q is a set of states, $q^0\in Q$ is the *initial state*, P is a set of *ports* and $\rightarrow\subseteq Q\times(2^P\setminus\{\phi\}\times Q$ is a *transition relation*. Each transition is labeled by an *interaction* $\phi\neq a\subseteq P$. P is the interface of B [167].

An interaction model over P is a set $\gamma\subseteq 2^P$ where the set ports $P=\bigcup_{i=1}^{n}P_i$ and P_i are pairways disjoints, i.e., $\forall i\neq j, P_i\bigcap P_j=\phi$ and $B=\{B_1,B_2,...,B_n\}$ is a finite set of components with $B_i=(Q_i,q_i^0,P_i,\rightarrow)$ [167].

An *architecture* that includes any set of components and dangling ports is defined by $A=(C,P_A,\gamma)$ such that $\bigcup_{C\in C}P_C\subseteq P_A$ and $\gamma\subseteq 2^{P_A}$ is an interaction model over P_A.

The composition of architectures is the conjunction of their respective constraints and is formalized based on the following definition. Let $A_i=(C_i,P_A,\gamma_i)$ for $i=1,2$. Therefore, the composition of architectures A_1 and A_2 is also an architecture $A_1\oplus A_2=(C_1\cup C_2,P_{A_1}\cup P_{A_2},\gamma_\varphi)$, where $\varphi=\varphi_{\gamma_1}\wedge\varphi_{\gamma_2}$. Considering γ_1 and γ_2 as the corresponding interactions of A_1 and A_2, these precisely satisfy the coordination constraints enforced by both composing architectures. The execution traces allowed by $A_1\oplus A_2$ on a set of components B are also allowed by both A_1 and A_2, which guarantees the preservation of properties by the composition of architectures.

This thesis applies the composability technique to handle the mentioned design challenge by automatizing various components' selection, combining, and configuration processes in different levels, considering requirements while reducing development costs. In this context, the inter-relationship among components and scalability are considered, which means the components can be combined in different levels and directions while keeping a certain and acceptable level of performance and efficiency [168], [169]. Therefore, the composable model version of the DCV and Heating system is developed. For this reason, individual components, i.e., sensors, actuators, standard rooms, enclosures, and windows, are modeled as subsystems and placed in the library.

4. SIMULATION FRAMEWORK OF THE DCV AND HEATING SYSTEM

This chapter describes the simulation framework of the DCV and heating system in MATLAB/Simulink. The component-based integration technique of the DCV and heating system is discussed based on the composability. The overall system model is structured into simulation building blocks. The C code can be synthesized for the control algorithms using MATLAB coder, and it can be integrated into real-time embedded systems. The simulation framework is established as a prototype to test and evaluate diagnostic techniques described in the following chapters. The simulation model is based on the analytic models described in chapter three and is extended for the wireless communication infrastructure.

4.1. MODELING AND SIMULATION OF THE SYSTEM MODEL

This part describes the implementation of the models and the simulation of the system model. The analytical model developed in chapter three is experimentally implemented in MATLAB/Simulink. Analytical models are prevalently used because their development includes mapping the physical and thermodynamic laws governing the system's dynamics to mathematical relations using variables and constants that are easy to manipulate and evaluate [170].

The modeling structure includes building blocks of input variables, output variables, and the primary system. The input variable block includes controllable variables, e.g., internal heat gains, set points of the actuators, and uncontrollable variables, e.g., solar gains, outdoor air specifications such as temperature, and wind speed. The output variable block includes building outputs such as room temperature, CO_2 concentrations, and energy consumption. The modeling of the demand-controlled ventilation and heating system, which considers the thermal and air-quality demand is complex due to several nonlinearities and different factors. Some factors are:

- Physical characteristics and system specifications: Thermal capacity, heat transfer coefficients, U-values and building elements' material,
- Local weather condition: temperature, solar radiation angle, wind speed, and pressure,
- Heating system specifications: efficiency and control method,
- Occupants requirements and behaviors,
- Internal air quality such as desired fresh air and desired temperature,
- Air pollutant's parameters such as CO_2,
- Internal heat gain: depending on the number of occupants and appliances as the heat sources.

Therefore, a suitable building performance model and simulation tools are needed to evaluate the entire system based on the mentioned parameters. This work is an optimization challenge to achieve an optimal balance among heating load (thermal comfort), zone temperature, and indoor air quality. Further, the simulation tool helps develop reliable failure detection and fault diagnosis techniques in case of fault occurrences.

This work presents models developed in various stages dynamically simulated by MATLAB/Simulink using the Simscape toolbox. The automatic code generation feature of MATLAB/Simulink allows deploying the simulated model on real-time processors and actual hardware platforms. The Simscape blocks illustrate the physical system associated with their schematic and not only by their abstract mathematical equations [10]. The model considers the

heating and ventilation (not the cooling) because the model studies the winter season when the range of outdoor temperature is below the acceptable indoor temperature range. Figure 4.1 shows the office building sketch of the Chair for Embedded Systems at the University of Siegen, Siegen, Germany. The model was established with thermal dependencies among different rooms or spaces and the outside environment during a typical winter day in February. The model dynamics consist of various equations and coefficients that show the heat transfer effects of different nodes of various building zones on each other. The differential balance equations for each node have been solved with an explicit numerical method using MATLAB/Simulink ode45 solver.

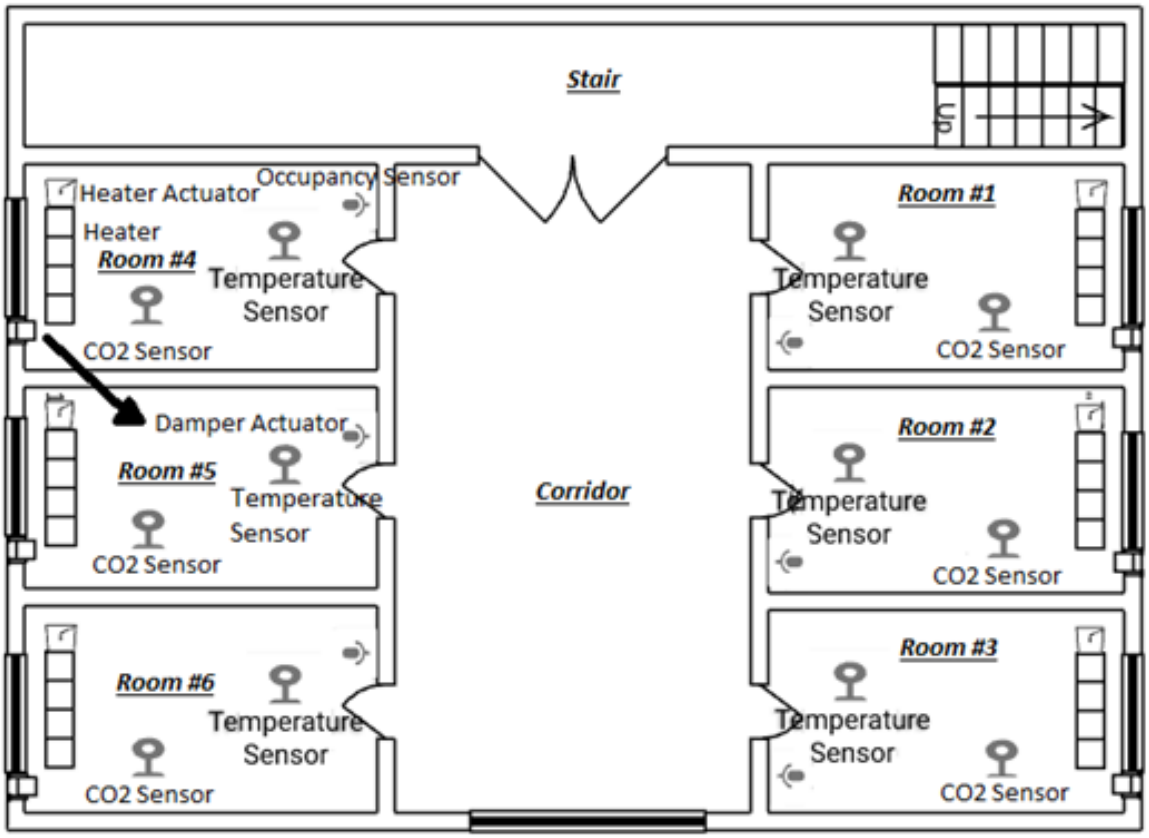


Figure 4.1 Office building sketch [24].

Figure 4.2, Figure 4.3, and Figure 4.4 indicate the vertical and horizontal view of the office building in the first stage of the development with only six rooms and one corridor.

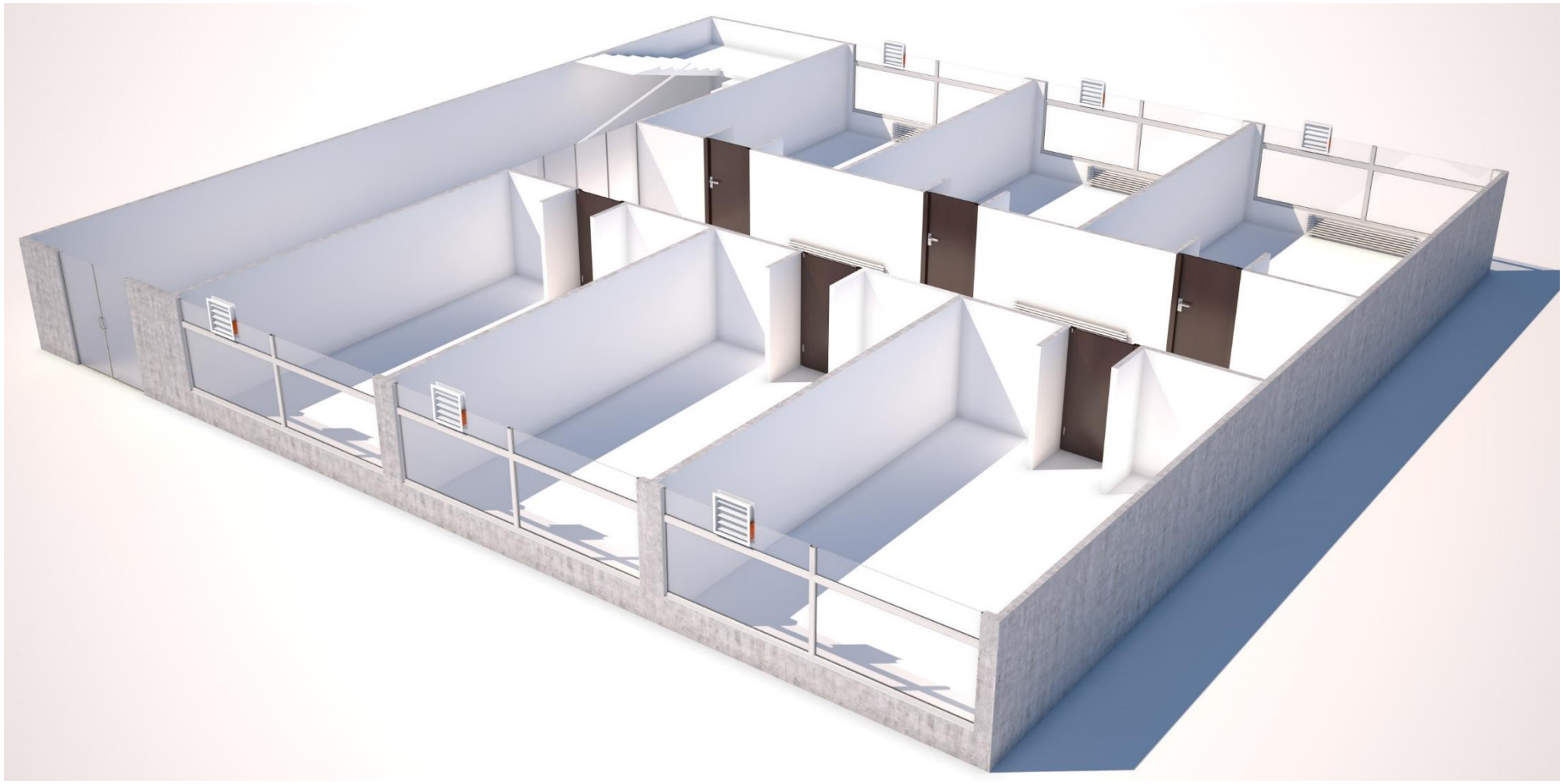


Figure 4.2 Southwest view of the building plan.

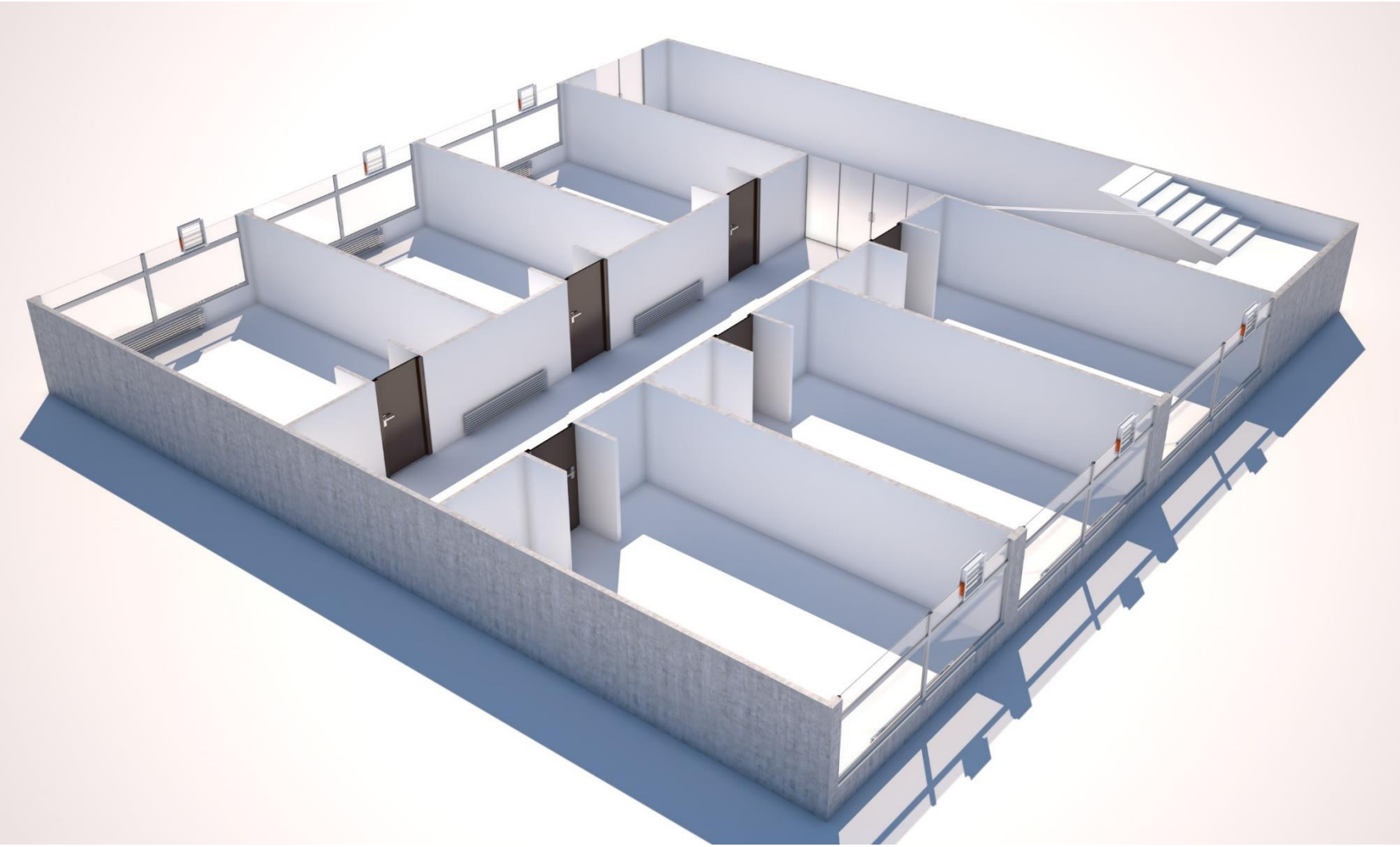


Figure 4.3 Southeast view of the building plan.

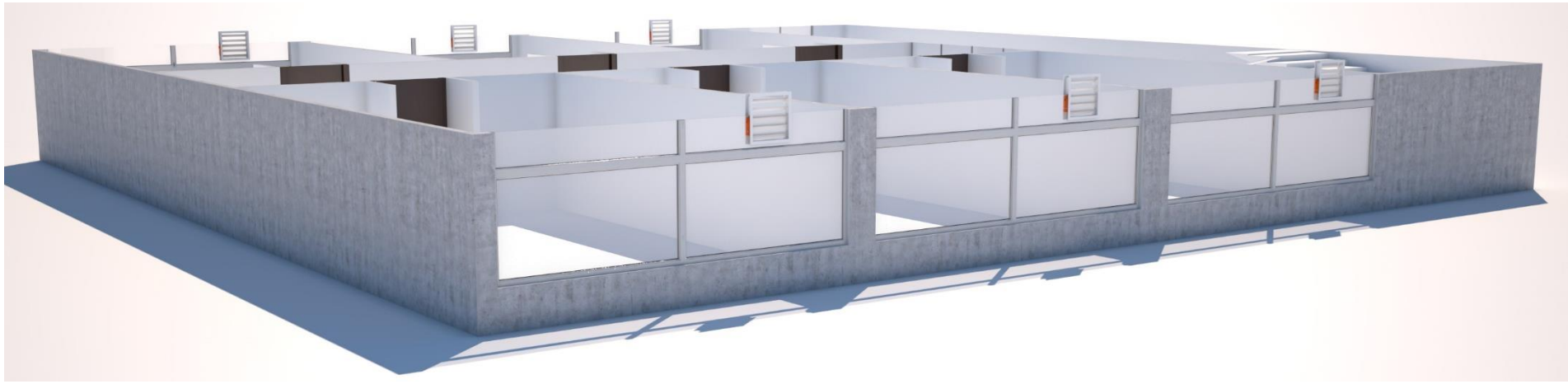


Figure 4.4 Horizontal view of the building.

Figure 4.5 represents the highest Simulink view of the office building model. The blue-colored block, which is named the main block, includes different zones of the building and their interconnections as subsystems. The input signals to the main block are the set parameters and input variables/functions, e.g., the daily temperature signal, the upper flow temperature signal, stair temperature signal as an adjacent zone, and the thresholds. The Simulink multiplexer mux block combines the output signals from every zone, and the composite signal is directed to the scope block for monitoring the signals over time.

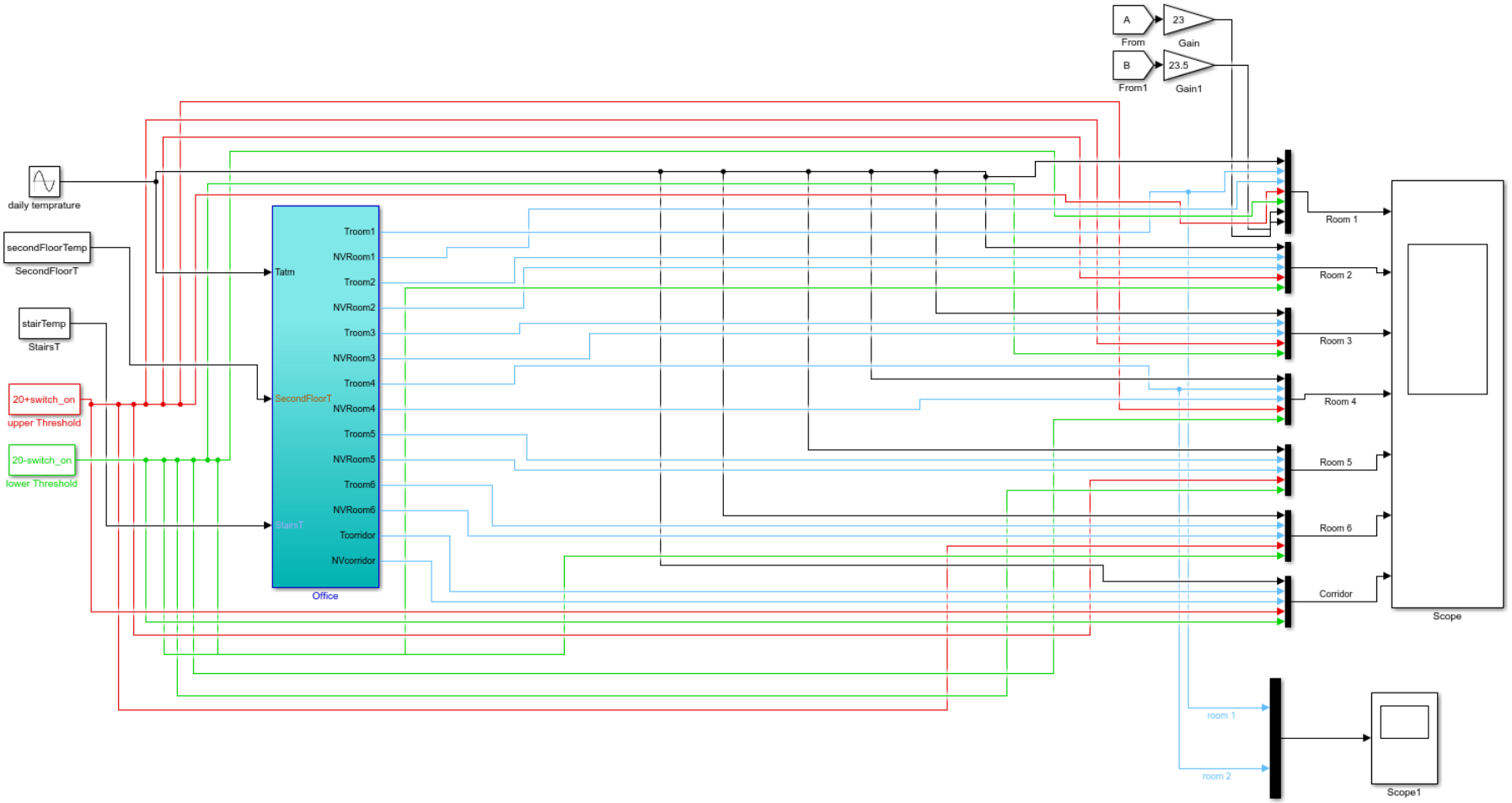


Figure 4.5 The top-level Simulink view of the office building model.

Figure 4.6 shows the Simulink model of the building, including six rooms and one corridor.

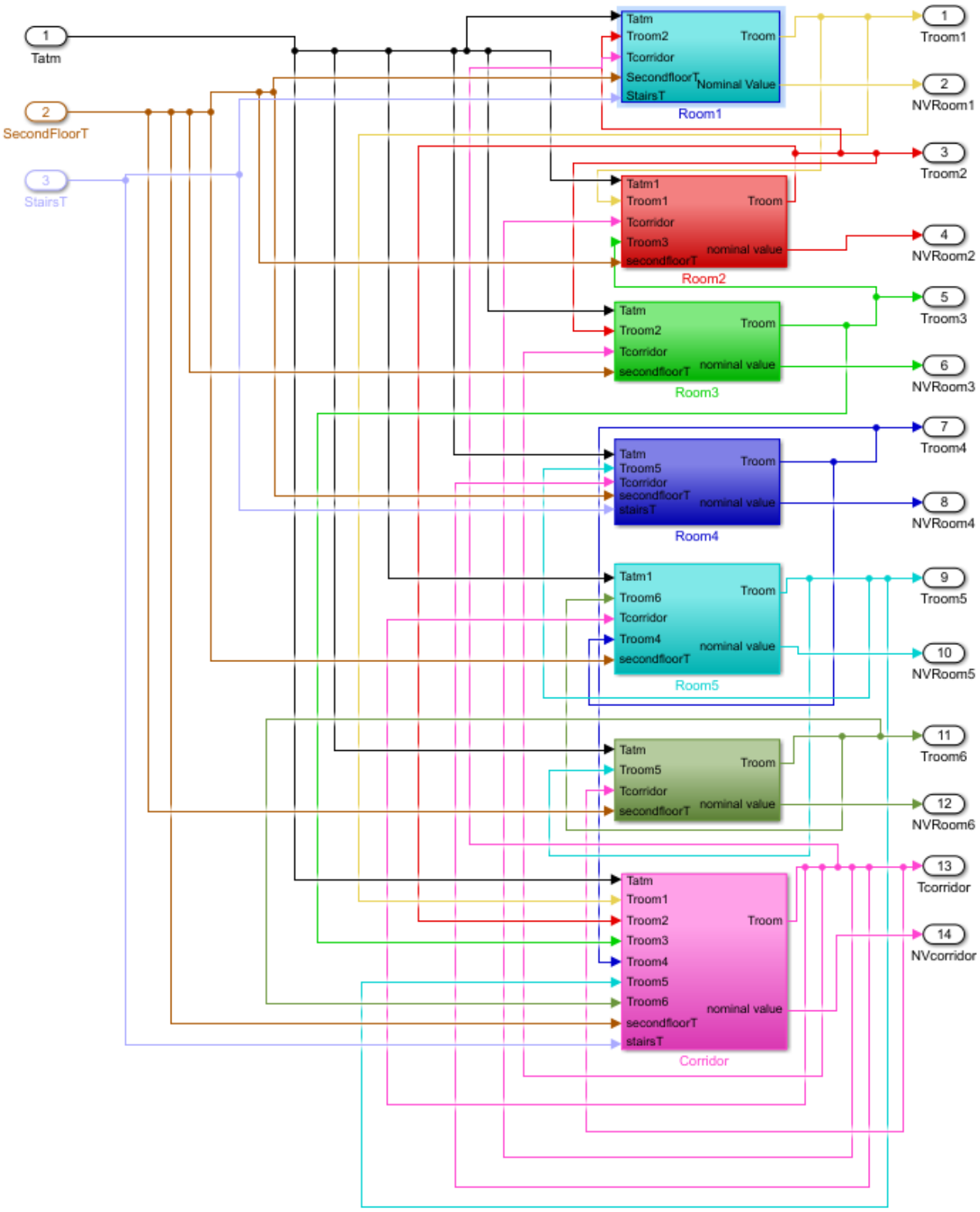


Figure 4.6 Simulink model of the building, including six rooms and one corridor.

Each room has its block (subsystem) and can be connected to adjacent zones from its four sides horizontally and two sides vertically. Therefore, the temperature signals of the adjacent rooms are connected to the corresponding blocks.

The outside temperature (daily temperature) is modeled as an analog signal, namely a sine wave (sinusoid) with an amplitude of 5, a bias of ± 7, and a frequency of $2\pi/(24*3600)$ in $rad/\sec$. Therefore, the T_{atm} signal is produced for one day (24*3600 seconds), and the phase of 0 can be seen in Figure 4.7.

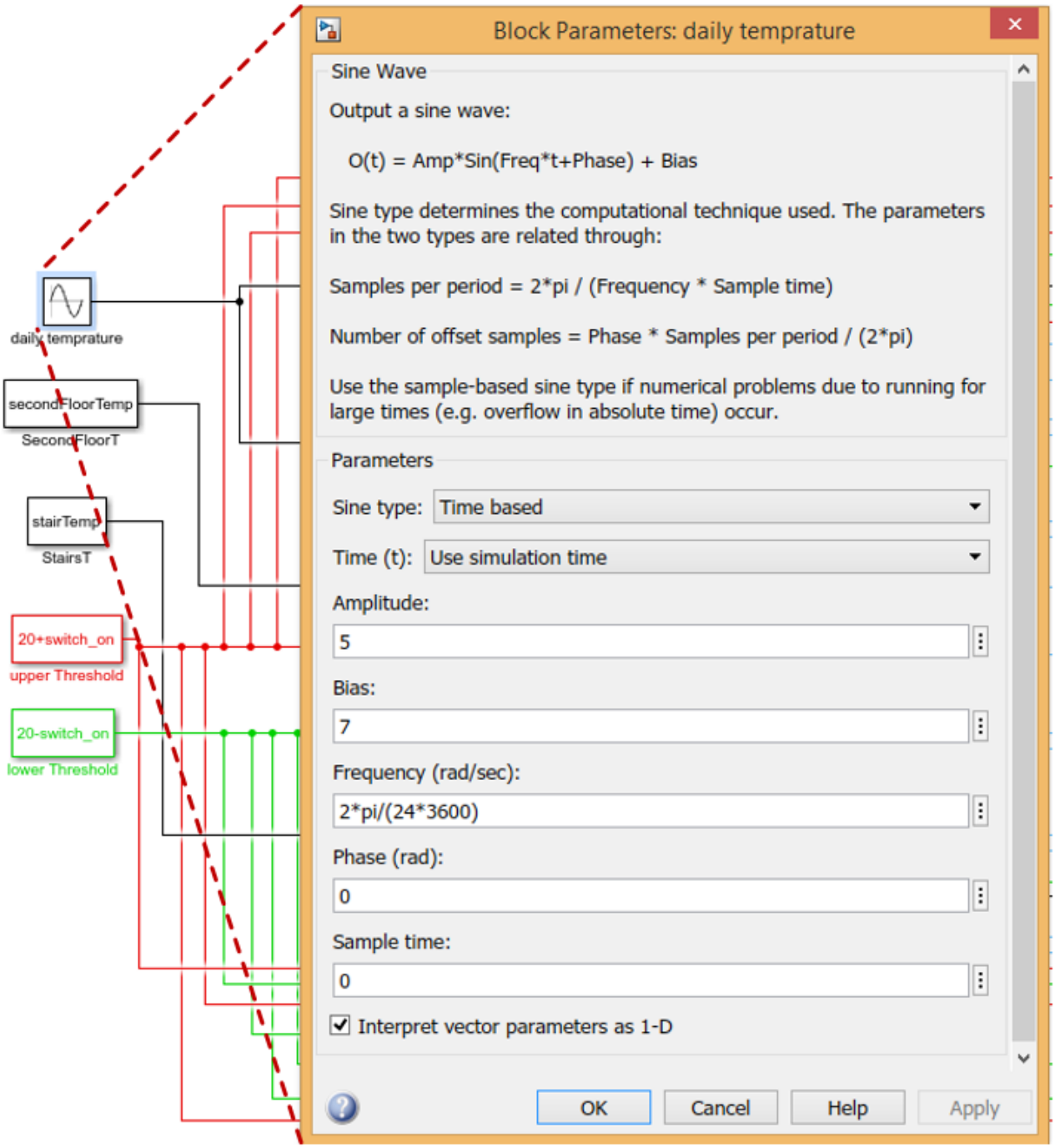


Figure 4.7 Daily temperature.

Figure 4.8 demonstrates the Simulink model for one room. The room block consists of three subsystems: heater subsystem, room thermal subsystem, and damper subsystem. The nominal values after insertion by GUI will appear, e.g., a nominal value for temperature that is the desired room temperature set by the user and the nominal value for the desired CO_2 concentration $\pm$ upper/lower thresholds. There are two controllers: heater controller (thermostat) and CO_2 controller (or damper controller). The thermostat controls the heater's status (ON/OFF), and the damper controls the status of the damper (OPENED/CLOSED). The heater subsystem produces the heat flow signal and sends it directly to the room thermal subsystem as its input to compose the heat flows within this block. The temperature signals of the adjacent zones and the ventilation load calculated by the damper subsystem are also inputs to the room thermal subsystem. The room thermal subsystem processes all the heat flows based on the heat transfer principles discussed in chapter three using a solver and produces the room temperature signal.

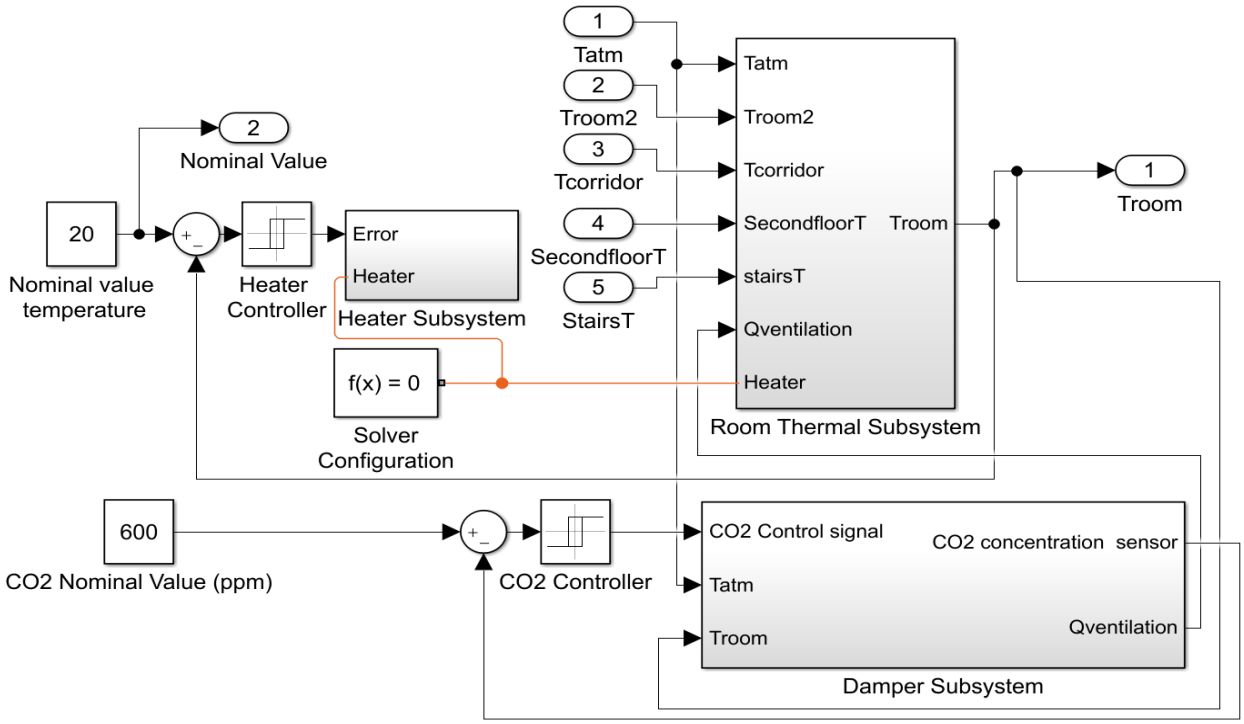


Figure 4.8 Schematic of a typical room implemented in MATLAB/Simulink (room-level).

The Simulink model of the room thermal subsystem includes the heat transfer elements of the model, and it can be observed in Figure 4.9. This model is based on the MathWorks Inc. House Heating System[4] example developed according to section 3.1.4, The Lumped Capacitance Method. After passing through the Ideal Heat Flow Source block using the Thermal Reference, the temperature signals are converted to the heat flow that can further interact with other heat flows. The heat flow is the temperature difference between adjacent zones and the heating load due to the natural ventilation and heater gain. The mass of each wall section is assumed to be condensed in the middle layer showing by *Wall thermal mass* blocks, and the mass of air in each zone is assumed to be condensed in a point showing by the *Air thermal mass* block. The convective and conduction blocks are placed on both sides of each wall.

[4] https://de.mathworks.com/help/physmod/simscape/examples/house-heating-system.html

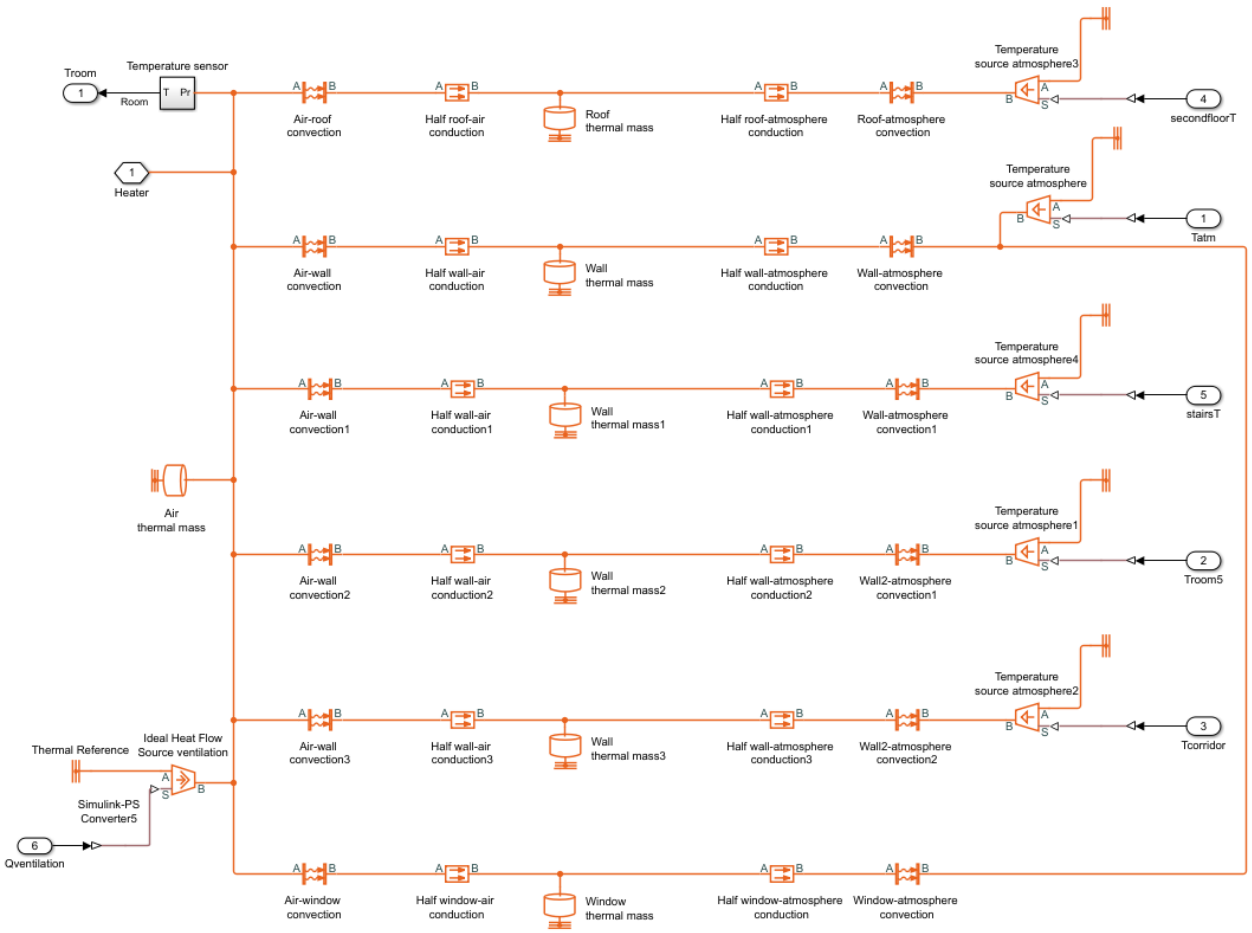


Figure 4.9 Simulink model of the room thermal subsystem.

The model contains a constant power electrical heater which water-based heaters can replace with the assumption of no radiation of the heater to the room's equipment (emissivity 0). Therefore, all the heater energy will be consumed to warm up the room air, and the heater actuator (thermostat) can control the heater's status by simply turning it ON/OFF. The ON/OFF switches are economical for HVAC systems as the zone temperatures change slowly, and precise temperature control is not required. The heater controller is designed based on the hysteresis to reduce ON/OFF cycles. Figure 4.10 shows the heater subsystem designed for each room with capabilities to monitor and track the ON/OFF switch and the heater duty cycle proportional to the heating energy consumption. The duty cycle is defined as the fraction of one period in which a signal or system is active [171]. The equation below may be used to calculate the duty cycle (as a percentage):

$$D = \frac{PW}{T} \times 100\% \tag{4.1}$$

Where,

PW=The pulse active time in one period,

T=The total period of the signal.

For example, a duty cycle of 89.83% means that in a simulation period of one day, for 89.83% of the day, the heater signal is ON (heater warms up the room).

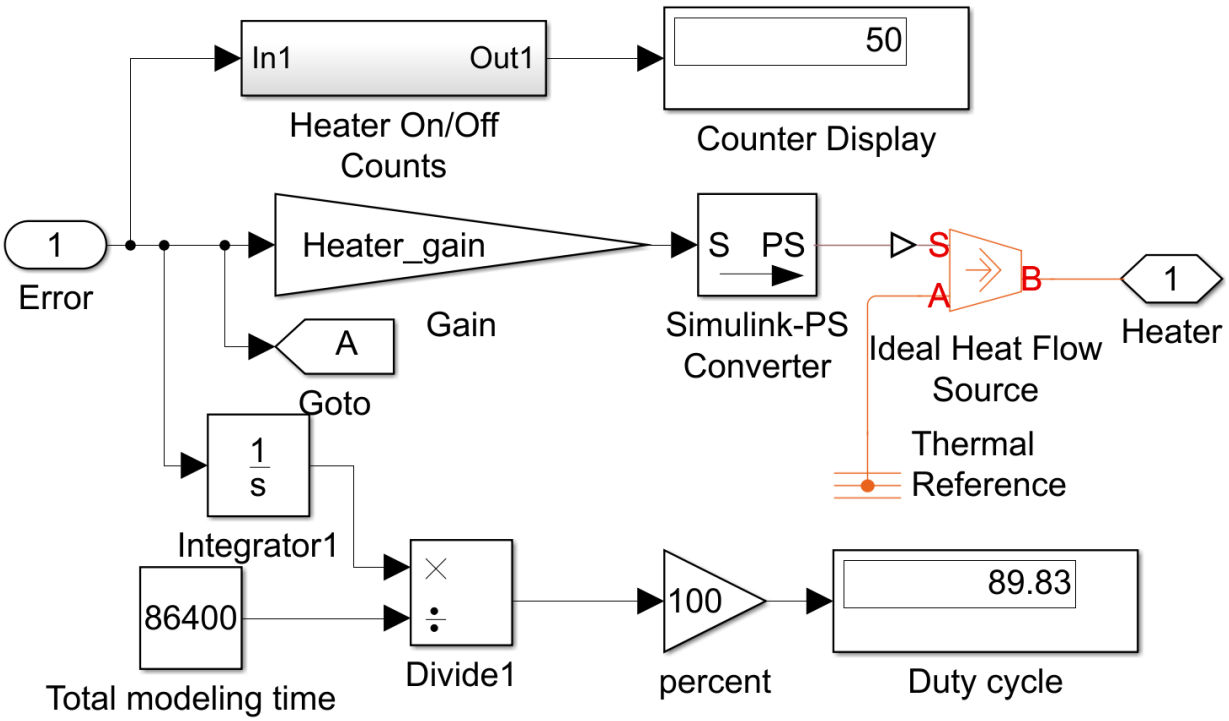


Figure 4.10 Simulink model of the heater subsystem.

The heating cost for each heater can be calculated by inserting a *gain* block with the energy consumption cost per watt after the *Heater_gain* block [172].

Two sensors are modeled and simulated in each zone to get the room's temperature and the CO_2 concentration output signals. One inward opening is also available connected to an actuator in each office room to play the air exchange role. This controllable opening is considered the air damper, and the actuator can control the status of the damper (OPENED/CLOSED) that belongs to the DCV system. The amount of ventilation is affected by outdoor wind and Buoyancy phenomena, and indoor and outdoor temperature is discussed in section 4.1.1.

Assumptions during this thesis for the simplification of derived equations with neglectable effects on the accuracy of the models are:

- The air inside each enclosed space is assumed homogeneous across space (thoroughly mixed), and the temperature distribution is assumed uniform in the hypothetical horizontal plane.
- All the spaces are assumed to have the same pressure, and all the doors are assumed to be in the closed position; hence, the air mass inside spaces remains constant.
- The infiltration effect for the windows and doors in closed positions is zero.
- The temperature sensors are in sheaths.
- The radiation energy absorbed by the building facades is highly dependent on the sun position, which is constantly changing; therefore, a sinusoidal input for the sun irradiation is assumed.
- The heat gains from the people, light bulbs, and other appliances are neglected.
- The radiative heating inside the building spaces is neglected as the effect is minimal.
- The specific heat of the air (C_p) is assumed a constant value of 1.007. In real-world assumptions, C_p has the value of 1.006 at the temperature of 250 K and 1.007 at the temperature of 300 K, so our assumption is accurate within 0.1% error.
- There is no heat exchange between the ground and the ground floor of the building spaces.

- ✓ The model of outdoor air temperature is assumed as a sinusoidal wave during a day, which is 86400 seconds (simulation stop time) where the initial temperature is 7°C (considered at 6:00 a.m.) with fluctuations between 2°C and 12°C.
- ✓ The value of temperature for the second floor and the adjacent stair space were considered 20°C and 13.5°C, respectively.
- ✓ In general, outdoor environment CO_2 concentrations range between 300 ppm and 500 ppm, and indoor CO2 concentrations in office buildings often range between 400 ppm and 900 ppm [13]. This thesis considers the outdoor CO2 concentration a constant value of 400 ppm and the desired indoor CO2 concentration value of 600 ppm with upper and lower fluctuation thresholds with the value of ± 50.
- ✓ The constant value of 13 km/h (3.6 m/s) was considered for the reference wind velocity.
- ✓ The occupancy in each room was simulated as a continuous-time signal for occupancy pattern with a discrete-valued amplitude as a quantized boxcar signal which determines the number of persons. This occupancy pattern can be taken from visiting counting sensors that are modeled and simulated [173]. This pattern was modeled in a matrix by MATLAB code (occupants.mat) and shown in Figure 4.11.

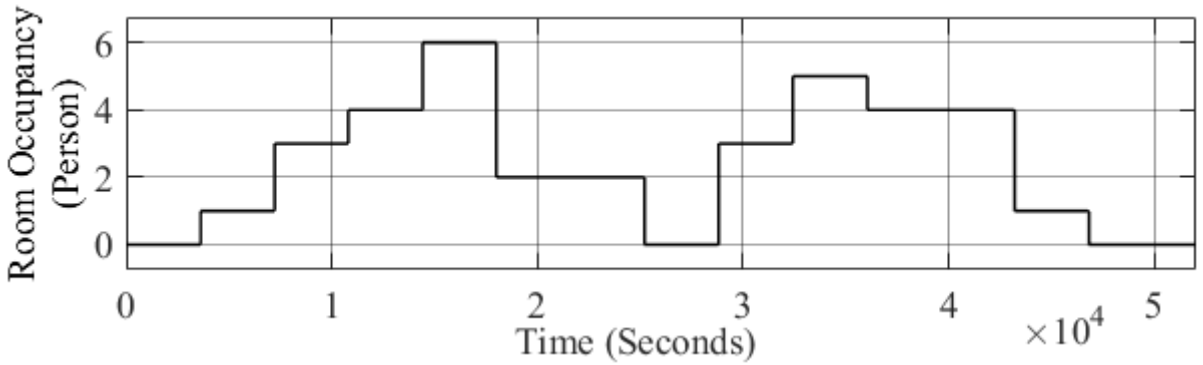


Figure 4.11 Occupancy pattern.

4.1.1. DEMAND-CONTROLLED VENTILATION MODELING

The American Society of Heating, Refrigerating, and Air-conditioning Engineers (ANSI/ASHRAE) standards [144] describes that indoor CO_2 concentrations must be no greater than 600-700 Parts Per Million (ppm) above outdoor CO_2 concentrations. The outdoor CO_2 concentrations in urban environments typically are 400 ppm on average. Therefore, a target level of 1,000 ppm was used to indicate acceptable IAQ. In this condition, a majority of occupants (around 80%) feel satisfaction and around 20% of occupants feel dissatisfaction [13], [174]. The indoor CO_2 concentrations above the hygienic limit (over 1000 ppm) can have adverse effects, e.g., fatigue, increased heart rate, increased noise level in hearing power, striking social behavior, lowered attention, and decreased learning/work performance of occupants. The European standard DIN EN 13779 classifies the indoor air quality in four classes, from IDA1, which represents the high air quality buildings where the indoor CO_2 concentration is less than 400 ppm above the outside CO_2 concentration, to IDA4 with low air quality where the indoor CO_2 concentration is more than 1000 ppm above outside CO_2 concentration [175]. According to the German ad-hoc team, office zones with CO_2 concentration values below 1000 ppm are harmless, whereas CO_2 concentrations above this value are hygienically unacceptable [176]. Therefore, the indoor CO_2 concentration level around 600 ± 50 ppm accounts for a high air quality selected for the acceptable CO_2 concentration thresholds [13].

This section shows the Simulink models of damper and airflow rate subsystem, whereas they have the most critical roles in a DCV system. Figure 4.12 demonstrates the Simulink model of the damper subsystem.

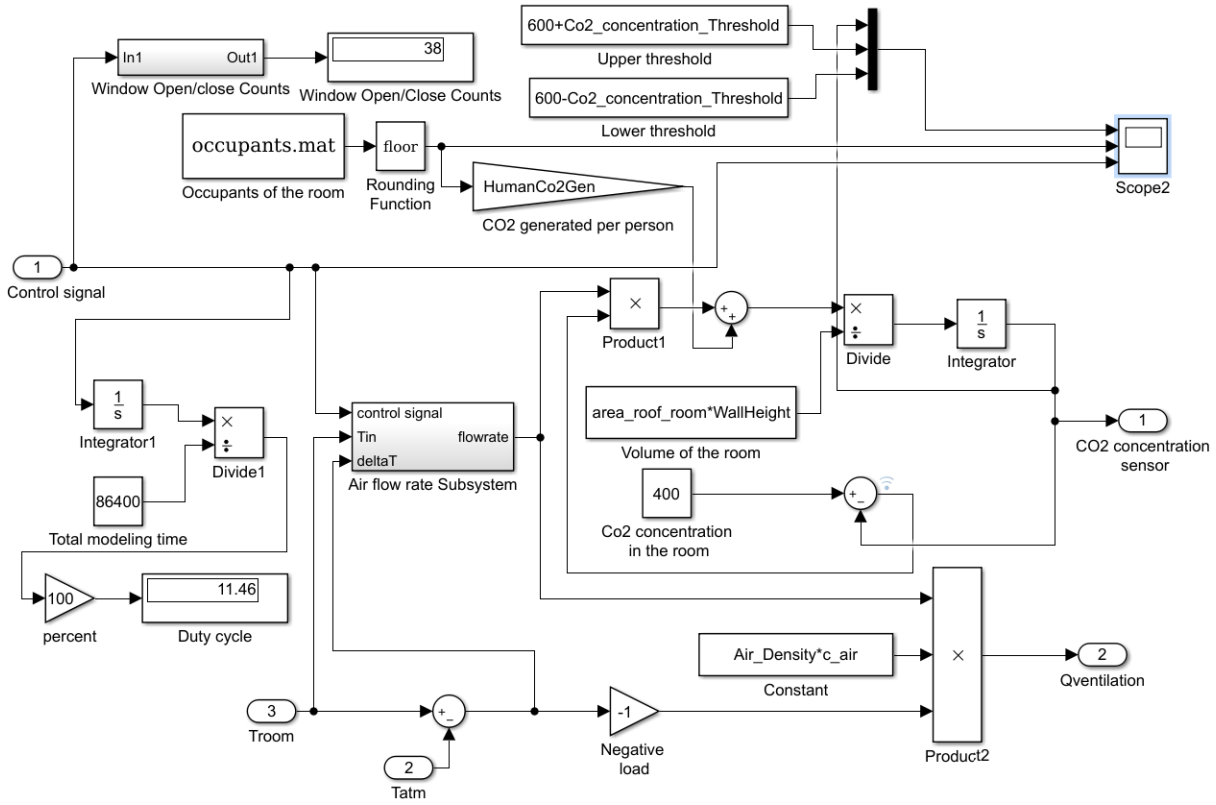


Figure 4.12 Simulink model of the damper subsystem.

Figure 4.13 demonstrates the Simulink model of the airflow rate subsystem.

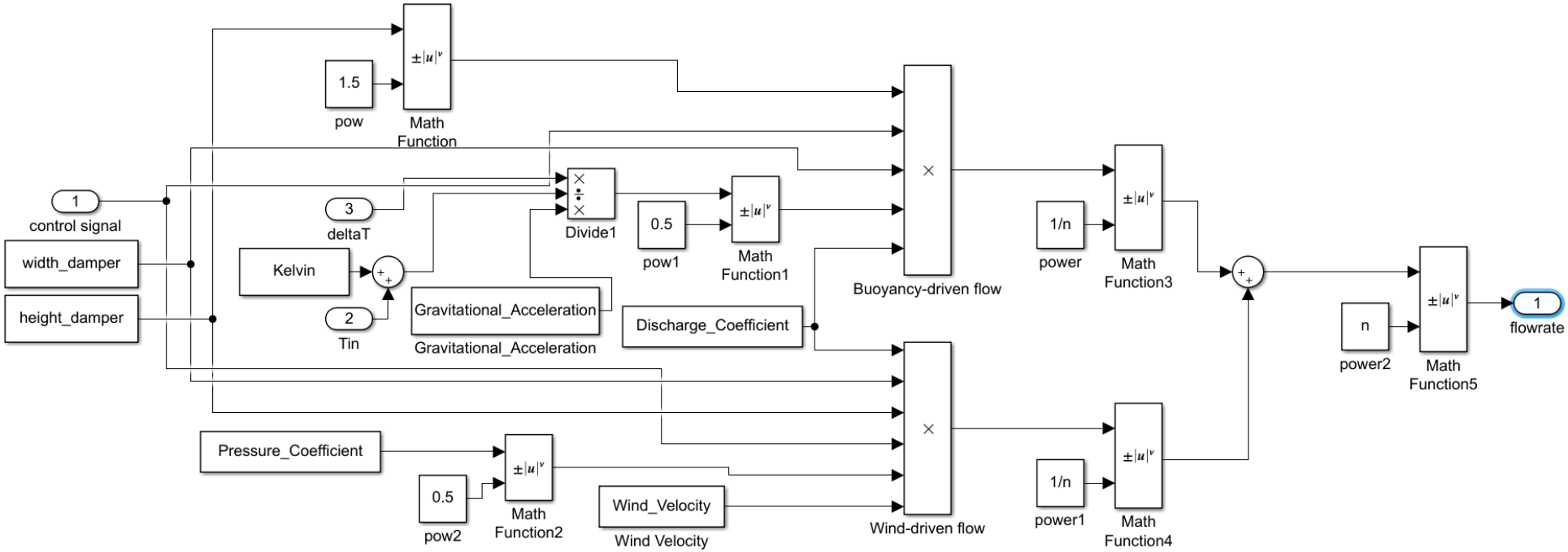


Figure 4.13 Simulink model of airflow rate subsystem.

The values of model parameters such as *wind speed* and *wind angle* for other cases can be updated in the model workspace. The value of dimensionless pressure coefficient C_p can be experimentally calculated based on the changes in wind-induced pressure caused by the influence of surrounding obstructions on the prevailing local wind characteristic considering the wind direction, the building orientation, and geographical properties [140]. Experimental results suggest the constant value of 0.7 for a wholly exposed wall at the angle of 0^o between wind and facade [177]. The wind speed at height H generally depends on the reference wind speed at height 10 meters in an open country and the terrain factors, which depend on sheltering. The constant value of 13 km/h ($3.6\,m/s$) was considered for the reference wind velocity in this investigated model with consideration of the geographical specification of the modeled office building. The variables written in the Simulink model are described in Table 4.1.

Table 4.1 Variables used in Simulink model design.

Parameter	Description	Value	Unit
Air_density	The density of atmospheric air	1.237	kg/m^3
Alt	Altitude	272	m
area_roof_corridor	Corridor area	57.99	m^2
area_roof_room	Room area	37.56	m^2
area_wall_room	Room wall area	17.675	m^2
area_window_room	Window area	11.25	m^2
c_air	Specific heat constant of air	1006	$J/kg.K$
c_roof	Specific heat constant of roof	835	$J/kg.K$
c_wall	Specific heat constant of wall	835	$J/kg.K$
c_window	Specific heat constant of window	840	$J/kg.K$
CO2_Conc_Threshold	CO_2 Threshold value considered for damper	50	ppm
CO2_Ri_Actuator_FI	Damper fault injection activator (linked with slider switch) for room i	0 for OFF, 1 for ON	
CO2_Ri_Actuator_FIV	Damper fault injection value for room i	1 for OPEN	
CO2_Ri_Sensor_FI	CO_2 sensor fault injection activator (linked with slider switch) for room i	0 for OFF, 1 for ON	
CO2_Ri_Sensor_FIV	CO_2 sensor fault injection value for room i	700	ppm
Delay_constant	Constant value used for delay block actuation	1	
Delay_time	Delay time set for fault activation	18000	sec
h_air_roof	Heat transfer coefficient of air adjacent to roof	12	$W/m^2.K$
h_air_wall	Heat transfer coefficient of air adjacent to wall	24	$W/m^2.K$
h_air_window	Heat transfer coefficient of air adjacent to window	25	$W/m^2.K$
h_roof_atm	Heat transfer coefficient of outside air adjacent to roof	38	$W/m^2.K$
h_wall_atm	Heat transfer coefficient of outside air adjacent to wall	34	$W/m^2.K$
h_window_atm	Heat transfer coefficient of outside air adjacent to window	32	$W/m^2.K$
height_damper	Damper height	0.45	m
height_room	Height of room	3	m

HumanCo2Gen	Human CO_2 generation rate	0.0052	L/sec
k_wall	Thermal conductivity of wall	0.038	$W/m.K$
k_roof	Thermal conductivity of roof	0.038	$W/m.K$
pressure	Atmospheric pressure	980008.94	$Pascal$
wall_density	Wall density	1920	kg/m^3
window_density	Window density	2700	kg/m^3
wind_speed	Wind speed	13	KPH
Lroof	Roof thickness	0.2	m
Lwall	Wall thickness	0.2	m
Lwindow	Window thickness (glass area)	0.01	m
Mean_Value_CO2	Desired CO_2 mean value	600	ppm
Mean_Value_Temp	Desired temperature mean value	288	K
secondFloorTemp	Second floor temperature	293	K
stair_temp	Stair temperature	286.5	K
Temp_Ri_Actuator_FI	Thermostat fault injection activator (linked with slider switch) for room i	0 for OFF, 1 for ON	
Temp_Ri_Actuator_FIV	Thermostat fault injection value for room i	1 for ON	
Temp_Ri_Sensor_FI	Temperature sensor fault injection activator (linked with slider switch) for room i	0 for OFF, 1 for ON	
Temp_Ri_Sensor_FIV	Temperature sensor fault injection value for room i	278	K
roofDensity	Roof density	32	kg/m^3
Tnominal	Nominal temperature	293	K
WallHeight	Wall height	3	m
Width_damper	Damper width	0.45	m

4.1.2. DCV AND HEATING SYSTEM MODEL VALIDATION

The simulated model needs to be evaluated to demonstrate the reliability and the correctness of the system response. The simulation serves for verification and validation. Verification is the process of determining that a model implementation accurately represents the developer's conceptual description and specifications. The verification agent is the model/simulation developer. Validation specifies that a model is an accurate display of the real-world system. The validation agent is the functional modeling or simulation expert. ASHRAE [143] and other organizations produced methodologies, tests, and standards for this purpose. This section includes the validation of the heating system model designed in this thesis and the validation of the demand-controlled system behavior and damper response signal that were investigated.

Figure 4.14 shows that the steady-state air temperature of the office rooms was matched to the outside environment temperature when the adjacent stairs and second-floor temperature are the same as the outside temperature, the heating system and the demand-controlled ventilation system are turned off, and the damper openings for all of the rooms are in the closed position.

The model was compared with another reference based on ANSI/ASHRAE standard 62-1989 to validate the modeled demand-controlled ventilation part. Figure 4.15 illustrates the CO_2 concentration variation pattern based on the occupant changes was matched compared to the reference [18].

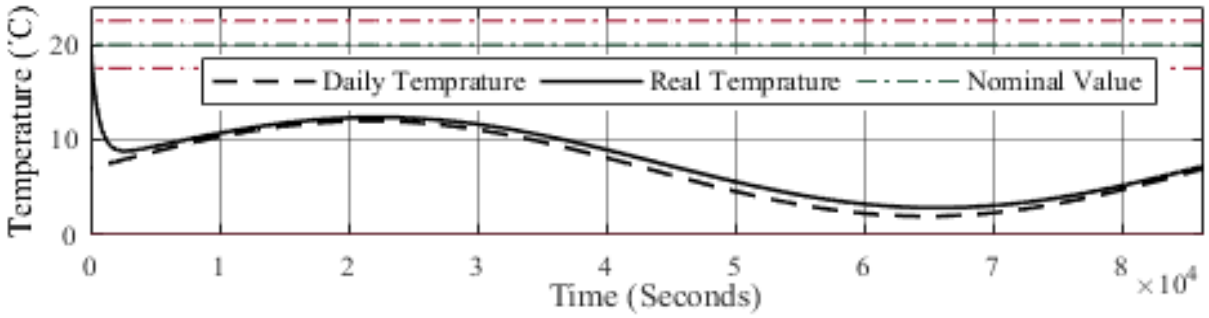


Figure 4.14 Heating system validation.

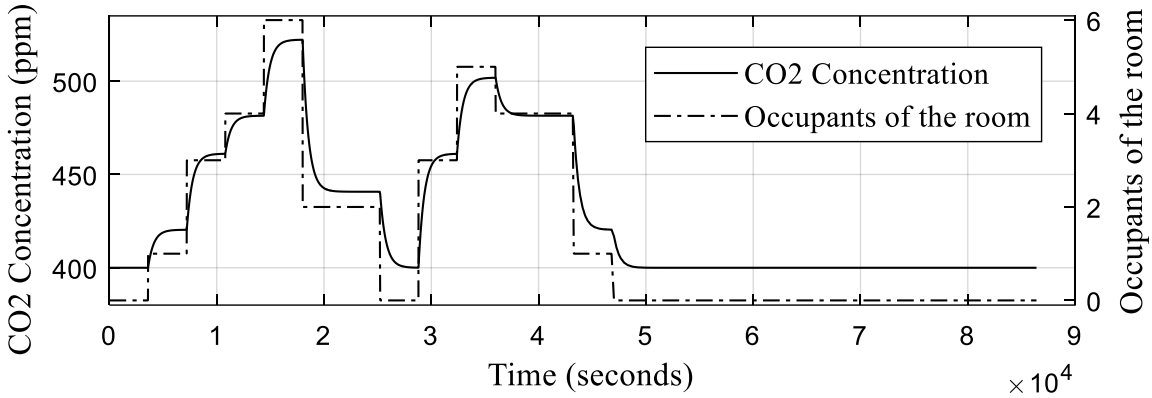


Figure 4.15 Demand-controlled system validation.

4.1.3. DCV AND HEATING MODEL RESULTS

The investigated model of this study shows that it can describe the system response considering input parameters that can be inserted in the model workspace. This model is a scalable model, which means that the user can configure the number of spaces on the same basis, or there is the capability to change input variables, e.g., occupants or outside temperature pattern, heating system output power and settings, dimensions of spaces, or their elements, e.g., windows or damper size, desired amount and limits for indoor CO_2 concentration, air, and building material specifications, and wind speed. An occupant scenario is defined in this model, which in reality can be output from the occupancy sensor measurements. An occupancy sensor detects if occupants are in a zone using the ultrasonic or infrared and can identify the number of people by putting it in a suitable location such as the door to work as a counter. This model can produce the output signals: indoor temperature and CO_2 concentration variation, the duty cycle of the heater, and the frequency of ON/OFF switching for heater and damper for each room. Also, the cost of the heating system for each zone can be calculated by putting a gain block after the heater gain block in the heater subsystem. The author considered example values, but this model is not limited to these values and can be changed for the other studies. The outdoor air temperature was modeled as a sinusoidal wave during a day or 86400 seconds (simulation stop time) where the initial temperature is 7°C (considered 6:00 a.m.), and it fluctuates between 2°C and 12°C. The value of temperature for the second floor and the adjacent stair space were considered 20°C and 13.5°C, respectively. The office room area and height were considered as 37.5 square meters and 3 meters, respectively. Generally, outdoor environment CO_2 concentrations range between 300 ppm and 500 ppm, and indoor CO_2 concentrations in office buildings often range between 400 ppm and 900 ppm [19]. In this study, outdoor CO_2 concentration was considered the constant value of 400 ppm for small towns according to the European standard DIN EN 13779 [175]. The desired indoor CO_2 concentration was considered as the value of 600 ppm with upper and lower fluctuation thresholds of 50 ppm that were

controlled by the embedded CO_2 concentration controller [13]. The model can monitor different system parameters by inserting a scope block, and example simulation results will come in the following text. Figure 4.16 and Figure 4.17 show that the studied model can keep the indoor temperature and the CO_2 concentrations of the office rooms around the setpoint (within the scalable thresholds) considering a minimum heating system output power and a maximum damper opening size. Figure 4.16 includes three subplots that demonstrate indoor CO_2 concentration based on the occupancy in an office room and damper status. For a better view, the figure was cropped for the first 52000 seconds of simulation. It can be observed that more occupants will produce more CO_2 emissions as steeper slopes can perceive it for indoor CO_2 changes.

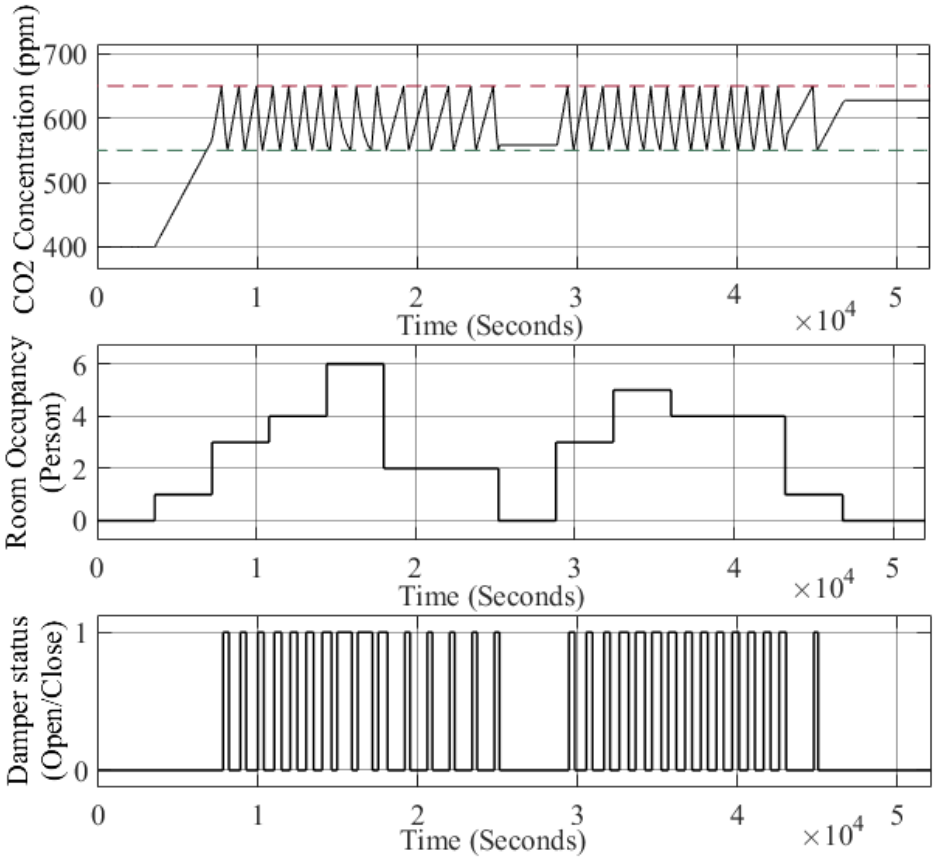


Figure 4.16 Indoor CO_2 concentration based on the occupancy and damper status.

Also, Figure 4.16 shows that the open position time of the damper is more frequent in more populated times. As a result, the damper status could be remained closed in the rest, and it prevents the coming of low-temperature excess air from outside the building (potential energy saving). The frequency of damper switching also depends on the size of the damper openings, meaning a bigger damper size brings more air into the room so that it would be closed more often. The user can change these parameters in the Simulink model to find the optimized one depending on other model parameters, e.g., wind speed or outside temperature pattern. The double y-axes Figure 4.17 shows the variation of room temperature signal of room number 1 based on outside temperature variation, heating system, and damper status. The temperature signals are continuous-time signals with a continuous amplitude referred to as analog signals. Heater and damper are considered to have two possible statuses, ON: 1 or OFF: 0, and open: 1 or close: 0. The room temperature variation was affected by the heat transfer among different rooms and the outside environment. When the inside temperature drops to the lower temperature thresholds, the thermostat switches on the heating system, increasing the inside temperature. Also, it can be observed that the environment temperature increases in the middle of the day (around 18000 to 32000 seconds) can help the heating system to

keep the room temperature within the desired thresholds. As a result, the heating system sometimes could be turned off in the middle of the day. The other aspect is that the fresh air due to the DCV system can be considered the heating load for the heating system, making the temperature drop in the room.

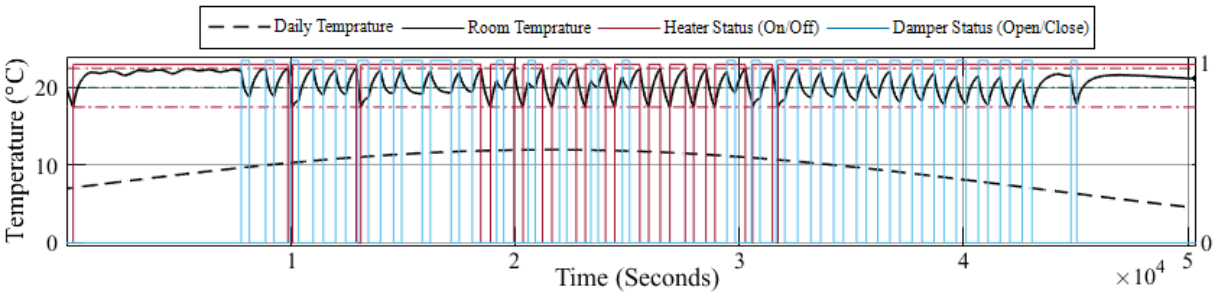


Figure 4.17 The room temperature variation.

4.1.4. EXTENDED MODEL BASED ON WIRELESS COMMUNICATION

The wireless sensor and actuator networks (WSANs) include a composition of embedded devices such as sensors, actuators, and control units distributed in a wireless network without physical connections in a plug-and-play fashion for system monitoring and tracking, collecting information, and control applications [27]. The traditional wireless sensor networks (WSNs) are typically open-loop control systems for measuring the physical world, while WSANs interact with the physical world in a closed-loop system [178]. The sensors measure and transmit the value of parameters required for the control process, and the control (corrective) commands are then forwarded to the actuators to affect the dynamics of the physical plant [179], [180]. The requirements of new technologies based on the WSANs and their advantages, e.g., sensor location flexibility and fewer constraints, easy, and low-cost, and low-disruptive establishment, besides the limitations of the old techniques, e.g., expensive and failure-prone cabled networks and the troubles in components' accessibility and maintenance, especially in retrofits, promote the application of WSANs in HVAC industry [181].

However, integration of the continuous time-driven nature of the DCV and Heating system with the discrete event-driven nature of WSAN creates a challenging development problem in the CPS area [27]. Recent advances in ICT, especially in embedded systems, enable the development of cyber-physical systems that profoundly couple our physical world to the computation world. The term CPS came out in 2006 by Helen Gill from the National Science Foundation in the U.S. refers to the integration of the computation world, performing by embedded computers, with physical processes via network fabric aims at monitoring and control [75]. Physical processes affect computations and vice versa via feedback loops. Therefore, it is not enough to separately consider cyber and physical parts of the system, but also the CPS designer has to be able to understand the intersection of these fields. This joint integration causes specific issues to appear that are not prevalent in general-purpose computing, e.g., the time of task operation. In CPS, it is not desired to perform tasks as quickly as possible. Instead, it is crucial to do the tasks at their proper schedule, considering the pace of the physical environment. Another issue is that in contrast to the sequential and discrete behavior of the cyber world, the physical processes comply with continuous dynamics. An overview of a Cyber-physical System is shown in Figure 4.18.

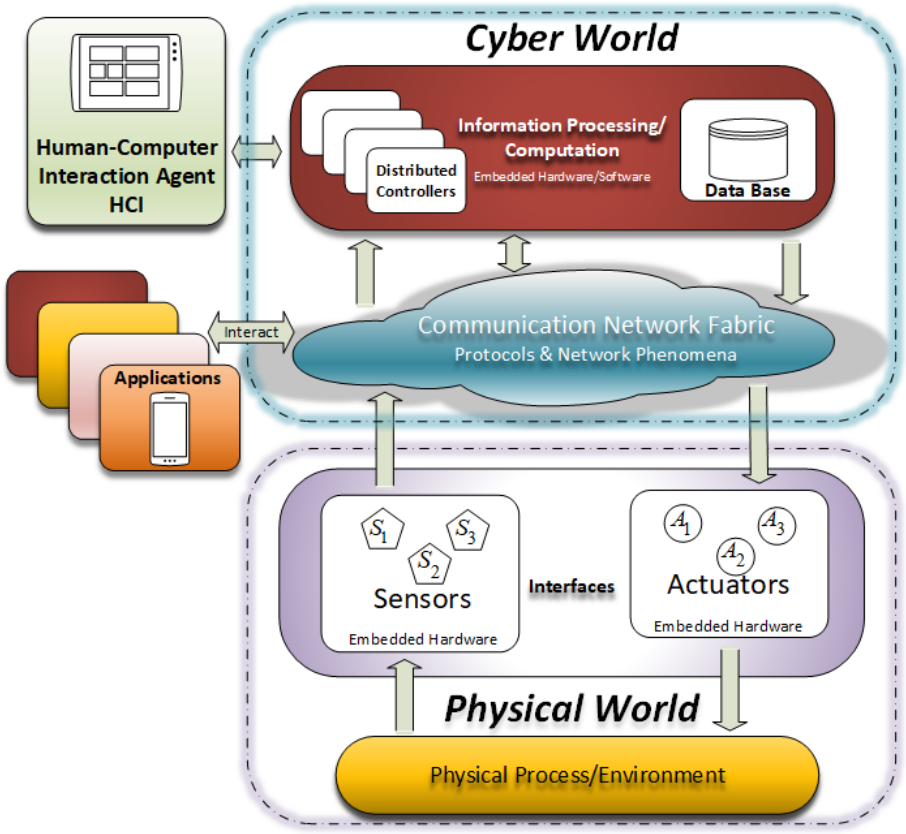


Figure 4.18 An overview of a Cyber-physical System.

The model implemented in this thesis is based on the CPS concepts with embedded processing units orchestrating the nodes of wireless sensors and actuators networks with the physical environment to adaptively control the air quality and temperature of the designed office building. The integration of the continuous time-driven nature of the DCV and Heating system with the discrete event-driven nature of WSAN highlights the mentioned issue of developing the CPS.

The model is associated with ambient sensors, e.g., temperature sensor, occupancy sensor, or air quality sensors, e.g., CO_2 concentration sensors placed inside and outside the building to collect measurement values. Each sensor sends its information to the main controller in the corridor via a communication protocol. The controller exerts the corrective or control commands on the actuators, e.g., damper actuators or heater actuators (thermostats), to interact with the physical plant (the DCV and Heating system). This wireless framework is established in MATLAB/Simulink 2018a using the TrueTime 2.9 Block Library.

4.1.4.1. TRUETIME 2.0

TrueTime was introduced by Cervin et al. in 2010 [182]. TrueTime simulates the real-time behavior of multi-tasking kernels containing controller tasks, network transmissions, and continuous plant dynamics. Tasks are used to simulate both periodic (time-driven) activities and aperiodic (event-driven) activities. Event-driven tasks are executed when network blocks such as the actuators, routers, and coordinator nodes receive a signal; otherwise, they do not execute. Only for the sensor nodes, the authors use the Kernel block to measure the physical attributes periodically.

The scheduling policy is arbitrary and can be given by the user. The support of batteries enables TrueTime to support physical systems operated on a separate power source [183]. TrueTime supports wired network protocols such as Ethernet, CAN, Round Robin, FDMA, TDMA, Switched Ethernet, FlexRay, PROFINET, and NCM, and wireless network protocols such as ZigBee (IEEE 802.15.4), WLAN (802.11b), and NCM_WIRELESS. As the network blocks are event-driven, they

execute when messages enter or leave the network. The transmitted message is put in a buffer at the receiver node. The Acknowledgment (ACK) signal is issued when a receipt of data is recognized back to the sending station (source). Data block must conform to the protocol in use to be recognizable. When the source receives the ACK signal from the destination, it transmits the next block of data. Configuring the network blocks involves specifying the number of general parameters, such as transmission rate, network model, and probability for packet loss. More information and instructions related to TrueTime 2.0 Block Library are available in the guide [182].

This thesis uses TrueTime blocks, e.g., TrueTime Kernel block, TrueTime Receive block, TrueTime Wireless Network block, TrueTime Network block, and TrueTime Battery block.

The Kernel block is an event-driven Simulink S-function that simulates a real embedded system processing unit with embedded codes, instead of mathematical equations, which executes user-defined tasks and interrupt handlers with writing functions of each task in a MATLAB code M-file or C code by the developer and then calling the function in the block to execute the task at given times. An arbitrary number of tasks can be created as the function codes to run in the TrueTime kernel. These tasks may be created for the initialization function or for dynamically executing functions as the simulation progresses, such as the sensor node transmission function from the source to the destination. Tasks are used to simulate both periodic (time-driven) activities and aperiodic (event-driven) activities. Event-driven tasks are executed when network blocks such as the actuators, routers, and coordinator nodes receive a signal; otherwise, they do not execute.

In this model, the kernel block was used in the following different aspects. It has been used as a sensor node to periodically measure the physical attributes and simulate the transmitter of measured values from the plant to the control node wirelessly through the router and coordinator. The input of the block is a sent value and the node's power, whereas the output is the value that should be sent. It has also been used as an actuator node to simulate the command receiver from the control node. The input value is received wirelessly, and the output goes to the plant (heater/damper subsystem). It has been used as a transceiver node (of the controller) to simulate the receive of measured values by the sensor and the transmission of commands to the actuator node. The input is a command value from the controller, which is sent wirelessly to the actuator node through the coordinator, whereas the output is the measured value that is received wirelessly and is provided to the controller as a feedback signal. It has been used as a coordinator of wireless sensors/actuators networks to simulate the getaway between the router of a cluster head in each room and the controller. It has been used as a router of wireless sensors/actuators in each room as a cluster head. The main steps of transmission are as follow. The measured values of the temperature/CO_2 sensor node are sent to the cluster head of a room (i.e., the router) that are then are forwarded by the router wirelessly to the coordinator to complete the feedback loop. Then, the destination controller receives the feedback values and determines the corresponding command, which will be sent to the actuator node through the coordinator. Finally, the actuator node will receive the command value and provide it to the plant (heater/damper subsystem).

The primary purpose of using the TrueTime Battery is to model the node's power consumption in Simulink by connecting it with the feedback of the TrueTime kernel block. The kernel node stops working in the simulation when the output of its TrueTime battery is zero. The input of the block is the consumed power of the node, whereas there is information about this power that can be obtained from the output of the wireless network. The TrueTime Wireless Network is used to define several factors such as the network protocol, the number of nodes in each cluster, frame size, and transmit power.

The Kernel Block has the following parameters that the user can set. "*Name of the Init function*" describes the name of the desired M-file or a MEX file with the initialization code, "*Init function argument*" shows the optional arguments that are to be passed for a successful initialization, "*Number of analog inputs and outputs*," "Node numbers" defines the node number in the network.

Likewise, the Wireless Network Block has the following parameters. "*Network Type*" describes the type of network and specifies the desired MAC protocol for that network, "*Network Number*" defines the unique id number of the network. "*Number of nodes*" describes the number of nodes in that network. "*Data Rate and Minimum Frame size*" describes the bandwidth and the minimum size of each packet. The acknowledgment timeout specifies the maximum time for the acknowledgment. "*Show Power consumption output port*" shows the power consumed by the network, and it can be monitored.

4.1.4.2. COMMUNICATION PROTOCOL

In building HVAC technologies, heterogeneous devices exist in a wireless network typically. The communication protocol helps the engineers to connect these devices from various brands for data exchange and communications. The most prevalent communication protocols were studied and compared in Table 4.2 against various aspects of compatibility with TrueTime, functionality, advantages, and disadvantages. The comparison declares the ZigBee protocol's superiority in WSANs' HVAC application because of its high-reliability, low-cost, and energy-saving characteristics.

Table 4.2 Wireless Communication Protocols [27].

	ZigBee	***Bluetooth***	***Wi-Fi***	***WirelessHART***	***MiWi***	***Z-Wave***
Standard/Protocol	IEEE 802.15.4, ZigBee	IEEE 802.15.1	IEEE 802.11 (a-b-n-g)	IEEE 802.15.4	IEEE 802.15.4	Z-Wave
Network Topology	Star, Peer-to-Peer, and Mesh	Star, Peer-to-Peer	Star, Peer-to-Peer	Star, Peer-to-Peer, and Mesh	Star, Peer-to-Peer	Mesh
Power Consumption	Very Low	Medium	High	low	Low	Low power
Battery life (Days)	100 to +1000	1 to 10	0.5 to 5	Depends on Battery Specifications	Depends on Battery Specifications	Depends on Battery Specifications
Range (meters)	10–300	10	10–100	100	20–50	30 -100
Market Adoption	High	High	Extremely high	High	Medium	Medium
Network size (nodes)	64000	8	2007	100	1024	232
Application Areas	Demand Response, remote control and automation in residential and commercial buildings	Wireless connectivity between personal devices such as headphones, medical, sport & fitness, mobile phones or laptops	Wireless LAN connectivity, broadband Internet access	Industrial Control, building control the sensory data conveying temperature, pressure or speed	Industrial monitoring and control, home and building automation, remote control lighting control and automated meter reading	Remote control lighting and automation, control, security systems, thermostats, windows, locks, swimming pools
Advantages	Endurance, Low Cost, Low Power consumption, several application profiles (home automation, smart energy), and topology flexibility	Easiness, Speed, and flexibility	Speed and flexibility	Communication Security, reliability and Environment with wired HART infrastructure	Flexibility, cost-effective platform	Controllers and slaves network, flexible network configuration

ZigBee Alliance extracted the IEEE 802.15.4-based ZigBee standard, a high-level communication protocol for better reliability and cost-efficiency of nodes in wireless networks. In 2008, the BACnet

protocol in HVAC and building automation was integrated with the ZigBee because of its specifications such as high reliability, high security, low power consumption, low cost, low complexity, and easy implementation [184]. The embedded nodes in the network can operate on a battery [185]. The ZigBee devices (nodes) are categorized based on their functionalities into three types [186].

1. *Network coordinator:* It is a unique Full-Function Device (FFD) that chooses critical network configuration and initialization parameters. It is also responsible for storing network information and connecting other networks such as a bridge. In this study, the network coordinator is located in the corridor.
2. *Router:* It is an FFD node responsible for the data routing functionality by acting as an intermediate device to link different devices such as sensors and actuators of the network and forward messages between remote devices across multi-hop paths. A router communicates with other routers located in different zones and the end node devices. In this research, every room has been assumed as a cluster. Therefore, the routers play the role of the cluster head.
3. *End devices:* A Reduced-Function Device (RFD) with limited functionality to communicate with its parent node, possibly the network coordinator or a router. The end nodes are generally in the sleep mode as long as they are not sending or receiving information or executing tasks because of their limited functionalities to profit from a prolonged power source (e.g., battery) life. The end devices in this study are three sensors of CO_2 concentration, temperature, and occupancy and two actuators of the damper and thermostat area [27].

4.1.4.3. NETWORK TOPOLOGY

The network topologies enable robust communication over wireless networks. A ZigBee protocol's Personal Area Network (PAN) supports star, peer-to-peer (mesh), and cluster-tree topologies. Among the mentioned topologies, peer-to-peer (mesh) provides higher reliability using different routes enabled by routing algorithms. One convincing argument for using clustering technology is to reduce the data packets during data aggregation that decreases the cost of communications and energy consumption, therefore, a prolonged lifetime of WSAN. Hybrid topologies are better solutions that benefit from the superiority of a composition of two or more topologies. Therefore, this thesis selects the cluster-tree-mesh topology based on the building architecture that supports wireless and battery-powered nodes (devices) with minimum routing effort shown in Figure 4.19 [187].

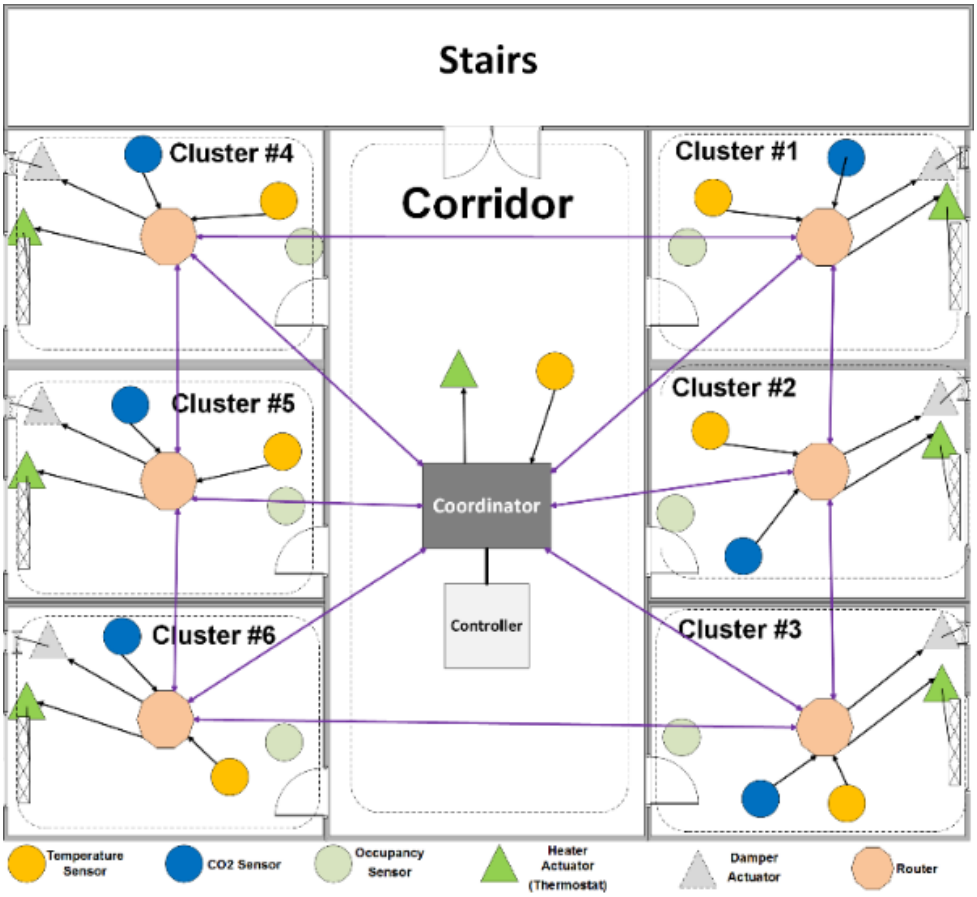


Figure 4.19 Network Topology based on Building Architecture [27].

In this topology, the sensors nodes located in each zone send the measured values, e.g., temperature, occupancy, or CO_2 concentration, to the cluster head of the zone that is a router. The router receives the values and forwards them to the controller via the coordinator. After calculations, the controller processes the received values, determines the commands, and exerts them on the actuators, e.g., heater and damper actuator, which is finally applied to the plant. The Simulink model presents all the components by two parameters of ***Network Number*** and ***Node Number*** as the elements in a two-column and one-row matrix that the first element shows the network number and the second element represents the node number, e.g., in [1 2] or 1:2, the network number is 1 and the node number is 2. Figure 4.20 shows the corridor Simulink model and wireless communication blocks.

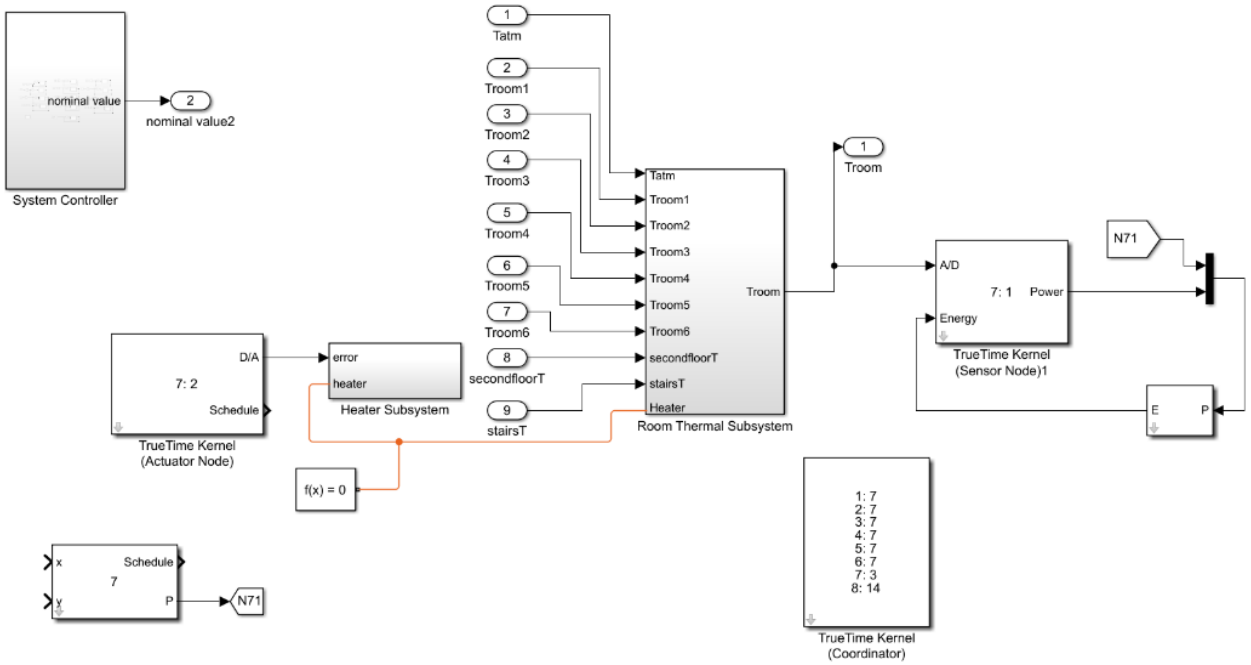


Figure 4.20 Corridor Simulink Model.

4.1.4.4. EXTENDED MODEL VALIDATION

The wireless architecture proposed in this thesis is validated by mapping properties such as the temperature signals in the extended model based on the wireless network compared to the system output of the wired network [172]. The simulated results at the same operating conditions and assumptions in wireless and wired architectures describe an excellent match when mapping the validation properties such as the temperature signal that can be seen in Figure 4.21[172].

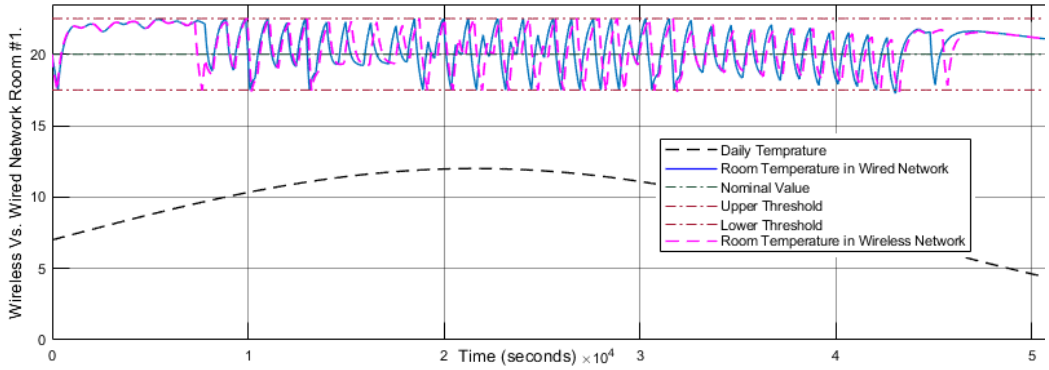


Figure 4.21 System Model based on Wireless Network Validation [27].

4.2. MODELING AND SIMULATION OF THE MODULAR COMPOSABILITY

State of the art in composable modeling today stems from the advancement of the Modeling and simulation sciences and the rapid growth in simulations tools such as MATLAB/Simulink or LabVIEW. Currently, users only identify their model requirements. Then the developer constructs the model usually with an iterative process with functionally being added at each iteration, and later the model is delivered with an instruction manual on how to use and maintain [188]. As the complexity of systems is on the rise, the main difficulties and problems in system analysis that affect model composability stem from these aspects [166].

- Integration of models in different disciplines and fields
- The abstraction of the model descriptions
- Restrictions and constraints of the development tools
- Level of composability of the model components

This thesis investigates a DCV and heating system model for an automatic, scalable, and composable generation. The aim is to introduce a method in MATLAB/Simulink to generate multiple models from a single one. The novelty of this work resides in the ability to generate a new model using pre-defined Simulink libraries made from the original model blocks that represent the main components of the original model. Furthermore, this project introduces a GUI that automatically generates real-time artificial faults for fault injections and communicates with a remote database for storage and diagnostics. The implementation part of composability modeling requires some pre-requisites. For example, an indexing structure is vital for developing the building architecture as the composable DCV and Heating system model foundation. Therefore, to generate the composable model, which consists of *N* number of rooms in *K* number of floors, an indexing structure for the building with the depicted layout below, in Table 4.3, is defined [163].

Floor 1		**Floor 2**		**…**	**Floor K**	
1	(N ÷ 2) +1	(N + 1)	(N+(N ÷2)) +1	…	(N * (K-1)) +1	(N * (K -1)) + (N ÷2) +1
2	(N ÷ 2) +2	(N + 2)	(N+(N÷2)) +2	…	(N * (K -1)) +2	(N * (K -1)) + (N ÷2) +2
3	(N ÷ 2) +3	(N + 3)	(N+(N÷2)) +3	…	(N *(K -1)) +3	(N * (K -1)) + (N ÷2) +3
...	...	...	…	…	...	...
N÷2	N	N + (N ÷ 2)	(N * 2)	…	(N * (K -1)) + (N ÷2)	(N * K)
N: ***The total number of rooms per floor*****, K:** ***The total number of floors in the model***						

Table 4.3 Rooms indexing structure [163].

A novel part of this thesis is based on the idea and algorithm developed to create a modular, composable architecture in MATLAB/Simulink, partly published in [163]. The algorithm is written in the MATLAB environment and *.m* format. The algorithm shows the logic beyond the connection patterns of different Simulink blocks to create the composable version of the base model and highlights the connection between MATLAB and Simulink. This algorithm is also helpful in developing diagnostic modules. Different types of faults can be injected to test the diagnostic technique developed in this thesis. The user needs to insert and set the input values into the designed GUI and run the simulation. Then, the system with the desired composable architecture is spontaneously created, and the required fault blocks and diagnostic modules are built automatically. In the end, the fault injection blocks and diagnostic modules will be activated.

The inter-relationship identification is the first pace of composable model generation. Therefore, the DCV and Heating system developed in this thesis was studied in detail to extract the logic of components and inter-connections of adjacent zones. This logic is highly dependent on the thermodynamic interaction among adjacent zones. The hierarchical model proposed in Figure 4.22 describes the hierarchical order of this configuration based on the subsystems available in the high-level model and low-level model of abstraction, which is beneficial to grasp a complete understanding of the model. The model subsystems, including Heater, Thermal, Damper, Actuators,

and Sensors Subsystems, are developed and connected first in Simulink using Simscape library blocks. Later, the MATLAB code of the composable version of the Simulink model is generated.

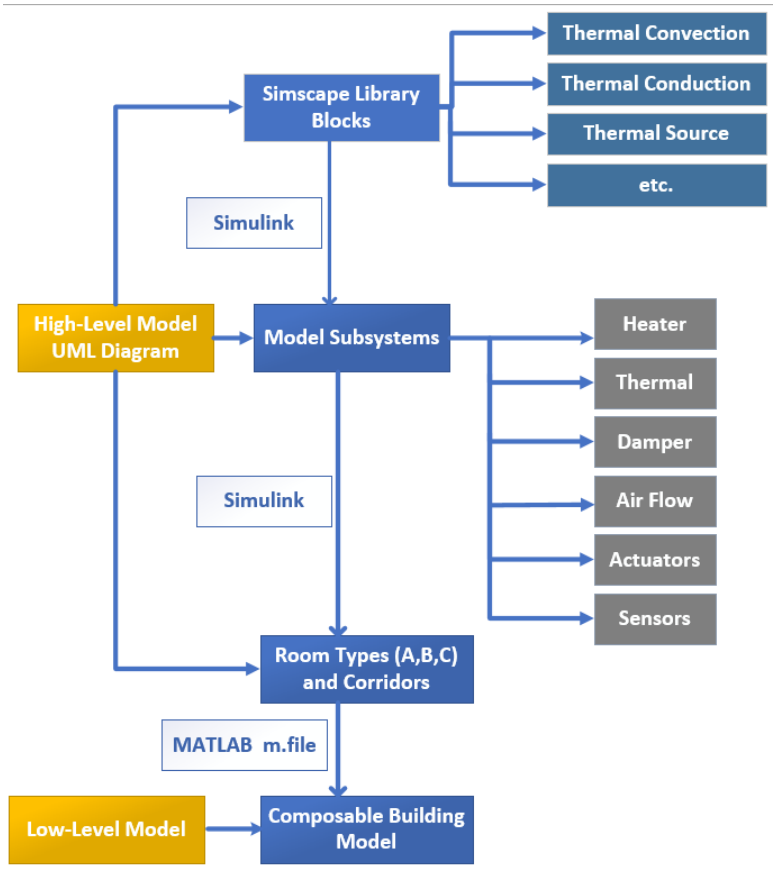


Figure 4.22 Hierarchical View of Composable Model [163].

4.2.1. The Composable Model Constraints

The composable architecture provided in this thesis assumed some constraints to maintain integrity. The physical and mathematical analyses of the system define these limitations. The rooms in the provided model can be placed beside each other, generally considering their types concluded from the analysis stage. Figure 4.23 explicates the main room types of A, B, and C based on their

vicinity and the adjacent environments/zones, e.g., outside environment, stairway, corridor, other room, ground, and roof.

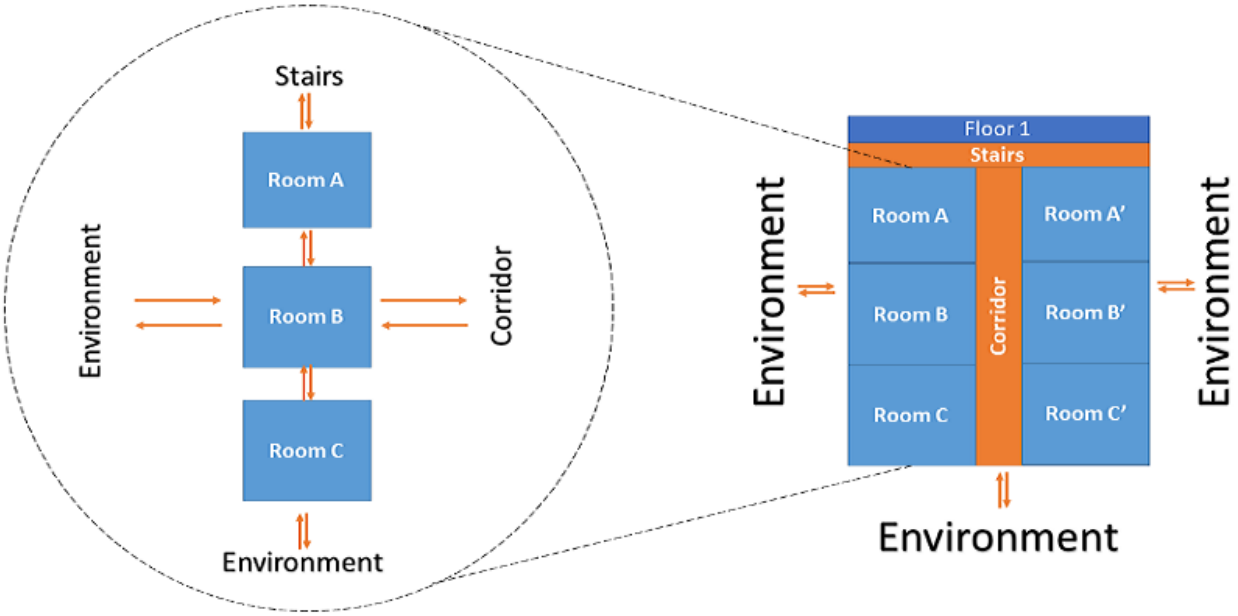


Figure 4.23 Interactions among Rooms [163].

The Simulink model in its basic version is composed of three types of rooms (with variations in terms of the thermodynamic interactions considering the design symmetry): A (Room 1), B (Room 2), and C (Room 3), as seen in Figure 4.24. Also, each floor has a corridor block and a stairway on each floor, represented in the model by a scalable block.

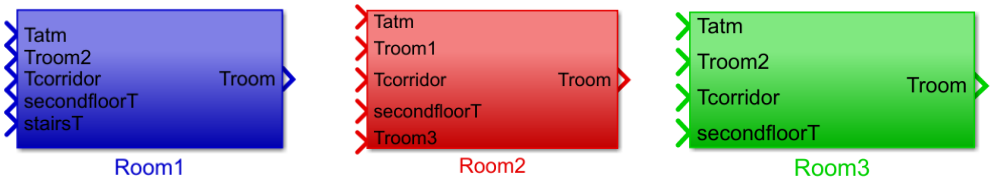


Figure 4.24 The main Simulink blocks representing the main types of rooms.

The proposed algorithm in this study can arrange the rooms to create a composable model, and the rooms, depending on the position that must be placed in the composable model, are duplicated and placed automatically [163]. Figure 4.25 clarifies an example with three room types in three colors of blue, red, and green with unique physical and mathematical specifications, e.g., heat and mass transfer, are placed in two horizontal and vertical directions.

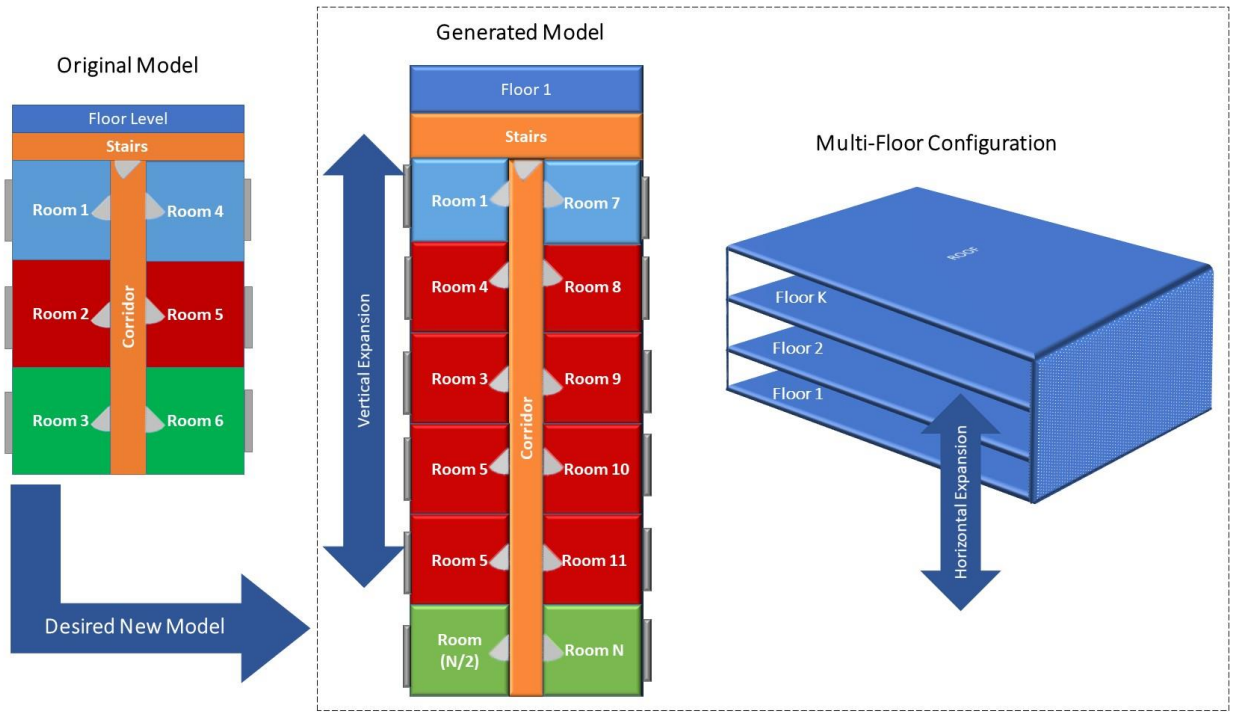


Figure 4.25 Model Scalability [163].

The building architecture can be expanded or contracted to N number of rooms and K number of floors using the designed GUI as an input for the composable model algorithm. The mentioned Room Indexing Structure describes the reference of the rooms' placement. The principal codes that are frequently used in this thesis are shown here:

Table 4.4 Most frequent used MATLAB/Simulink commands

Number	Command	Description
1	*new_system(obj)*	Create blank Simulink model
2	*open_system(obj)*	Open the Simulink model
3	*h = add_block(source,dest)*	Adds a copy of the block source from a library or model to the specified destination model
4	*function [y1,,yN] = myfun(x1,...,xM)*	Declare function name, inputs, and outputs
5	*for index = values statement end*	For loop to repeat a specified number of times
6	*h = add_line(sys, out,in,'autorouting',autoOption)*	Add line to Simulink model between model components
7	*chr = int2str(N)*	Convert integers to characters
8	*s = strcat(s1,,sN)*	Concatenate strings horizontally

9	if, else, elseif, end	Execute statements if the condition is true
10	*Set_param(Object,ParameterName,Value, ...ParameterNameN,ValueN)*	Set system and block parameter values
11	*assignin(ws,var,val)*	Assign value to variable in specified workspace
12	*ParamValue = get_param(Object, Parameter)*	Get parameter names and values
13	*global var1 ... varN*	Declare variables as global
14	*T = array2table(A)*	Convert homogeneous array to a table
15	*[output1,,outputN] = eval(expression)*	Execute MATLAB expr ession in text
16	*sqlwrite(conn,tablename,data,Name,Value)*	Insert MATLAB data into database table and creates database tables
17	*fastinsert(conn,tablename,colnames,data)*	Add MATLAB data to database tables
18	*disp(X)*	Display value of variable
19	*pause(n)*	Stop MATLAB executi on temporarily in seconds
20	*get_param('sys', 'SimulationStatus')*	check the status of a simulation
21	*set_param('vdp','SimulationCommand','sta rt')*	Start a simulation
22	*set_param('vdp','SimulationCommand','pa use')*	Pause a simulation
23	*set_param('vdp','SimulationCommand','co ntinue')*	Continue a simulation
24	*set_param('vdp','SimulationCommand','sto p')*	stop simulation
25	*set_param('vdp','SimulationCommand','up date')*	update the changed workspace variables dynamically while a simulation is running

4.2.2. UNIFIED MODELING LANGUAGE

Unified Modeling Language (UML) is a graphical modeling language to achieve the insight and describe the attributes, operations, relationships, and associations. The UML includes many diagram types, e.g., class diagram. [189]. The class diagram denotes the interaction among the system components.

This thesis uses the class diagram, a type of UML diagram, to extract information, e.g., attributes, operations, relationships, and associations of the DCV and Heating System, as a requirement of composable modeling. Figure 4.26 explicates the UML diagram of the designed model in this thesis that the main parts of the class diagram include [163]:

a) Building

The most upper element of the UML is that the building includes the main attributes of the composable architecture such as the type, e.g., type A, B, or C, and the number of rooms and floors, and the building HVAC systems.

b) DCV and Heating System

This class includes the attributes of the used HVAC system and is connected to the Process Interface and Ventilation and Heating Control classes. The relationship between temperature sensor class and thermal class is of the association. However, Figure 4.26 separately shows a specific UML diagram of the DCV and Heating system.

c) Process Interface

The processing units of the model, e.g., sensors and actuators, are described in this section.

- **Sensors:**

  The sensor class inherits to CO_2 sensor and temperature sensor.

- **Actuators:**

  The actuator class inherits to heater actuator (thermostat) and damper actuator.

d) Ventilation and Heating Control System

The main compositional classes are heater, thermal, and damper.

- **Heater**

  This class is associated with the heat generated in a zone.

- **Thermal**

  The thermal class collects the temperature values from the outside environment and adjacent zones through its input ports and solves all the Ordinary Differential Equations (ODEs) using Simulink blocks to calculate the temperature value of each particular zone.

- **Damper**

  Damper class calculates the airflow through damper to each zone that is affected by the CO_2 measurements.

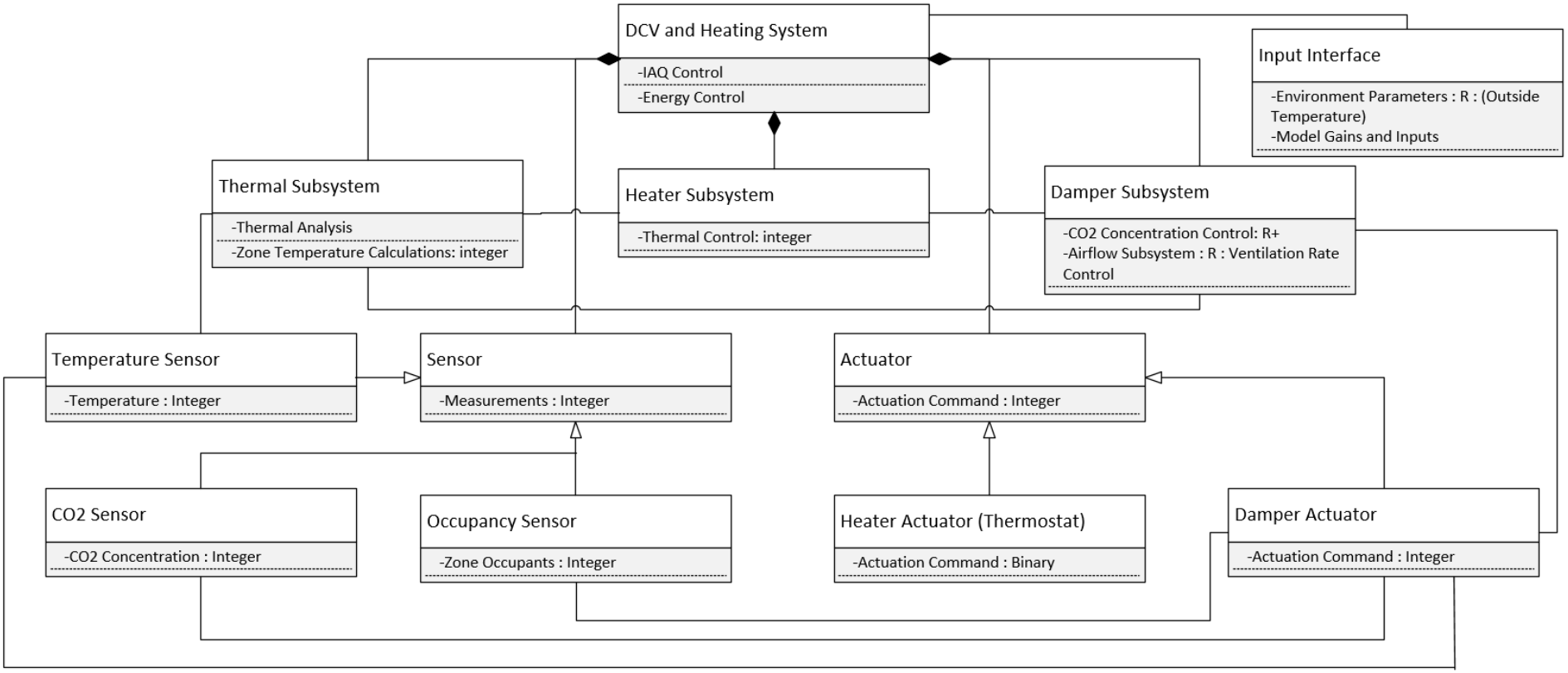


Figure 4.26 The UML Class Diagram of the DCV and Heating System.

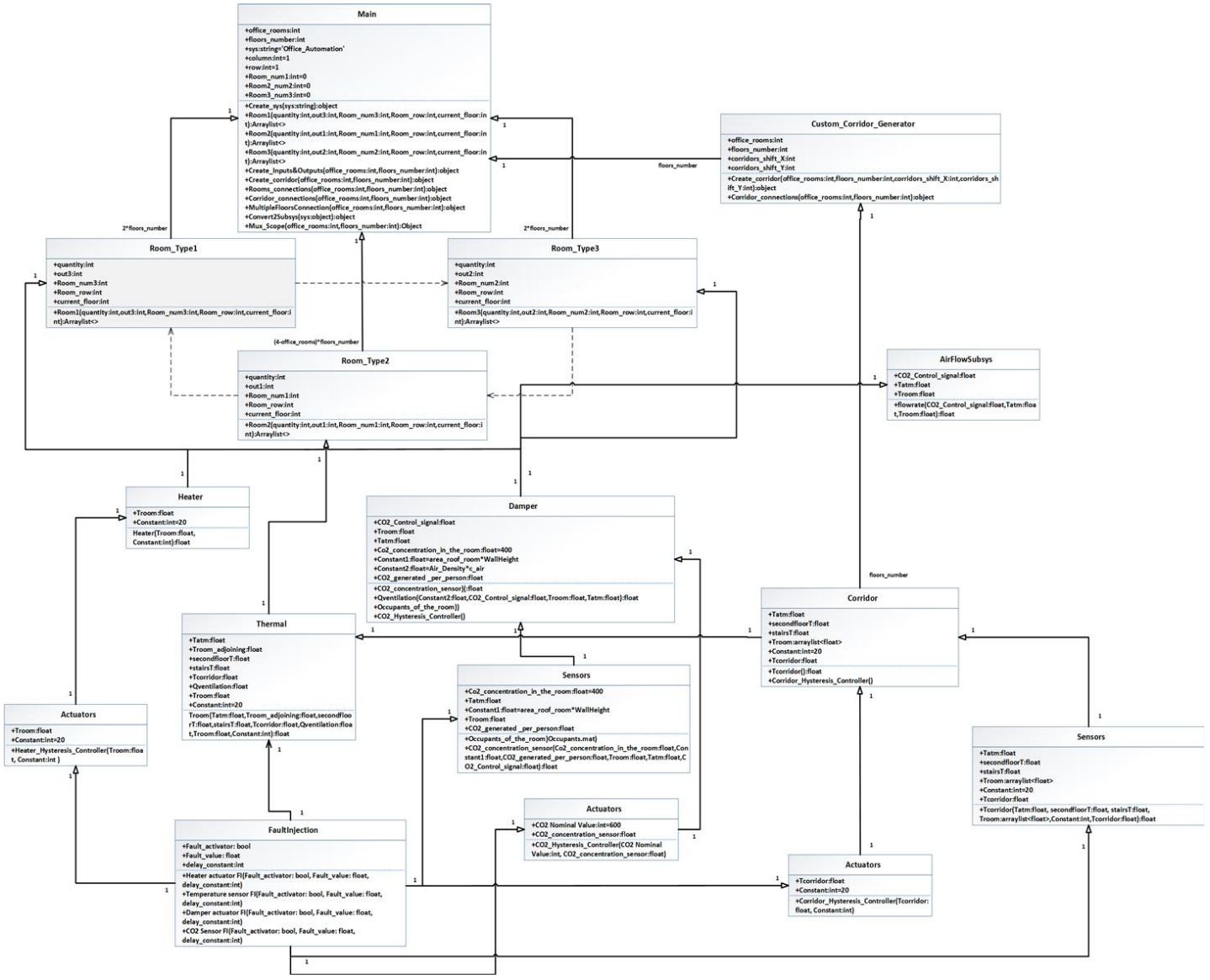


Figure 4.27 The UML Class Diagram of Designed Composable Model [163].

4.2.3. MODULAR COMPOSABLE MODEL IN MATLAB/SIMULINK

This section is a step-by-step delineation of modular composability modeling implementation in MATLAB/Simulink [163]. Figure 4.28 draws the ability of the developed algorithm to create customized building architecture by extending the basic model in horizontal and vertical directions.

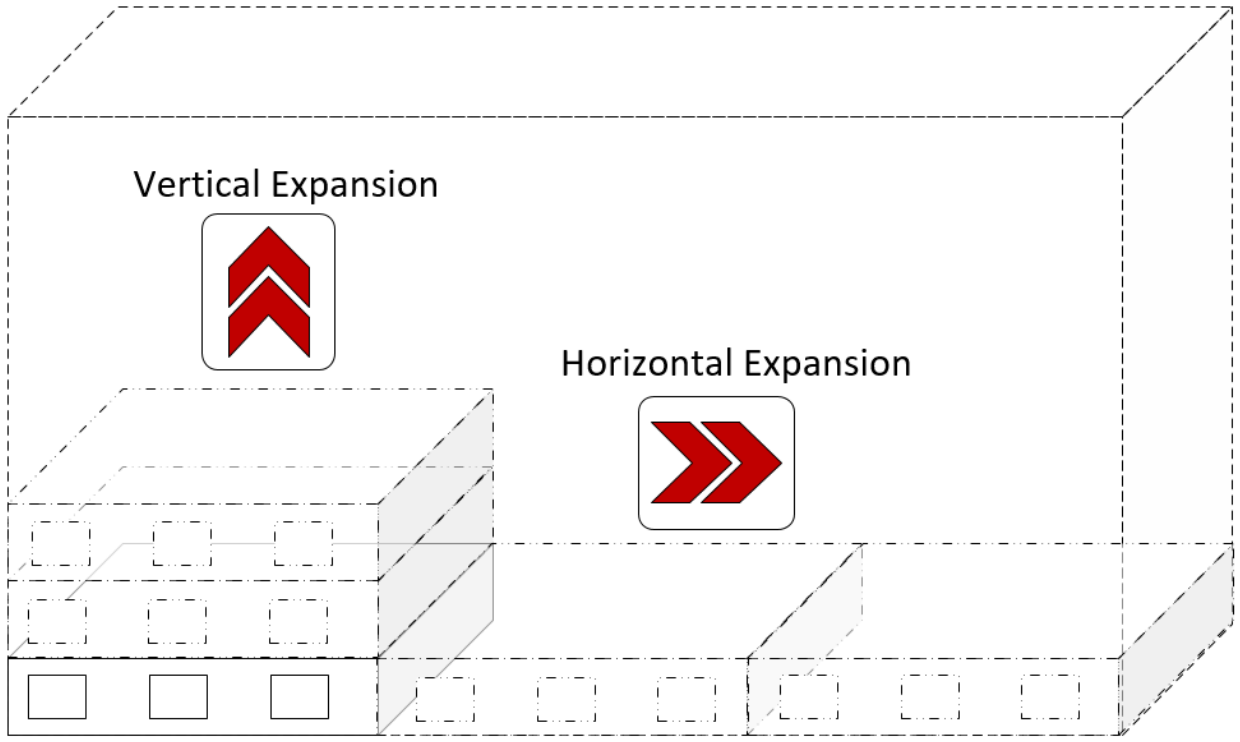


Figure 4.28 Composable model expansion in vertical and horizontal directions

The code that was written to generate a new configuration of the office ventilation model consists of 10 main sections, these sections are written in MATLAB language, and they serve as the building block of the new model. Check Figure 4.29 for the main program sections. Each section is mentioned and described briefly below.

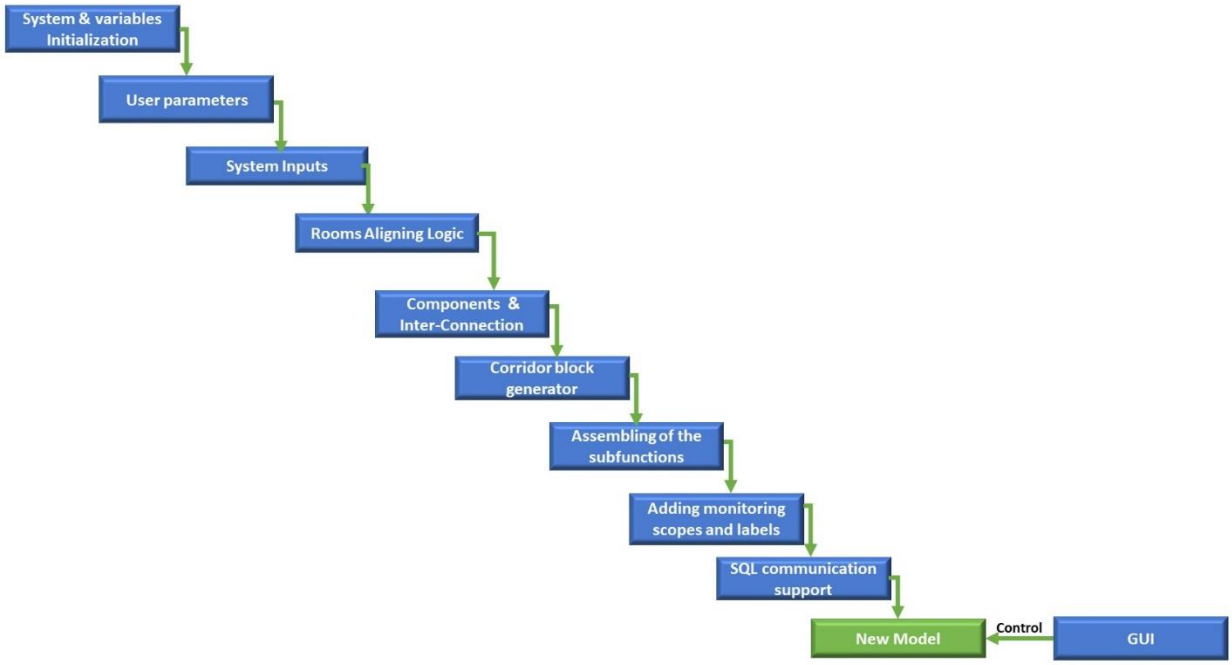


Figure 4.29 Waterfall view of the programming technique.

Graphical User Interface (GUI) provides a point-and-click software application. It eliminates the need to use programming languages to execute an application [22]. MATLAB offers the ability to design and run a GUI for various ranges of applications. The GUI tool can be accessed by typing *>GUIDE* in the MATLAB command window. This tool will allow users to drag and drop elements to the user's own GUI window, or the user can add each component programmatically using the corresponding command for each desired element in the GUI, i.e., for a pushbutton, the user has to use the following command in his code and the button will be created each time the user compiles his script [23].

The design steps to create the composable model are listed below:

- ✓ **System & Variables Initialization:** The model and the variables (i.e. index variables) are initialized in this section.
- ✓ **User Parameters:** The user can enter the desired input values, e.g., the number of floors and the number of rooms on each floor.
- ✓ **System Inputs:** In this section, the system inputs are imported from a predefined library.

```
%% B-User defined parameters
% min is 6 and must be an even number
office_rooms=6;
floors_number=1;
SQL_Support=1;
```

These inputs are the outside temperature block (sine wave Simulink block), the upper floor temperature constant that acts as the roof temperature in the model, and finally, a stairway temperature constant connected to the zones through an input port labeled as Stairs. The code will import these blocks and place them on the created Simulink page using the *add block ()* command.

- ✓ **Rooms Alignments Logic:** The established composable model in this study is scalable, and the expansion in horizontal and vertical directions includes the thermodynamics principles of building zones and environment (surrounding).

- ✓ **Main Components & Inter-Connections:** After creating the desired building structure, the main model components are added to the model in this section, and the inter-connections among them are established. Also, an output index is assigned to each component's output port to capture data for monitoring and diagnostics.
- ✓ **Corridor Block Generator:** This section of the code is responsible for a customized corridor block for the composable model which suits the new model. The new customized corridor block is based on this block's rescaled or expanded mathematical and thermophysical interactions among different zones. The design of this part was crucial as the scalability of the corridor block should be maintained. After examining the original model corridor block, the Simscape model of the thermodynamic interaction was thoroughly explored, and later it was determined which part of the model should be altered when scaling the model. After successfully generating the new corridor block, it was validated against the original model corridor block, and it was found that it produces the same behavior.
- ✓ **Assembling the Sub-Functions:** This code section deals with arranging the blocks in the simulation environment and layouts.
- ✓ **Adding Monitoring Scopes and Labels:** This study was another challenge in preparing suitable and flexible scope blocks that automatically adapt to the new system design. The designed model must show the system's behavior in output signals from sensors and actuators for every design. Therefore, this thesis solved this challenge based on the desired system architecture and the suitable output figures available to the user for the aims of monitoring and diagnostics. For this reason, each output port and signal gets a label which later is categorized. A scope represents the data from a single component in the model. For example, each room in the office will have five significant signals that are automatically forwarded to a scope. These signals are room temperature, heater (thermostat) status, temperature sensor reading, and the CO_2 sensor data. Each significant component will have its monitoring scope, which is automatically generated by this section code. Figure 4.30 indicates the connected signals to scope for a standard room.

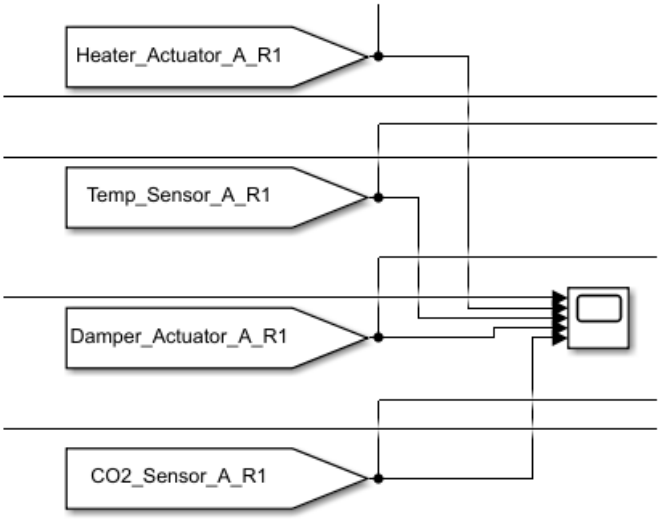


Figure 4.30 Connected signals to scope for a standard room.

- ✓ **SQL communication support:** The model can produce the simulated model's database as the core of easier post-processes for an efficient, reliable, and robust failure detection and fault diagnosis approach. The model is simulated with a fixed step size of one second. All the parameters required for running the simulation are connected to the “To Workspace” Simulink block, which gets the signals and writes their values to a workspace .m file. Data

during the simulation are written to an internal buffer. When the simulation is paused or completed, the data is written to the workspace. Therefore, data is not available until the simulation is paused or finished. The data format of the variables can be in time series, arrays, or structures with time, such as Microsoft Excel spreadsheet file using "xlswrite" MATLAB function or in an external database using a connection algorithm that connects the model to the Microsoft SQL Server. The Relational Database Management System (RDMS) manages a huge amount of data in linked tables to make them straightforward in understanding. RDMS provides a relationally comprehensive language for the data definition, retrieval, and update. The effective use of SQL can minimize the amount of data sent across the network as it performs tasks by the database rather than requiring application code to perform them. SQL is a structured query language used to create and manage data held in a database management system. Gupta and Mittal describe that SQL is based on relational algebra and tuple relational calculus [190]. SQL offers multiple statements used to manage the stored databases [191]. The SQL offers data manipulation as (insert, delete, and update), query, definition, access control [192]. SQL's primary purpose is to query data that exists in a database. SQL is a declarative programming language, not an imperative programing language like C/C++ and MATLAB. Declarative languages differ from imperative languages, that declarative languages operate with statements to execute orders while imperative languages use algorithms to achieve the programmer tasks[5].

After the model is generated from the previous code sections, the 9th step is to establish a connection to a database server and send data while the model simulation is running. This approach has the advantage of running model diagnostics on a remote and separate platform rather than on the system machine itself. The idea seemed applicable and not applied before. Nevertheless, unfortunately, it ran into a few problems to achieve it.

First of all, to understand the challenges faced during this task, one must understand how Simulink behaves while simulating a model. MATLAB stores its variables in "Base Workspace" and is separated from the Simulink workspace, which is created only on a simulation runtime and remains in the memory while the simulation is running. When the simulation is finished, Simulink variables are transferred into the MATLAB workspace. Therefore, a problem is imposed as data is frequently updated into the SQL database, but the workspace is not accessed frequently in simulation. During the simulation, multiple methods were tested to achieve frequent data communication between the Simulink and the SQL server.

Such methods like sending data between simulation pauses are controlled by Simulink model callbacks[6] or controlling simulation behavior using "assertion blocks"[7] to achieve a data transmission every (nth) time step. These methods failed to achieve the desired behavior because it was found that Simulink has strange behavior when it is paused programmatically, for example: if the simulation were to be paused using the *set_param()* command mentioned in (table 3.1, line: 21), the MATLAB would ignore to update the workspace. On the other hand, if Simulink was paused manually, the data will be updated in the workspace. Furthermore, in the model callbacks and the assertion blocks approaches, the Simulink accepts the pause command but refuses the continue command, which role is to ask for the simulation to continue automatically (table 3.1, line: 22).

As a last resort, the Simulink Level-2 MATLAB S-Function block was tested to fulfill this task, and this function block allows the user to use either MATLAB or C language to

[5] https://de.mathworks.com/help/simulink/ug/model-callbacks.html
[6] https://de.mathworks.com/help/simulink/ug/model-callbacks.html
[7] https://de.mathworks.com/help/simulink/ug/controlling-execution-of-a-simulation.html

create a custom Simulink block to do a particular function during the simulation[8]. This method proved to update the SQL server with data either each time step or every (n^{th}) time step. Nevertheless, it was found that this block causes a significant delay in the simulation time due to its real-time variable processing. The illustration and functionality of the ***S-function*** block are shown in Figure 4.31.

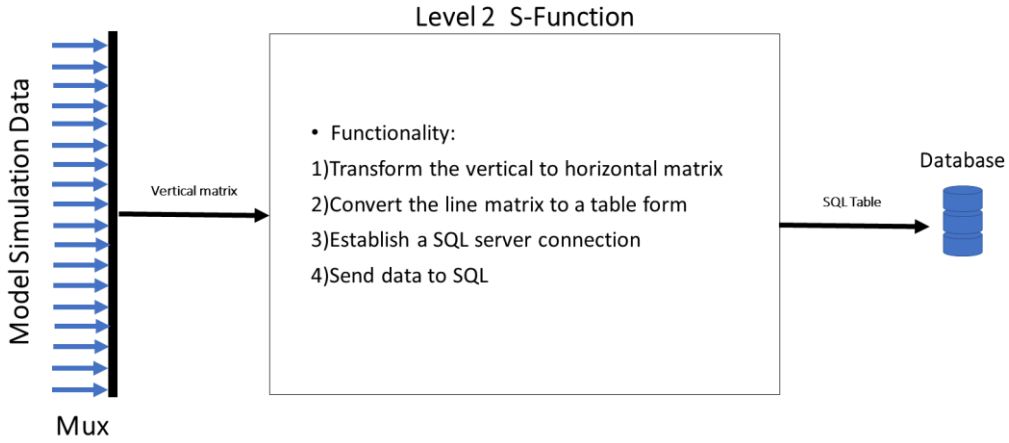


Figure 4.31 Simulink Level-2 MATLAB S-Function functionality

✓ GUI eliminates the need to use programming languages to execute a command in an application. The GUIs help the user to navigate and control the generated model easily by setting the parameters in a graphical and straightforward view. The Callbacks function helps the user to edit the GUI later. MATLAB offers the ability to design and run a GUI for various ranges of applications.

```
function simple_gui                                    % call the GUI function

f = figure('Visible','off','Position',[360,500,450,285]); % add the empty window

hcontour = uicontrol'Style','pushbutton',...
'String','Contour','Position',[315,135,70,25]); % creates pushbutton
```

In this thesis, a GUI is generated programmatically to help the user navigate and control the generated model. The shown function is an example of how a simple element is added to the GUI, after that the user can control the GUI elements behavior using function callbacks. The designed GUI in the composable model is shown in Figure 4.32 and includes these parts:

- In the component and fault parameters selector section, the user can specify the *Room* or *Corridor* and the type of component desired for monitoring or fault injection. This model can be used to study the system behavior (such as temperature signal, actuators' status, and sensor values) in case of different types of faults. Also, the user can specify the component index, fault intensity, and the delay, which is representative of the time of fault injection.
- In the *Fault Type* section, the user can choose the fault type and fault category, i.e., *Heater_Actuator_FI,* from the first fault category of *Hardware Fault*.

[8] https://de.mathworks.com/help/simulink/sfg/writing-level-2-matlab-s-functions.html

- In the *Monitoring and Scopes* section, the user can view the system behavior signals in the desired room by simply entering the room index.
- In the *Fault Signal Controller,* the user can choose the mean and variance values of the Temperature and CO_2 Sensor value.
- In the *Simulation Controls* section, the user can specify the intended duration of the simulation (simulation time). Also, there is an ability to start/stop/pause the simulation.

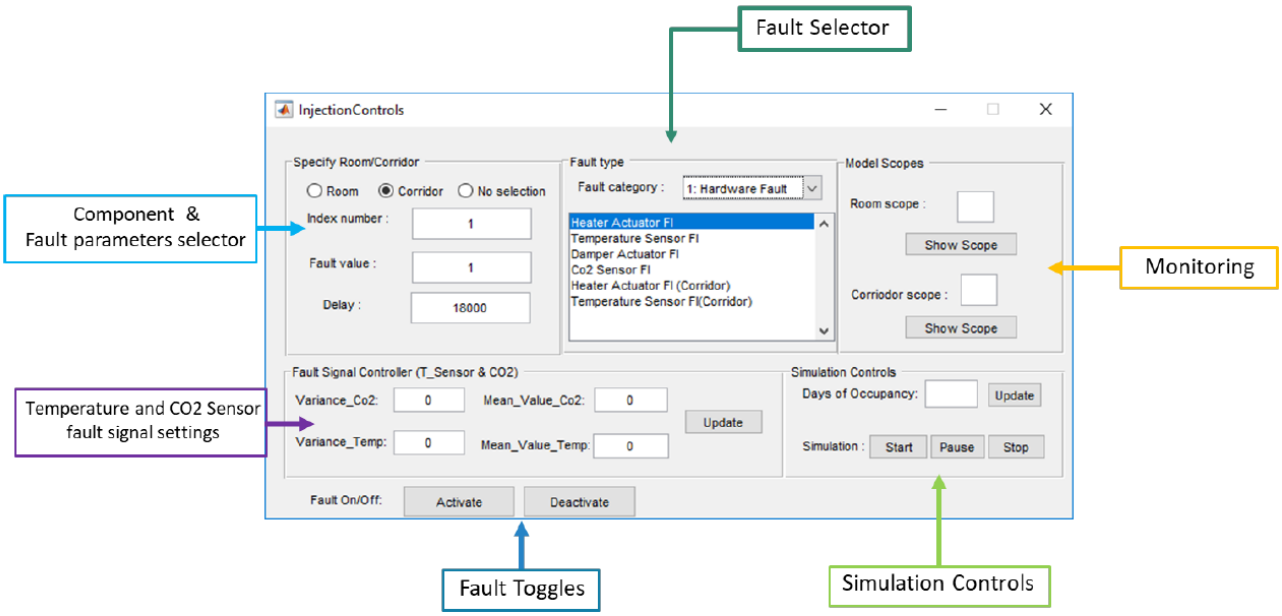


Figure 4.32 The Designed GUI [163].

4.2.4. COMPOSABLE MODEL VALIDATION

The composable model provided in this thesis is also validated by mapping the temperature signal's validation property, comparing the model signals collected from the basic model provided in section 4.1. and the composable model at the same architecture with the six rooms and one corridor. Figure 4.33 describes the validation results with an accurate matching result of mapped temperature signals in two models. In this figure, the sample temperature signals are shown from room number one.

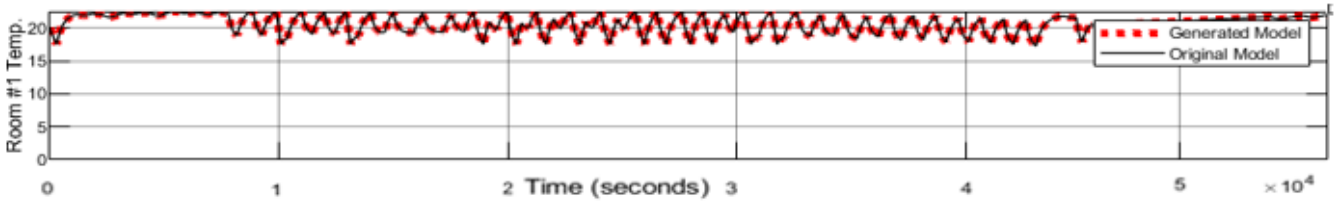


Figure 4.33 Model Validation Result [163].

4.2.5. DEMO

After demonstrating the theory and the main parts of the code that generate the new model, the results and outcomes are discussed. After the script is compiled, MATLAB will call the blocks from the predefined library to create the model, and after it finishes, it will show the model in a Simulink window with all the essential features. In Figure 4.34, the user has selected that a 6-room office is to be simulated, and after running the MATLAB script, this figure will be produced. The most important parts of the model are highlighted in this figure. The model will consist of three

crucial zones. The first one is the composable model block which contains the model's core with all the main components. In the second zone, the user is offered the scope of each system's main component (i.e., Room1, Corridor1). Each scope will have the main signals that run in each component, so the user can monitor the desired components when triggering fault values to study and analyze the system behavior.

In Figure 4.35**,** another example is given with a more expanded model. In this model, an office with 48 rooms and two floors is simulated. As shown in figure 4.35, the model becomes more complex, and the simulation time to produce such a system increases too. Also, note that the more sophisticated and enlarged, the more time it might need to transfer data to the SQL server. It is advised to run such a system on multithreaded machines. After the model is generated, the user can access the real-time simulation data for each component in the monitoring section or use the GUI as demonstrated in section 4.2.3. Figure 4.36 is an example of the scope of the data generated from a room. Each room will have five signals in its respective scope (Room Temperature, Heater-Actuator Fault Injection, Temperature Sensor Fault Injection, Damper Actuator Fault Injection, CO_2 Sensor Fault Injection). All these signals are updated during each time step in the simulation. The signals that carry the FI in their names represent the data in the scopes of Figure 4.35.

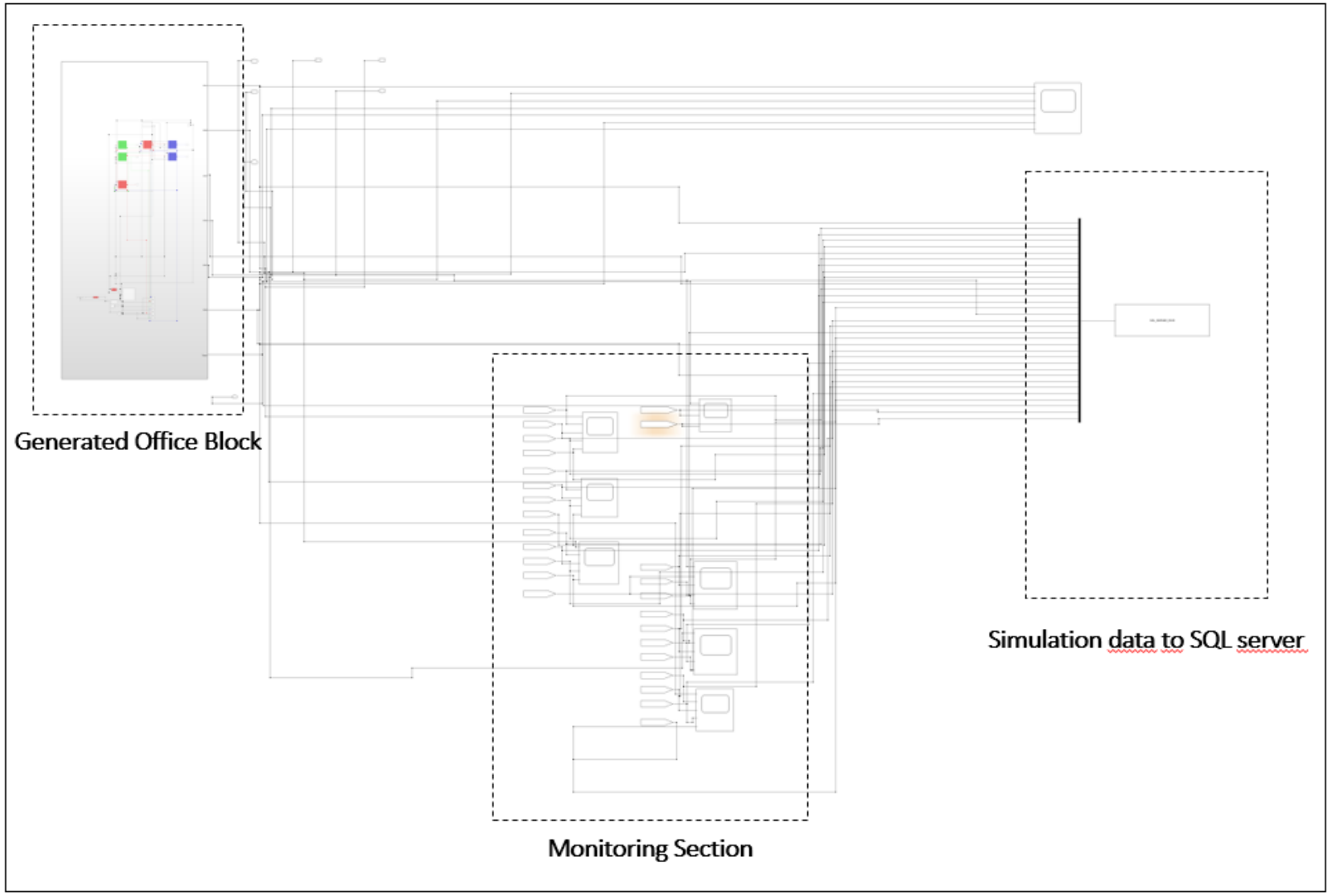


Figure 4.34 Overview of the composable architecture of a model with six rooms on one floor

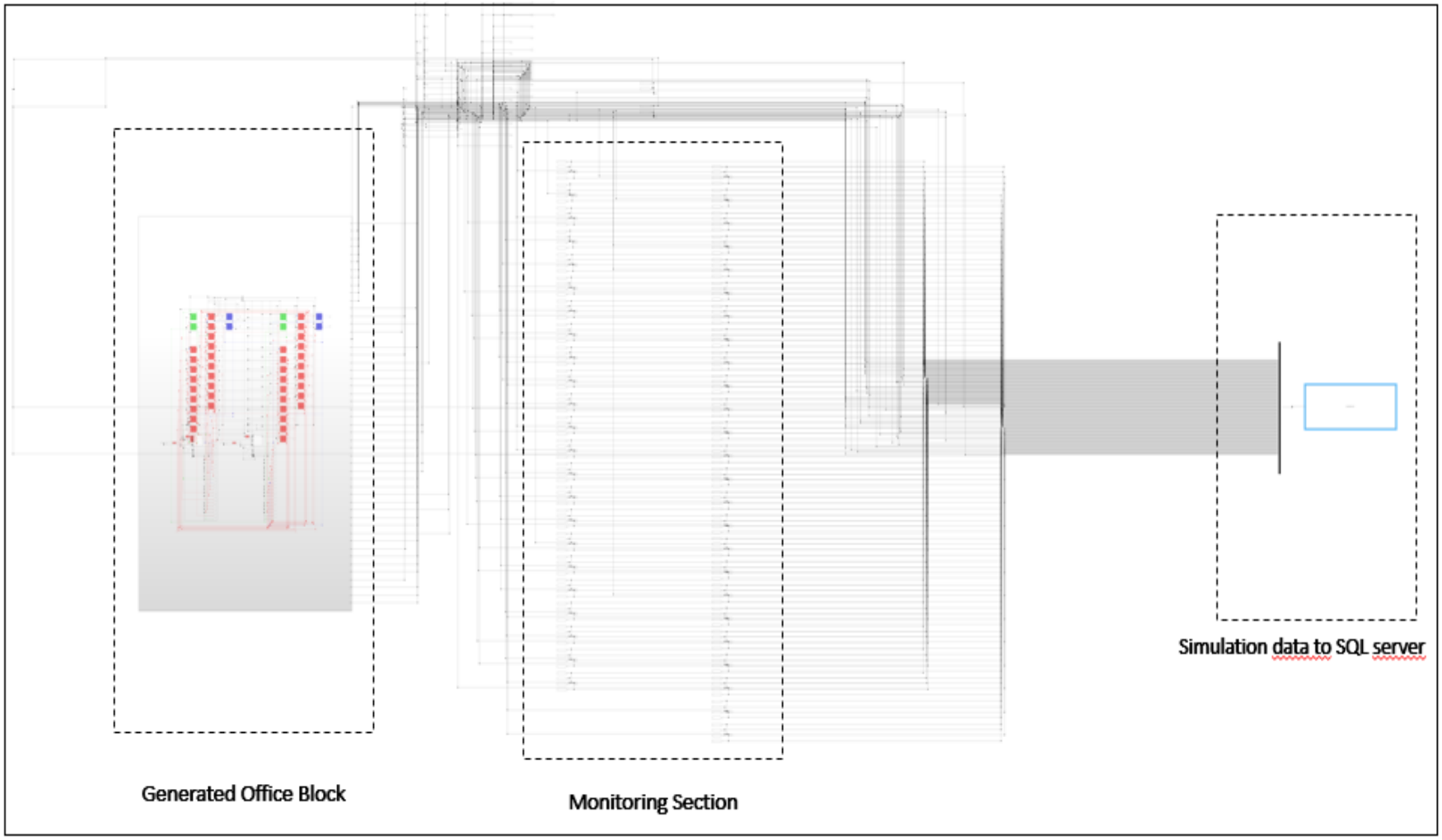


Figure 4.35 Overview of the composable model with 48 rooms on two floors.

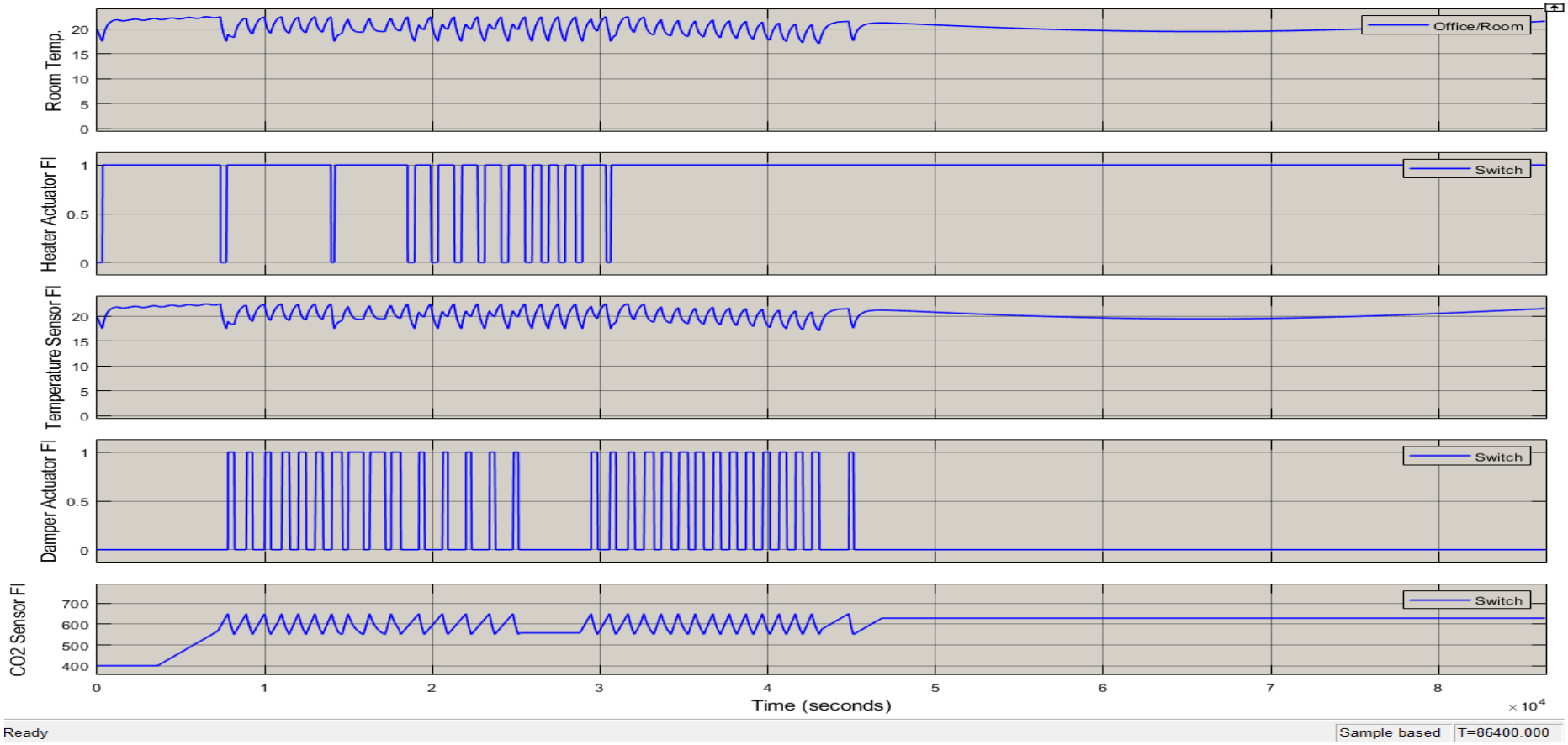


Figure 4.36 Example of a generated Room scope with healthy data.

Finally, the data can be optionally sent and stored in a SQL server, in Figure 4.37 is an example of simulation data stored in an SQL server.

Data Preview (First 100 Rows)

	Temperture_room1	HeaterActuator_room1	TempertureSensor_room1	DamperActuator_room1	Co2Sensor_room1	Temperture_room2	HeaterActuator_room2	TempertureSensor_room2	Damp
1	20	0	20	0	400	20	0	20	
2	19.9905	0	19.9905	0	400	19.9934	0	19.9934	
3	19.9810	0	19.9810	0	400	19.9868	0	19.9868	
4	19.9715	0	19.9715	0	400	19.9802	0	19.9802	
5	19.9620	0	19.9620	0	400	19.9735	0	19.9735	
6	19.9526	0	19.9526	0	400	19.9669	0	19.9669	
7	19.9431	0	19.9431	0	400	19.9603	0	19.9603	
8	19.9337	0	19.9337	0	400	19.9536	0	19.9536	
9	19.9242	0	19.9242	0	400	19.9470	0	19.9470	
10	19.9148	0	19.9148	0	400	19.9403	0	19.9403	
11	19.9054	0	19.9054	0	400	19.9336	0	19.9336	

Figure 4.37 A sample of SQL Data from the simulated model.

5. FAULT INJECTION FRAMEWORK

The mechatronic systems in cyber-physical systems such as DCV and heating systems include many sensors and actuators besides computational nodes and communication networks. Faults are inevitable events that affect the components, the system's functionality, and performance. Diagnostic techniques can serve for triggering appropriate recovery actions to achieve an acceptable level of service despite occurring faults. Considering the vast range of diagnostic methods, establishing an accurate diagnostic model that maps system failures to suitable faults is a time-consuming task that may involve extensive try and error efforts [160]. A fault injection framework is a helpful tool for studying the behavior in the presence of faults and evaluating diagnostic techniques. In this thesis, fault injection is used as a technique to determine the coverage of the fault diagnosis by producing faults in the system to trace the behavior of the system in existence or absence of different kinds of faults and evaluate their effect by monitoring several parameters, e.g., energy consumption and occupancy comfort. Fault injection framework this thesis evaluates the system and validates diagnostic services in the presence of faults.

5.1. FAULT

Faults may lead to a system-level or component-level failure or malfunction if they are not detected and mitigated. Failures can involve performance degradation, safety risks, and excess cost, and energy waste. From the time perspective, faults may occur during the whole operation of the system or may be limited to a specific period and an exact time. Faults with time dependency can be categorized into abrupt faults (stepwise/short), incipient faults, constant faults, noisy faults, and intermittent faults [28]. In the abrupt faults, a sharp value change occurs compared to the usual pattern of values. Constant faults arise when a sensor reports a constant value over time instead of the real and normal sensor values or when an actuator is stuck at a constant position. In the noise faults, faults affect several samples in contrast to a single sample in the abrupt faults [193]. An increased range of noise in the measured values may cause noisy faults, e.g., noisy sensor values [194]. Many types of faults are known in HVAC systems, such as design faults, installation faults, abrupt faults, and degradation faults categorized by many references such as the survey of the international energy agency [195]. In the DCV and heating systems, faults can occur either in components, computational nodes, or communication platforms. This section shows some examples of these faults.

5.1.1. FAULTS IN COMPONENTS

Faults in actuators may lead to loss of controllability. In sensors, they can affect reliable measurement information, and in the computational nodes, faults will change the behavior of the entire plant. Some examples of component faults used in this thesis are listed below:

- **CO_2 Sensor Fault**

  The CO_2 sensor fault represents a wrong sensor reading that can be a constant or noisy value, such as a constant value of 700 ppm or noisy values within the range of 550 ppm to 750 ppm. Figure 5.1 and Figure 5.2 show the simulation blocks for injecting this type of fault and the switch model that assigns the value of the fault.

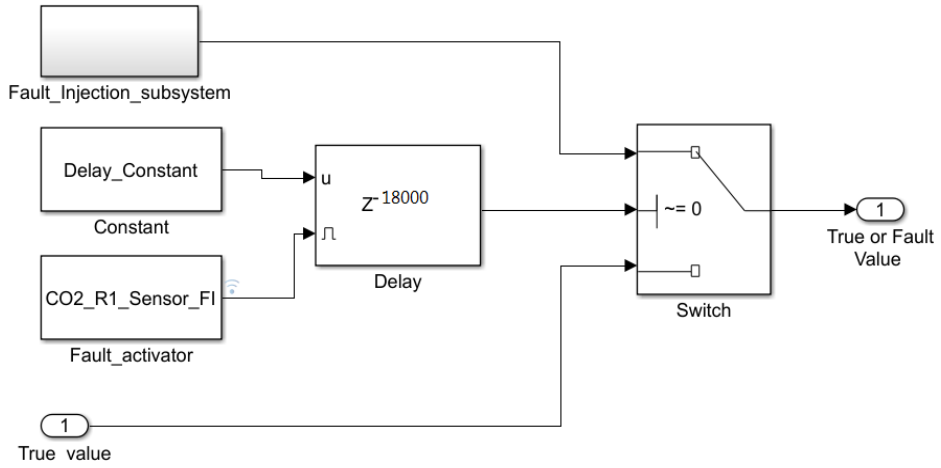


Figure 5.1 Simulink model for fault injection inside the CO_2 sensor fault injection subsystem.

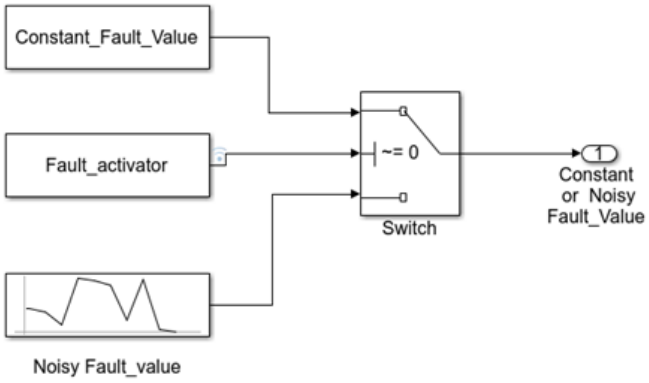


Figure 5.2 Switch for noisy or constant value fault injection.

- **Temperature Sensor Fault**

  This type represents a wrong sensor reading with a constant value (e.g., 15°C).

- **Damper Actuator Fault**

  This type of fault represents a stuck-at fault where a damper sticks at a specific position. For example, if the damper is stuck at its open position, it gets the binary value 1, which means that excess low-temperature fresh air comes inside. Therefore, the inside temperature will decrease, and the heater must constantly work to compensate for the heat loss. If the damper is stuck at its closed position, it gets the value 0, which means that the inside air temperature will increase, and the indoor CO_2 concentration will pass the maximum permitted limit.

- **Heater Actuator (Thermostat) Fault**

  This type of fault represents a stuck-at fault where the heater sticks at a specific position. For example, if the heater is stuck at its ON position, it gets the value 1, which means inside air temperature tends to increase. If the heater is stuck at its OFF position, it gets the binary value 0, which means the inside air temperature tends to decrease.

5.1.2. FAULTS IN NETWORK

Typically, WSAN must satisfy requirements such as robustness, fault tolerance, self-configuration, and increased life span despite their fault-proneness. The incipient faults in WSANs that are not identified and detected can degrade the control performance, safety, and Quality of Service (QoS), higher maintenance cost, and efforts. Various faults are possible in WSANs that their

nature can categorize as hardware, software, and communication faults. Faults in hardware modules, e.g., sensors and actuators, may cause hardware failures. Programming problems may cause software failures. The issues arising in the data transmission, such as interference faults, may cause communication failures in a network. Sensor nodes are resource-constrained and may report incorrect sensor readings or fail [196]. Links in the network may fail, or the data packets may be stuck in a loop due to a link disconnection [197]. Early detection of wireless network faults prevents subsequent failures and enhances the lifetime of the network. Raposo et al. illustrate a comprehensive WSN fault taxonomy based on the phase of creation or occurrence, system boundaries, phenomenological causes, dimensions, objectives, intent, capability, persistence, state, reproducibility, and source system [198]. This thesis selects relevant fault cases from this reference to the model and implements them.

- **Fault Classification based on Phase of Creation or Occurrence:**

Three types of faults concerning the phase of creation or occurrence in wireless sensor networks are requirement faults, deployment faults, and operational faults [198]. Requirement faults in WSN design occur when setting the system requirements where functional and non-functional requirements are incorrectly specified. Some of the requirements may be ambiguous, and specification errors may occur. Development faults occur in the development phase or case of software/firmware update and hardware production. Operational faults occur after WSN deployment when the network is operating.

- **System Boundary Fault**

Each system has a boundary, which defines the border between the external environment and the system components. The faults occurring inside the WSN components are internal faults. For example, firmware or hardware component faults are internal faults due to their source inside the WSN system [198]. The faults could also originate outside the system in the environment. An external system can, in turn, lead to fault propagation into the system by the interaction between external systems and the WSN. For instance, channel noise, radiation, electromagnetic interference, operator mistakes, and environmental extremes belong to this category.

- **Persistence**

There are three types of faults based on the time span. The first category is transient faults. Transient faults are defined as temporary faults caused by interaction with an external system in particular conditions and in a short period of time. The second type is intermittent faults, which are defined as temporary faults that occur randomly and repeatedly. This fault can arise in logic components, e.g., software, or physical components, e.g., hardware. For example, a low battery voltage may lead to intermittent hardware faults [198]. The last type is permanent faults which are defined as faults that always produce errors. Specific inputs could trigger faults in the main firmware loop of the sensor node that always generate an error [198].

- **Source System Fault**

WSN systems contain several components or subsystems, like the energy supply sub-system, the data acquisition subsystem, the processing, storage sub-system, and the communication subsystem. All these parts can be sources of faults and should be taken into account for fault characterization.

- **Data Acquisition Faults**

These types of faults occur due to biased or faulty sensor readings. Sometimes hardware faults occur in the data acquisition sub-system of sensors which may lead to the wrong sample readings. Data acquisition faults are called soft faults because the data acquisition faults take into consideration the characteristics of the sensed data in determining the faults. They can be categorized into various categories such as offset fault, gain fault, stuck-at fault, out of bounds, spike faults, and data loss fault.

Offset fault: It refers to a change in sensed data by an additive constant from the expected data and might occur due to improper calibration of the sensor [199].

Gain fault: In this fault, the rate of change of sensed data does not match expectations over an extended time. In a gain fault, the non-faulty sensor data gets multiplied by a constant value, and it also might be caused by an improper calibration of the sensors.

Stuck-at fault: In the stuck-at fault, the difference or the variance of data from the data series is zero, which implies that the sensed data is constant.

Data loss fault: A data loss fault occurs when measured data is lost from the time series for a given node [199].

- **Energy Supply Faults**

WSNs can be deployed in all indoor or outdoor conditions, where they are subject to all kinds of conditions that can affect the performance of the power source, such as batteries or super-capacitors. This type of fault refers to faults that are related to the sensor node power source. Battery failures are the primary cause of inaccurate data and battery depletion, leading to the transmission of erroneous data by the sensors. The battery voltage may drop to the point where the sensor can not report correct data. Common behaviors could be zero variance or an unexpected gradient followed by a lack of data. The effect of battery supply on the system performance can be by either adding noise or giving incorrect data depending on the type of sensor [200], [201].

- **Communication Faults**

Communication is considered one of the primary sources of faults in WSNs, because of its inherently distributed and dynamic nature. The wireless links may fail permanently or temporarily when an external object, environmental changes block the link. Communication faults can be transient and intermittent [200]. Wireless communication is affected by many types of interference, like ambient noise, channel noise, multipath fading, and RF interference. Moreover, sensor nodes may also be subject to several types of faults. Furthermore, routing faults can be caused by errors in routing algorithms and protocols ,leading to traffic being caught in network loops and never arriving at the destination. Additionally, the routing algorithm is not the only cause of a fault in communication networks. Sensor nodes also could suffer from faults related to message processing. For example, if nodes are slow and cannot process messages in time, then local congestion can occur in the buffers between the various protocol layers. As a result, some messages may have to be dropped [198].

- **Processing and Storage Faults**

The processing and storage fault can occur in the WSN hardware and the software/firmware, affecting either the quality or consistency of the stored data or the operations performed on them [202]. Bit flips in memory, or special registers are one example of a processing fault. Another example is increasing the value of the processor temperature, which may cause the drifting of the clock and loss of synchronization of the sensor node from the network [198]. Table 5.1 summarizes the types of faults which were mentioned above.

Table 5.1 Fault Types in WSNs [198]

Viewpoints	Fault types	Examples
Phase of Creation or Occurrence	Requirement faults, Development faults, Operational faults	Wrong routing protocol and wrong topology, poor design, architectural faults, software coding mistakes, route misconfiguration during deployment, battery depletion, wrong configuration or reconfiguration parameters
System Boundary	Internal faults, External faults	Bit-flip faults in memory or special registers, logical bridging, physical bridging, battery damages, channel noise, environmental extremes, operator mistakes, radiation and electromagnetic interference
Persistence	Transient faults, Intermittent faults, Permanent faults	Timing fault, fail-stop fault; hardware noise fault; incorrect computation fault, crash fault, physical damage faults (corrosion, strokes, fires)
Source System	Energy supply faults, Data acquisition faults, Processing and storage faults, Communication faults	Battery degradation fault, low battery voltage faults, gain data fault, offset data fault, corrupted routing, maintenance packets fault, routing loops fault, degraded route path fault

5.2. FAULT INJECTION IMPLEMENTATION

In this thesis, we select data-centric faults and system-centric faults for the implementation. The fault injection framework of this thesis can inject faults at any time using a delay block. However, the default setting for the start of fault injection is 18,000 seconds from starting the simulation and can be changed in MATLAB Workspace.

5.2.1. DATA-CENTRIC FAULTS

The data-centric faults are related to the generated data from the components. In this thesis, the faults in this category are modeled based on the measured values from the sensors such as temperature sensors, CO_2 concentration sensors, and command or status values in actuators, e.g., heater actuators (thermostats) and damper actuators.

The equation that describes the generated data from a sensor and actuator node can be modeled as a function concerning time *f (t)* [199], as can be seen

$$f(t) = x \tag{5.1}$$

Where *f(t)* represents the value sensed by the node at the time t, x is the non-faulty sensor value at the time *t*, knowing that in the real world, there is no ideal signal. Therefore the noise n will be added to Eq 5.1. As a result, this equation can be written as the following:

$$f(t) = x + n \tag{5.2}$$

Some factors are added to normalize Eq.5.2. For instance, A, B, n are factors that can determine the fault type, where A is the multiplicative constant and called gain, whereas B is the additive constant

and called offset, and *n* is the external noise. According to the above, the general form of the equation 5.2 can be seen here

$$x' = Ax + B + n \tag{5.3}$$

Four different types of faults are modeled as examples, gain fault, offset fault, noise fault, and stuck-at fault. A gain fault can be modeled with the following equation where $x' \in f(t)$. Figure 5.3.A demonstrates a gain fault with the values of $A = 4, B = 0, n = 0$.

$$x' = Ax \tag{5.4}$$

The same procedures are repeated with the offset fault, as seen in Eq.5.5, where B is the constant value added to the expected value. Offset fault refers to a deviation in sensed data by an additive constant from the expected data, as can be seen in Figure 5.3.B

$$x' = x + B \tag{5.5}$$

The noise fault is defined as a random value added to the expected value, which leads to erroneous data. The noise fault is described in Eq.5.6 and can be seen in Figure 5.3.C

$$x' = x + n \tag{5.6}$$

Knowing that *x* represents the data and should not be zero in the case of offset and noise fault, otherwise, the fault will be considered a stuck fault as seen in Eq.5.7. Stuck-at fault can be defined as faulty data that has a constant value. Figure 5.3.D shows the stuck-at fault, where the variance of the measured values is zero and the sensed data at the steady-state with the constant value $A = 0, B = 40, n = 0$

$$x' = B \tag{5.7}$$

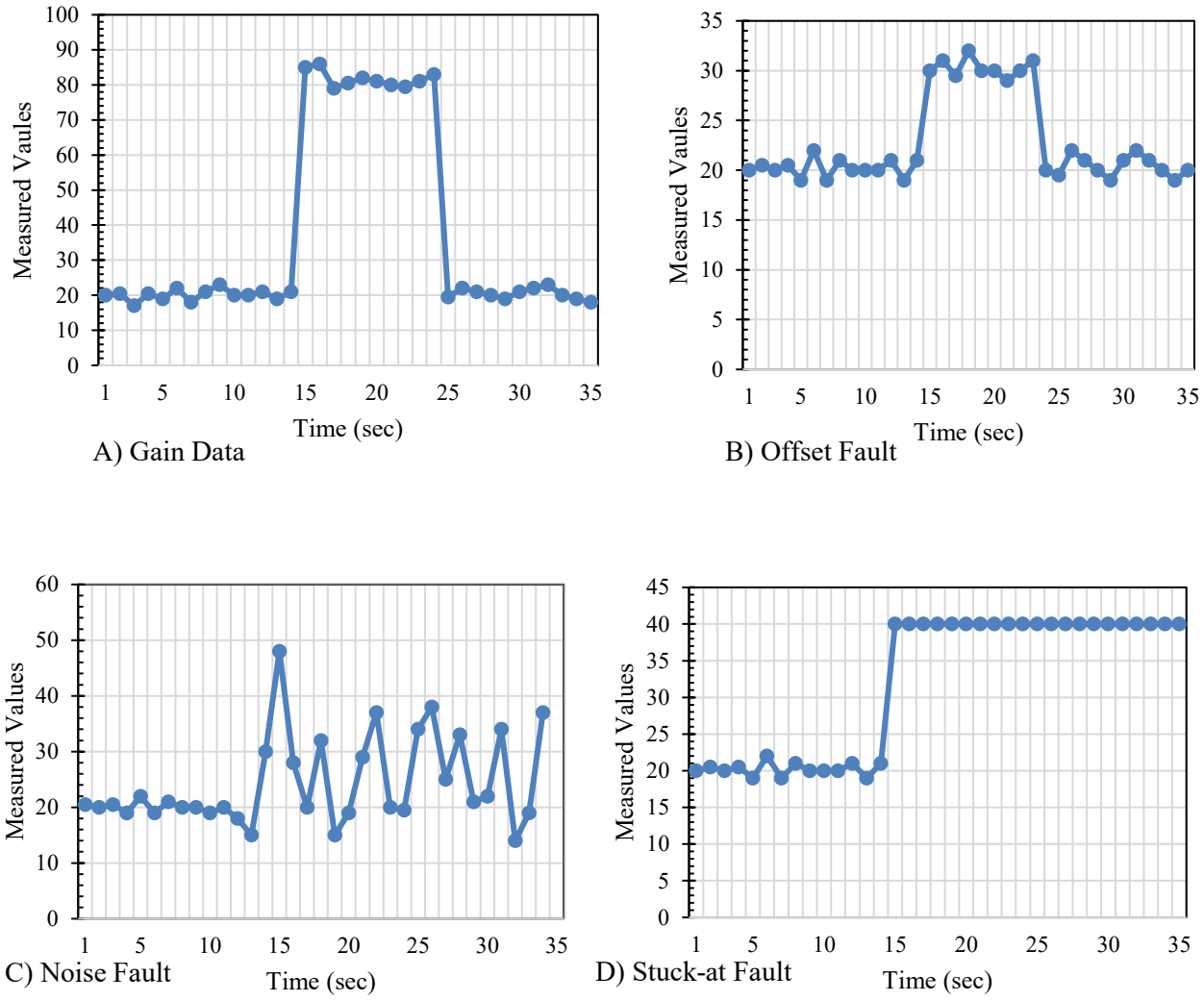


Figure 5.3 Data-Centric Fault

5.2.2.SYSTEM CENTRIC FAULTS

Communication in a network is considered one of the significant sources of faults in sensor networks since wireless communication is usually subject to considerable interference, e.g., ambient noise, channel noise, multipath fading, and RF interference affects the communication between sensor nodes and may lead to faulty data. Moreover, sensor nodes may also be subject to several types of faults [199]. In this thesis, two types of communication faults are modeled including low battery faults and routing faults. Figure 5.4 demonstrates the low battery fault, showing the effect of decreasing the battery voltage level on the measured readings. The data of the sensor begins to be faulty at the time when the voltage decreases under a particular value. For example, when the battery voltage falls under 82% of the required voltage, the temperature sensor readings remain at a constant value of 22 for the rest of the operation.

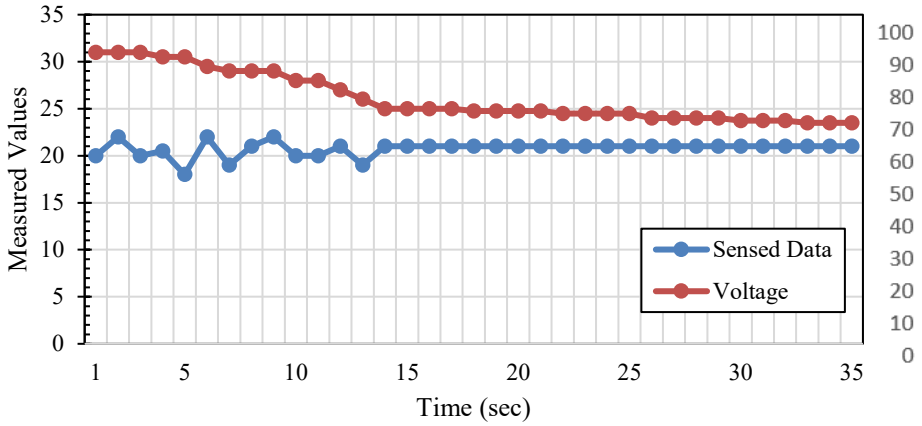


Figure 5.4 Low Battery Fault

On the other hand, a routing fault is considered a type of communication fault that can occur due to several reasons. For instance, routing algorithms and protocols errors can lead to packets being caught in network loops and never arriving at their destination. Sensor nodes can also cause a fault related to message processing, which means that nodes cannot process messages at the rate they receive them or due to local congestion in the buffers between the various protocol layers [199]. As a result of the routing fault, the data, which should be transmitted to the coordinator will be lost. Figure 5.5 shows the sensed data by the sensor and the loss of data after a specific time when the routing fault activation takes place.

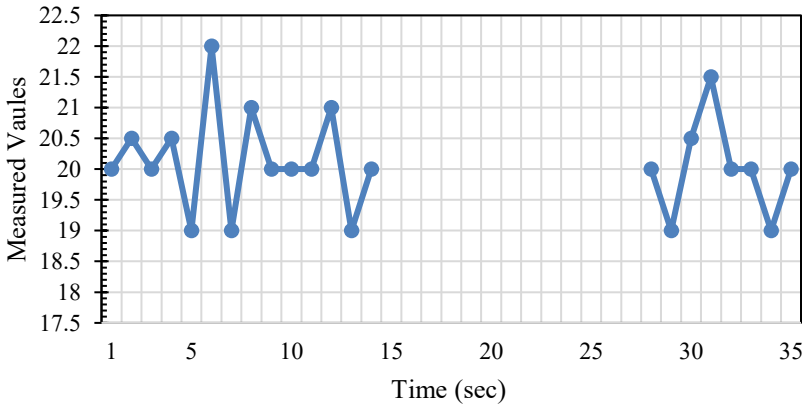


Figure 5.5 Routing Algorithm Faults

Figure 5.6 demonstrates a routing loop on a ZigBee protocol between two nodes where node A (source node) transmits the data to node C (destination node) via node B (neighboring node). At the same time, node B has information that the data can be transmitted from node A to C. If the link between nodes A and C breaks down and A has not updated node B regarding this issue, therefore, node B transmits the data to node A assuming that the link A-B-C is yet working. Node A knows the broken link and sends the data to node B. Node B consults its routing table and finds node A as the best route as it is not informed, thus sends its data back to node A that creates an infinite loop.

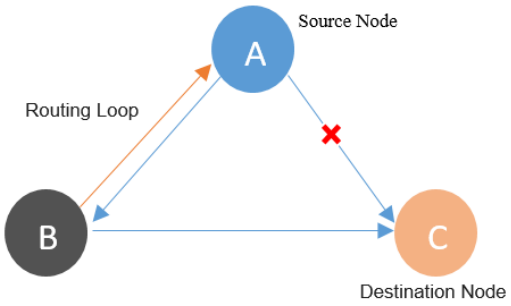


Figure 5.6 Routing Loop

5.2.3. GRAPHICAL USER INTERFACE

The fault injection dashboard is designed to activate/inject various faults, including data-centric and system-centric faults, using a GUI. The dashboard includes a range of various faults related to the components, e.g., CO_2 sensor fault, damper actuator fault, temperature sensor fault, and heater actuator (thermostat) fault, or with a focus on the communication (network fabric). Figure 5.7 exemplifies the developed GUI in this thesis.

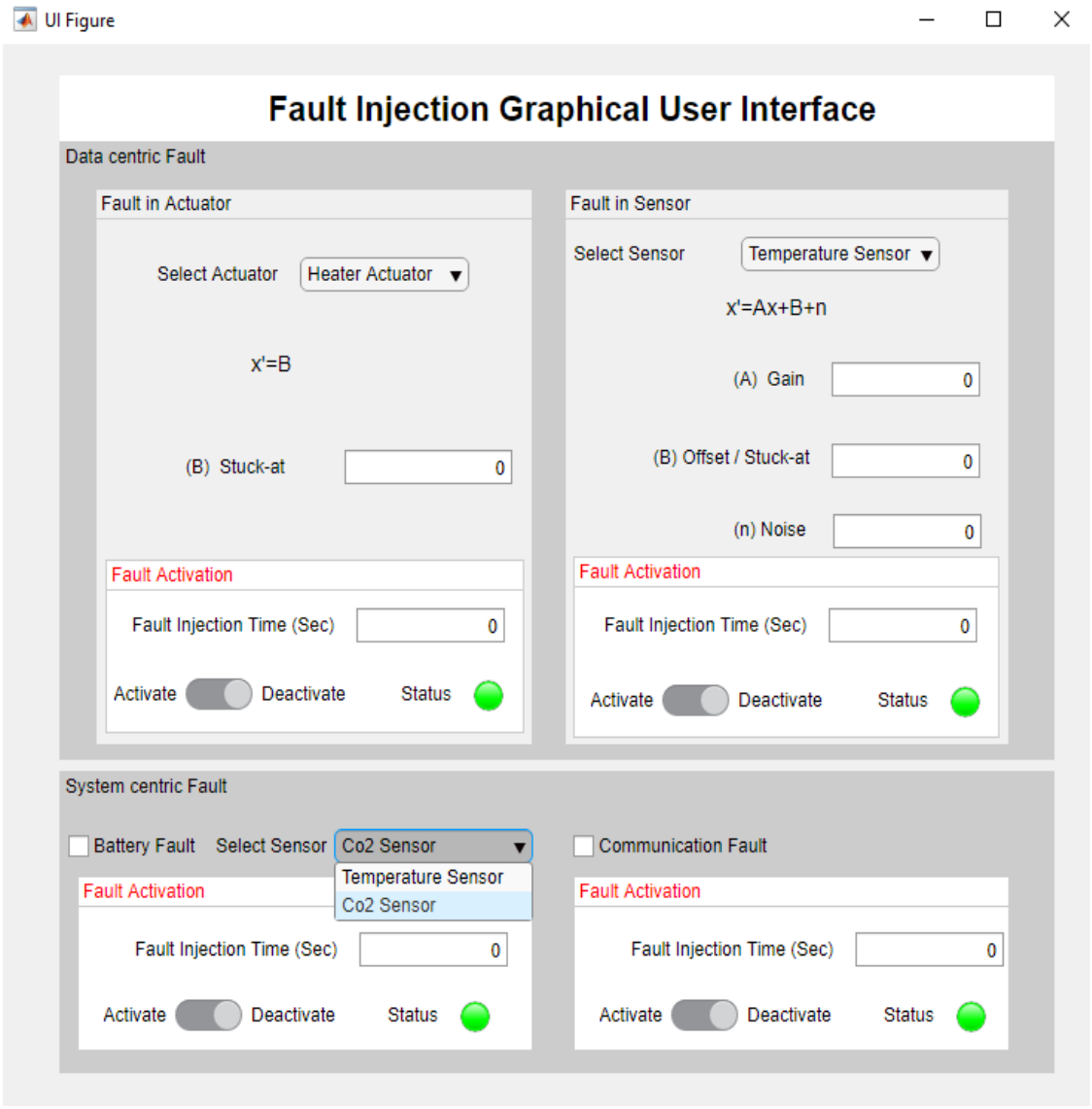


Figure 5.7 The Fault Injection GUI.

For the data-centric faults, if x is the measured value of a sensor or the status of an actuator, the equation below gets the faulty value of $\dot{x}$ for different faults:

$$\dot{x} = Ax + B + n \tag{5.8}$$

Where A is the gain value, B is named the offset value, n describes the noise value. The mentioned values are set in the designed GUI. For the gain faults, A and n are required. The offset faults are expressed with *A, B,* and *n,* while noise faults only depend on the n value. To model the stuck-at faults, $A = 0$, $n = 0$, and B gets a constant value where $B = 1$ for the actuators means the heater is stuck-at ON and the damper actuator is stuck-at the fully opened position while $B = 0$ is related to the heater in OFF and the damper actuator is in a fully closed position or a value in this range ($0 \le B \le 1$) such as $B = 0.5$ shows that the damper is 50% open [27].

5.2.4. Fault Injection Blocks

In the fault injection framework, the user simply selects a component to inject and simulate the fault. The designed fault injection blocks can be placed into the Simulink model with wired communication networks and wireless ones. Figure 5.8 demonstrates the Simulink model for one

room equipped with four fault injection blocks (subsystems) highlighted in red color. The fault injection blocks are designed so that for the regular operation, they will allow the values to pass through them from the previous block to the next block, and during the fault activation, they will hold the correct values to replace them with the fault values.

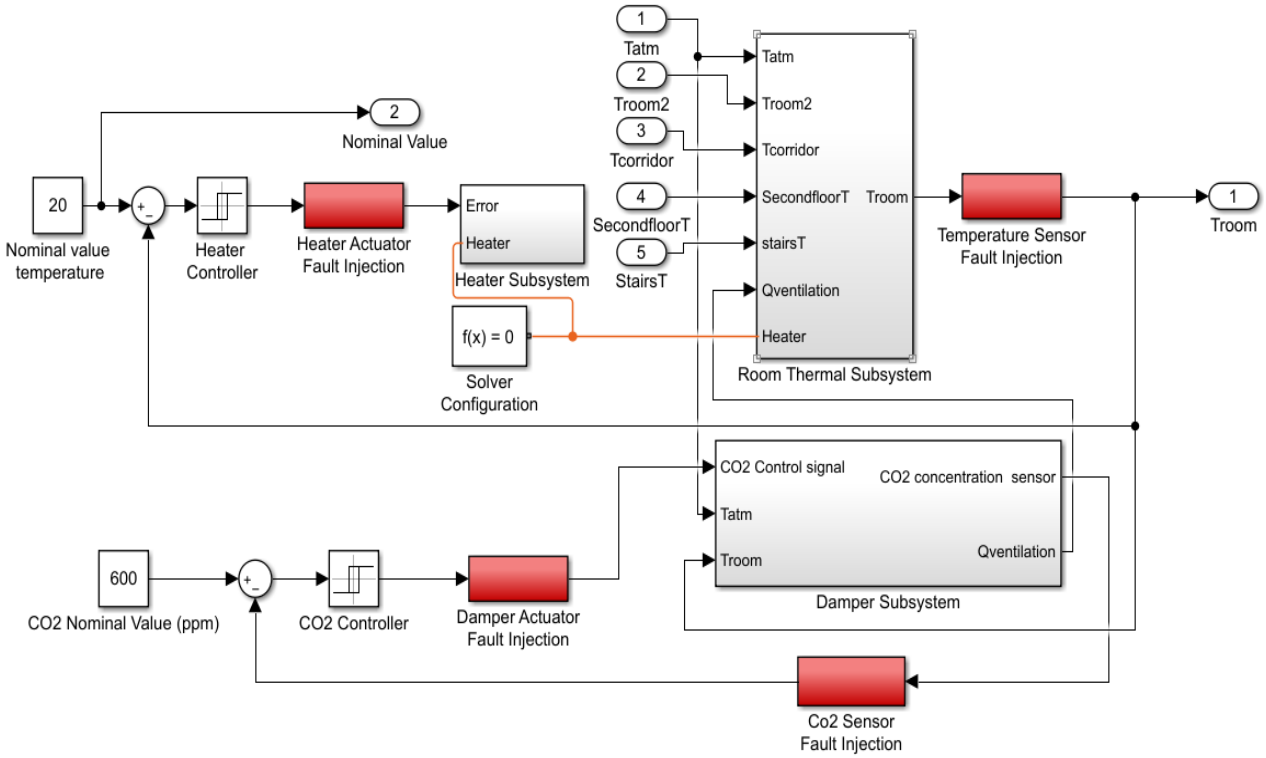


Figure 5.8 Simulink model with FI blocks.

Figure 5.9 describes the working principle of the fault injection block for an example of the fault injection block for the CO_2 sensor. In this example, the above block reflects the desired fault case, e.g., it injects a continuous wrong sensor reading with a constant value of 700 PPM for indoor CO_2 concentration or noisy fault values within the range of 550 PPM to 750 PPM using switch block. The *Switch* block allows values to passes through the first input (faulty values) or the third input (healthy values) based on the value of the second input. The first and third inputs are called data inputs. The second input is called the control input, and this control input is linked to the GUI through a delay block. The value passes through the control input changes based on the switching between the first and the third input values based on the fault activation mode. The *Delay* block produces the input to control the switch block. The delay time is the time to activate the fault injection block set in the GUI.

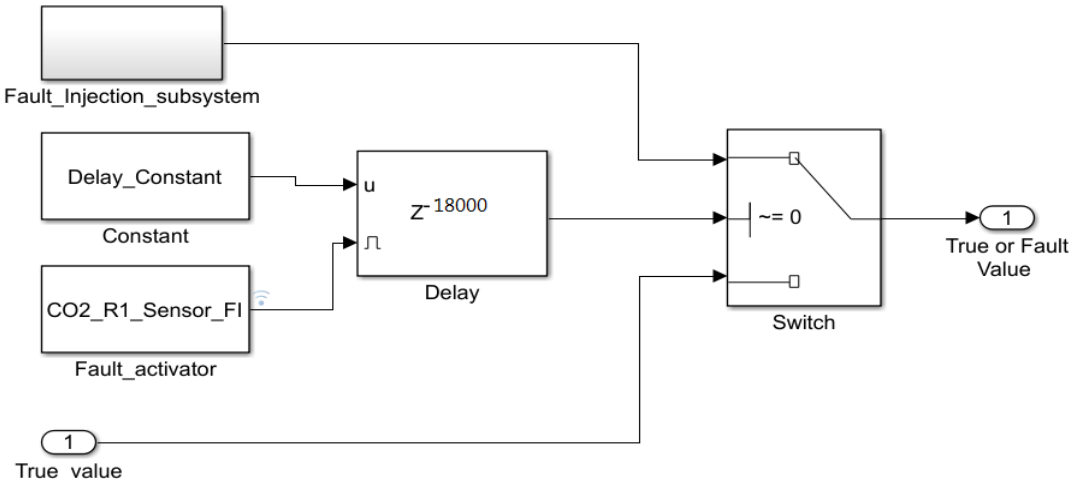


Figure 5.9 Basic components of Fault injection block for the CO_2 sensor.

Also, the fault injection blocks are placed into the model with the wireless communication network. For example, in the communication part, the battery fault due to battery depletion or a communication fault (routing fault) are foreseen. In the fault-free (healthy) mode, the message will be sent to the coordinator over the shortest available path. If the communication link between the node and router breaks, the message will be sent to the nearest router in the adjunct cluster. There, the message will be sent to the coordinator. In the routing loop fault, the next router sends the message back to the source node, and this process will be repeated like a loop. Henceforth, the data will never get to the coordinator, and the message will be stuck in the loop. Algorithm 1 describes the developed routing algorithm in this thesis for the routers of each cluster (each zone).

Algorithm 1 Routing Algorithm.

Algorithm 1: Routing Algorithm (Router's Function)

```
1 Initialization of the TureTime kernel block
2 Call the function of the router A
3 Receive the message from the sensor node or a router
4   IF the routing fault is not activated
5       IF the message is not empty, THEN
6           Forward the message to the coordinator
7       ENDIF
8   ELSEIF (when the route to the coordinator is disconnected)
9       IF the message is not empty, THEN
10          Forward the message from router A to router B
11          Call the function router B
12      END IF
13      IF (when the communication fault is activated)
14          IF the message is not empty, THEN
15              Send back the message to the source
16          END IF
17      END IF
18  END IF
```

Figure 5.10 shows the Simulink model of an office room based on wireless sensors and actuators networks with the designed fault injection blocks in red color established in Simulink using TrueTime [27].

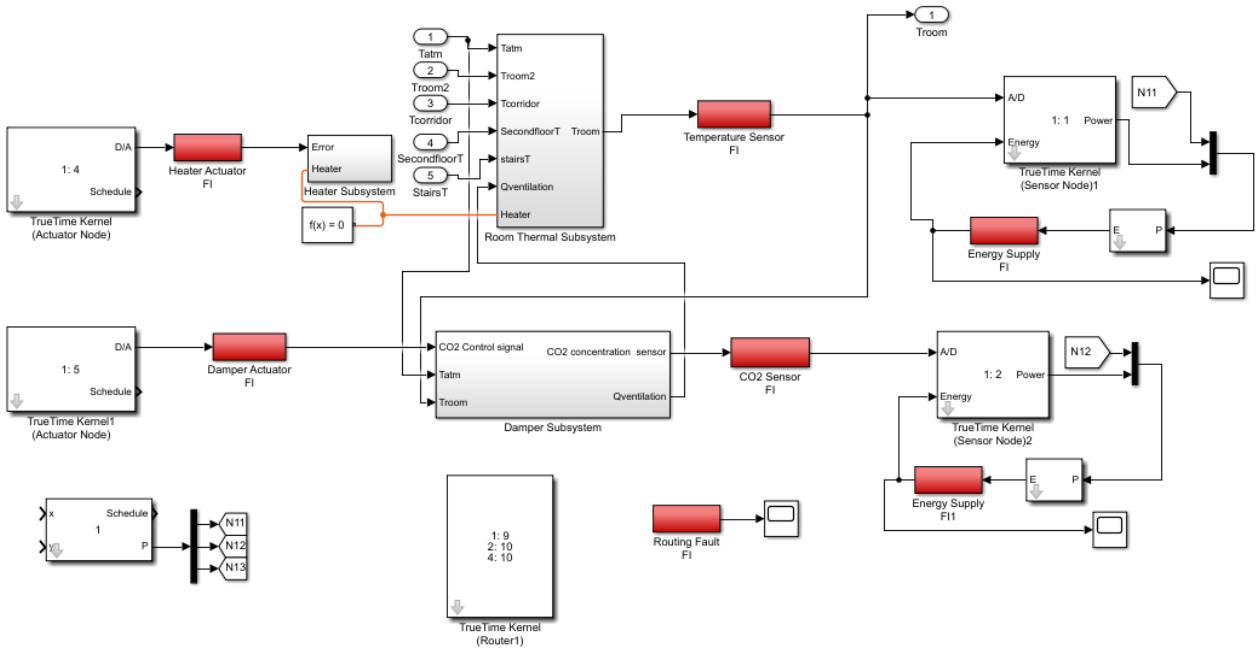


Figure 5.10 Simulink model based on WSAN with FI blocks.

5.2.5. Data Collection and Extraction from the Simulation

The failure detection and fault diagnosis parts are based on the data collected to the data sets from the created demand-controlled ventilation and heat system model with fault injection. The sampling interval for the collected data is one second. Parameters of interest for each office room are measured: the outside and inside temperature, inside CO_2 concentration, occupancy, heater status, and economizer damper status.

The system's behavior in the healthy mode of operation and the presence of faults can be studied by activating faults (fault injection), and the effects can be monitored. Data in healthy and faulty modes are captured in a database from the simulation. Table 5.2.1 shows a sample of captured data in five seconds of simulation for four types of faults in room one. Table 5.2.2 shows the variable changes if a CO_2 sensor fault is injected into the system, and Table 5.2.3 indicates the effect of fault injection of a room on the other rooms.

Table 5.2 Example of captured data of an example room

5.2.1. Example of captured data of room one, in the presence of different fault modes

Simulation Variables			Room #1																			
			Healthy Values				CO_2 Sensor Fault Values				Damper Actuator Fault Values				Temperature Sensor Fault Values				Heater Actuator Fault Values			
Time (Seconds)	Outside Temp (°C)	Occupants	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status
17998	11.8293	6	19.6256	1	583.73	1	19.6256	1	**583.73**	1	19.6256	1	583.73	**1**	**19.6256**	1	583.73	1	19.6256	**1**	583.73	1
17999	11.8294	6	19.6238	1	583.62	1	19.6238	1	**583.62**	1	19.6238	1	583.62	**1**	**19.6238**	1	583.62	1	19.6238	**1**	583.62	1
18000	11.8295	6	19.622	1	583.52	1	19.622	1	**583.52**	1	19.622	1	583.52	**1**	**19.622**	1	583.52	1	19.622	**1**	583.52	1
18001	11.8296	6	19.6202	1	583.41	1	19.6202	1	**700**	1	19.6202	1	583.41	**1**	**15**	1	583.41	1	19.6202	**1**	583.41	1
18002	11.8297	2	19.6184	1	583.12	1	19.6184	1	**700**	1	19.6184	1	583.12	**1**	**15**	1	583.15	1	19.6184	**1**	583.12	1
18003	11.8298	2	19.6167	1	582.84	1	19.6167	1	**700**	1	19.6167	1	582.84	**1**	**15**	1	582.88	1	19.6167	**1**	582.84	1

5.2.2. Example of captured data of healthy variables in other rooms

Simulation Variables			Healthy Mode of the Total System																			
			Room #2				Room #3				Room #4				Room #5				Room #6			
Time (Seconds)	Outside Temp (°C)	Occupants	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status
18136	11.8421	2	19.1591807	1	564.7195257	1	19.44525371	1	550.7150426	0	19.5027828	1	550.1011416	0	19.15918069	1	564.7195257	1	19.44525537	1	550.7150426	0
18137	11.84219	2	19.16318835	1	564.4718098	1	19.45771577	1	550.8073393	0	19.51586798	1	550.1934384	0	19.16318834	1	564.4718098	1	19.45771576	1	550.8073393	0
18138	11.84228	2	19.16719061	1	564.2245897	1	19.4701353	1	550.8996361	0	19.52890784	1	550.2857351	0	19.1671906	1	564.2245897	1	19.47013529	1	550.8996361	0
18139	11.84237	2	19.17118746	1	563.9778645	1	19.48251247	1	550.9919329	0	19.54190255	1	550.3780319	0	19.17118742	1	563.9778645	1	19.48251243	1	550.9919329	0
18140	11.84246	2	19.17517885	1	563.7316333	1	19.49484744	1	551.0842297	0	19.55485229	1	550.4703287	0	19.17517879	1	563.7316333	1	19.49484738	1	551.0842297	0

5.2.3. Example of captured data of the effect of CO_2 sensor fault in room one on the variables of other rooms

Simulation Variables			Effect of the CO_2 Sensor Fault Mode in Room#1 on the other Rooms																			
			Room #2				Room #3				Room #4				Room #5				Room #6			
Time (Seconds)	Outside Temp (°C)	Occupants	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status	Room Temp (°C)	Heater Status	CO_2 Value (PPM)	Damper Status
18136	11.8421	2	**19.1591686**	1	**564.719526**	1	**19.4452537**	1	550.7150426	0	19.5027828	1	550.1011416	0	19.15918069	1	564.7195257	1	19.44525537	1	550.7150426	0
18137	11.84219	2	**19.16316124**	1	**564.47181**	1	**19.45771575**	1	550.8073393	0	19.51586798	1	550.1934384	0	19.16318834	1	564.4718098	1	19.45771576	1	550.8073393	0
18138	11.84228	2	**19.16714257**	1	**564.22459**	1	**19.47013526**	1	550.8996361	0	19.52890784	1	550.2857351	0	19.1671906	1	564.2245897	1	19.47013529	1	550.8996361	0
18139	11.84237	2	**19.17111263**	1	**563.977865**	1	**19.48251238**	1	550.9919329	0	19.54190255	1	550.3780319	0	19.17118742	1	563.9778645	1	19.48251243	1	550.9919329	0
18140	11.84246	2	**19.17507146**	1	**563.731634**	1	**19.49484728**	1	551.0842297	0	19.55485229	1	550.4703287	0	19.17517879	1	563.7316333	1	19.49484738	1	551.0842297	0

5.2.6.RESULTS

This section describes the results acquired from implementing the fault injections in the basic model and the extended version with wireless communications. Figure 5.11 shows the indoor CO_2 concentration based on the occupancy and damper status in fault-free (healthy) and faulty mode of a CO_2 sensor fault. In a fault-free (healthy) system mode (green color signal), the open position of the damper is more frequent in more populated times. As a result, the damper status could remain closed in the rest, and it prevents the coming of low-temperature excess air from the outside into the building (potential energy saving). The frequency of damper switching also depends on the size. It can be seen that in the faulty mode (red color signal), the heater will remain in the on position, and the system will consume more energy .

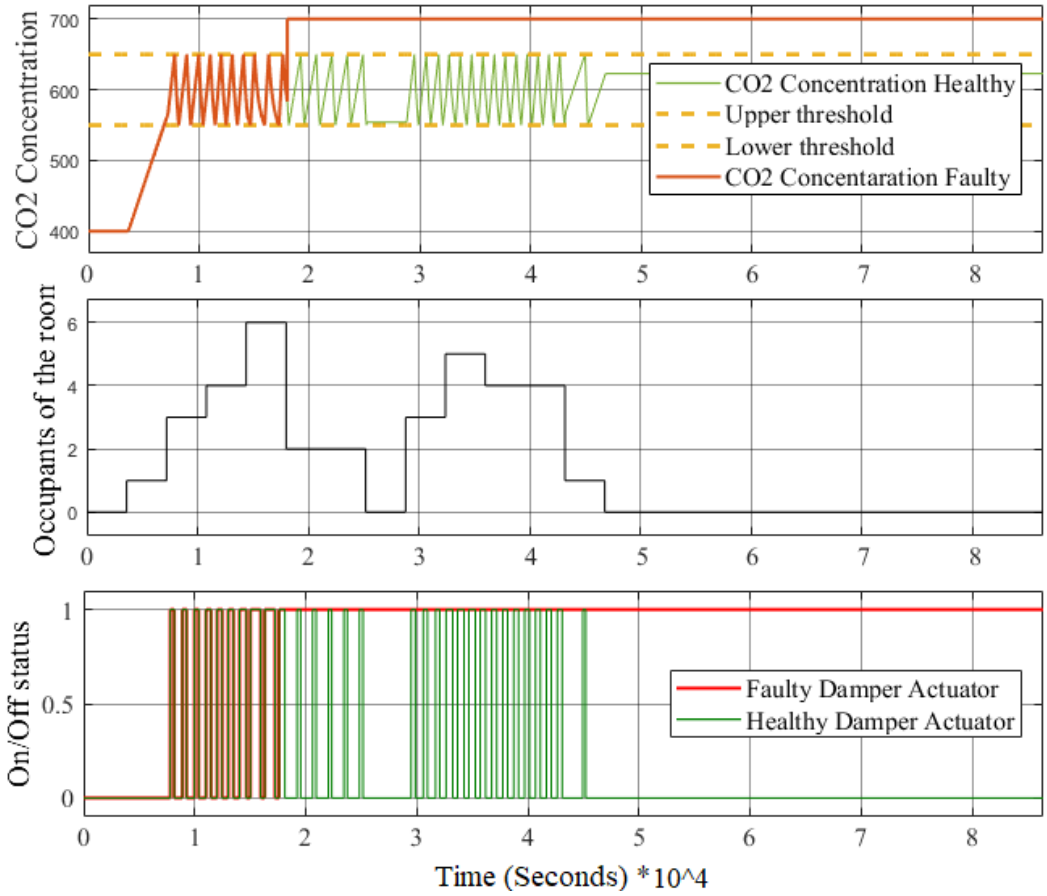


Figure 5.11 CO_2 concentration, occupancy, and damper status.

The system can detect the fault from the system's regular operation and diagnose this fault by specifying the particular cause (faulty component) using the mentioned diagnostic approach. Therefore, the system will trigger an alarm, and the correction work can be done, which comes with better energy efficiency. Figure 5.12, which is a double y-axis figure, shows the room number one temperature variation based on the outside temperature variation, heating system, and damper status in fault-free (healthy) mode (green color) and faulty mode (black color) of the system for different fault cases. Heater and damper are considered to have just two possible statuses, 1 for on and 0 for off in heater, or 1 for open and 0 for close in the damper. The upmost scope in the figure is the system's response when the user manually activates a CO2 sensor fault. The second upmost scope in the figure indicates the damper actuator fault mode, the third upmost scope in the figure shows the temperature sensor fault mode, and the fourth figure describes the heater actuator fault mode.

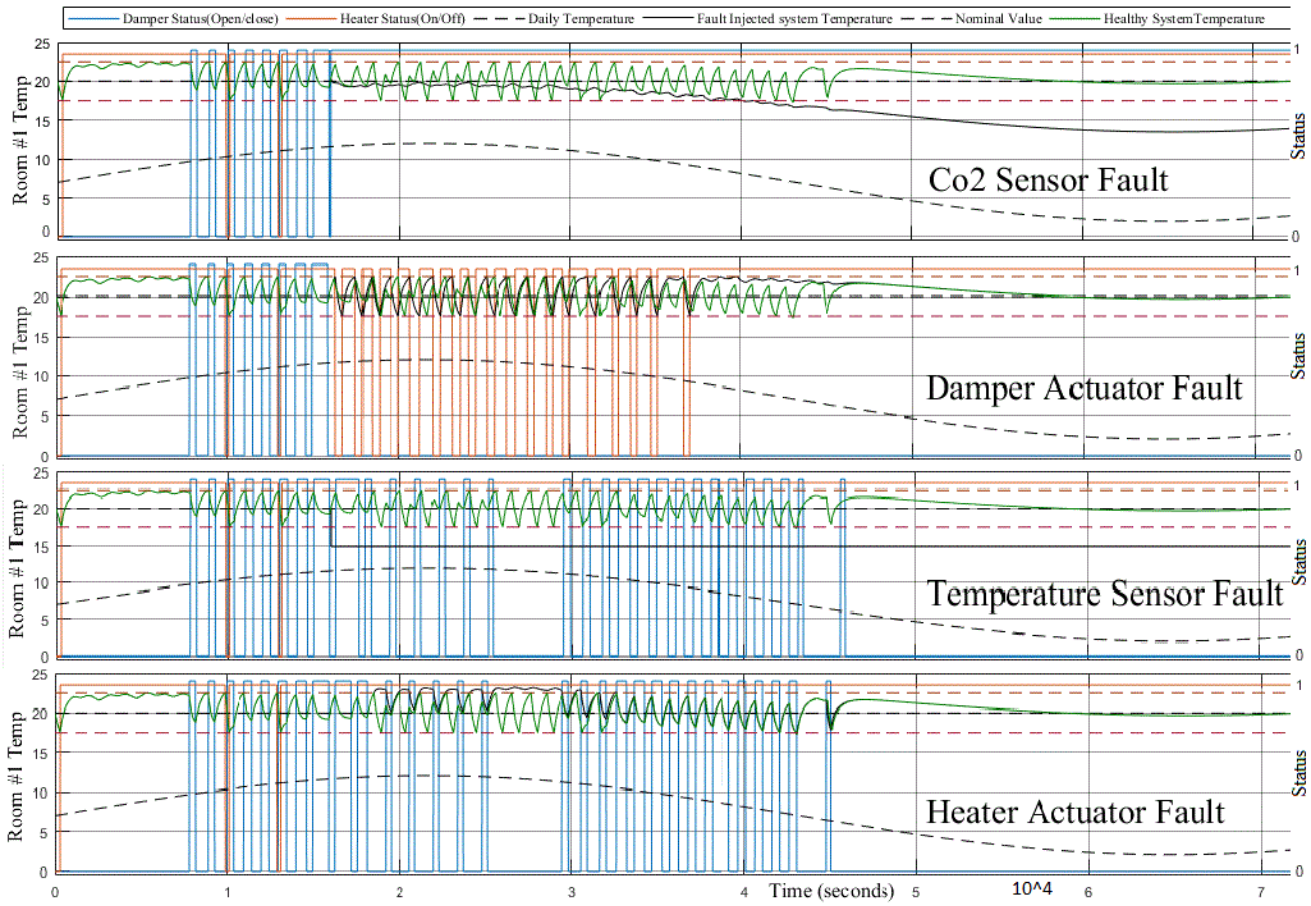


Figure 5.12 Temperature variation in healthy (green) and faulty modes (black).

The results of fault injection in WSAN are described here. The system output, including control parameters and fault injection results, is discussed in this section. The temperature sensor is selected, and the offset fault B gets 5 and 10, for the stuck-at fault of damper actuator B gets 1 which means the damper is continuously open, and n is assumed a random number which describes the effect of noise due to interference or humans in network channel. The sensor battery fault and communication fault (routing loops fault) are described in the previous section. The double-Y axes Figure 5.14 indicates some results for zone one (cluster one) as examples. The simulation describes the system behavior in one day, which is 86,400 seconds (simulation time), but for a better view, the system behavior in the first 50,000 seconds is reported. Figure 5.14(a) describes the behavior of the system in the fault-free (healthy) operation mode in dark blue color, and the rest (Figure 5.14(b) to Figure 5.14(e)) are the faulty behaviors in different fault cases. The room temperature signal as the real room temperature in violet color and faulty sensor readings (in case of fault activation) in light green are also shown. Besides, the heater and damper status in fault-free (healthy) and faulty mode are available for comparison. The results indicate that the user can observe the severity of faults by inserting different values into the fault model (via designed GUI), which was described in equation 1 in this study. Also, this figure indicates that in some cases of fault injection, the system's behavior will be changed; however, the signal value will still be in the desired operating range. For example, in Figure 5.14(d), if the user puts the value of 5 for the B parameter of equation 1, the temperature signal will differ from fault-free (healthy) mode. However, the room temperature mostly remains in the desired thresholds, and in Figure 5.14(e), if the user puts the value of 10, the signal goes beyond the desired range. These differences will be used in future studies as a symptom for predicting the failure, which shows a component, e.g., a sensor can fail shortly if the maintenance operator does not replace or repair it at the right time. Also, Figure 5.13 indicates the designed heater subsystem

and the duty cycle display of the heater that is proportional to the heating energy consumption. The calculation of the duty cycle is for one day (simulation time). The result of this study shows that the heater duty cycle in room one is 43.87 % in fault-free (healthy) (non-faulty) mode. This value will change if any fault is activated, and based on the fault-error-failure propagation model, the rise of fault in one system can affect the other system. Two examples will be discussed for a better understanding. If a fault is activated in the battery of the temperature sensor (see Figure 5.14(b) and compare it with Figure 5.14(a)), the results describe that the heater status will be stuck-at ON, which is proportional to more duty cycle of 48.73% which describes more energy consumption (4.86% more). However, some faults cause also discomfort of occupants, such as activation of offset fault in temperature sensor (see Figure 5.14(d) and Figure 5.14(e) and compare it to Figure 5.14(a)), which shows due to incorrect reporting of temperature which is very high, the heater will stuck-at OFF position as the control system assumes the room temperature is too high, however, in reality, the temperature is not that high. As a result, the duty cycle block displays 19.57%, which is significantly less in comparison to fault-free (healthy) mode (24.3% less), but the occupants experience a discomfort situation because the real room temperature signal in this figure is dropped lower than the comfort zone (desired range). Further, some faults can affect the IAQ parameters such as CO_2 concentration, e.g., when a damper stuck at a close position due to a temperature or CO_2 sensor fault causes the continuous increase in CO_2 concentration. Therefore, scholars or companies can use this framework to find the optimum point of cost and the quality trade-off for their products by monitoring and evaluating the fault effects on the system.

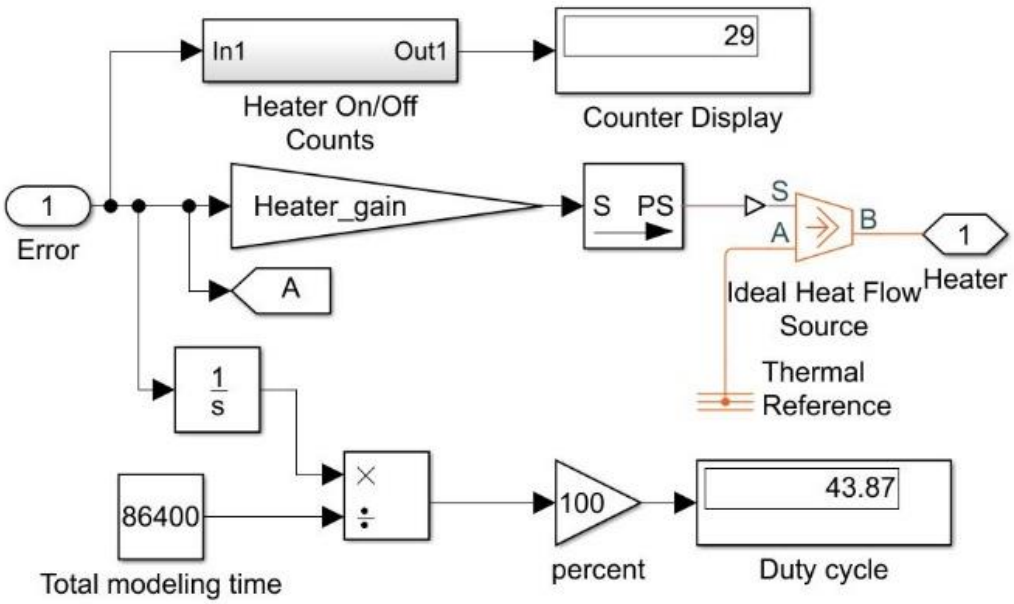


Figure 5.13 Heater Subsystem.

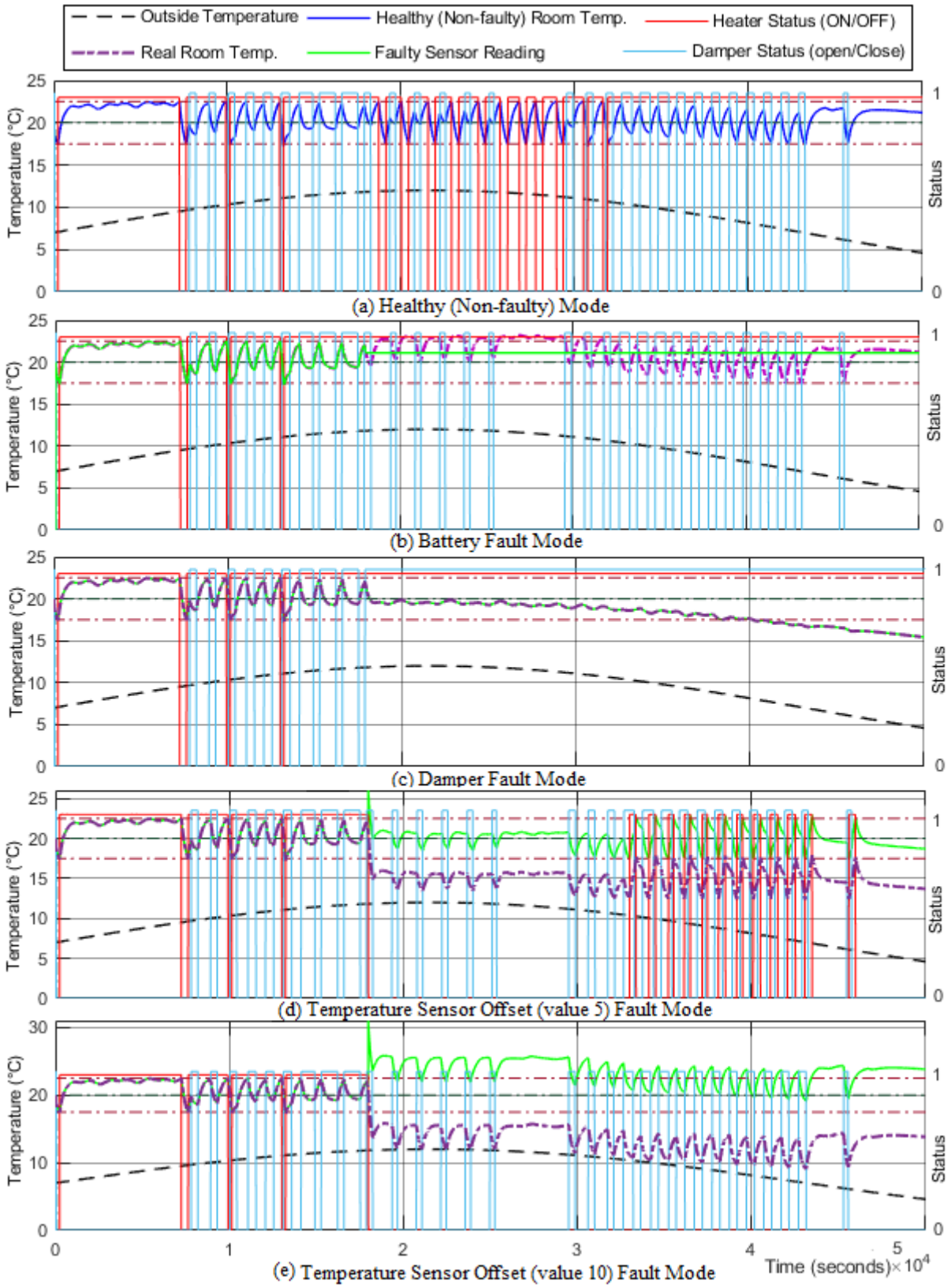


Figure 5.14 Fault Injection Results in WSAN.

6. FAILURE DETECTION AND FAULT DIAGNOSIS METHODOLOGY

The consequences of faults, e.g., in sensors/actuators, may cause significant damage, especially in automated decision and control scenarios. Faults in system components such as sensors and actuators can result in different types of failures and severe implications on the efficiency of DCV and heating systems [24]. The faults that are not detected and diagnosed at the right time can propagate into a component to produce an error. The fault must be isolated to prevent the component or system failure. Otherwise, the error may propagate in the form of a component failure to another component or system and cause a system failure. For this reason, it is necessary to apply fault management techniques to assess and increase the reliability of sensor information and actuator action to guarantee the dependability of the DCV and heating systems, i.e., by decreasing the number of failures. However, tracing the component and system behavior back to the faults is challenging [24]. This thesis conducts the research to study the health monitoring and fault diagnosis techniques on the developed system model for finding the nature, severity, time of occurrence, and locality of faults using a mapping from failures to faults. Therefore, this thesis introduced a failure detection and fault diagnosis framework based on a composed diagnostic classifier by combining data-driven and knowledge-driven diagnosis methods and a failure detection and fault diagnosis framework based on the data-driven multiclass classification using the deep neural networks.

6.1. COMPOSED DIAGNOSTIC CLASSIFIER BASED ON KNOWLEDGE-DRIVEN-BASED AND DATA-DRIVEN-BASED METHODS

This section cuts through the generic description of the combined diagnostic classifier that is generated based on the combination of a data-driven classifier with knowledge-driven inference, e.g., fuzzy logic and BBNs as shown in Figure 6.1. The combination of BBN and fuzzy logic themselves analyze the dependencies of the system signals based on MI theory. Further, the implementation is described based on the example scenario of the DCV and heating system.

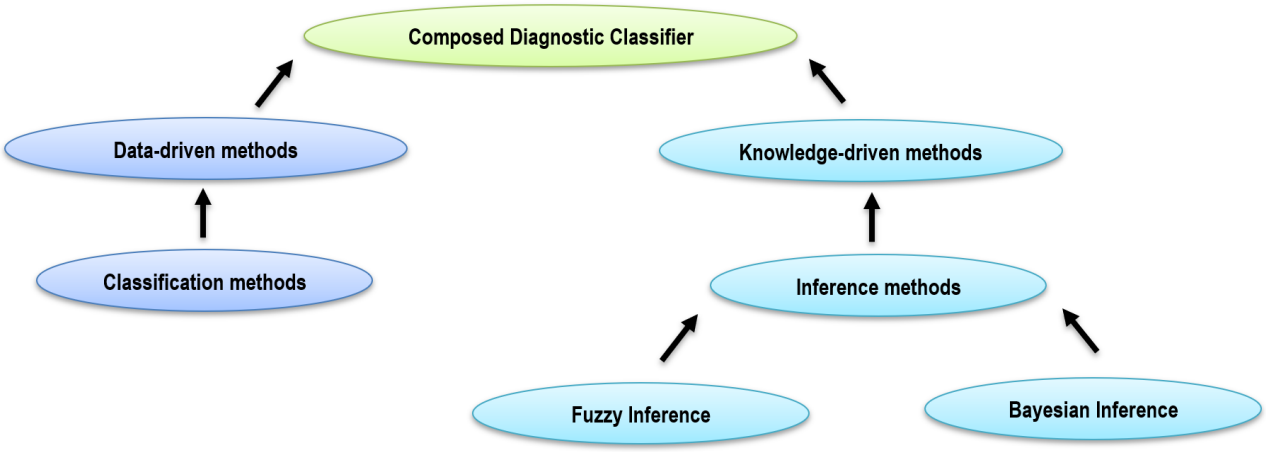


Figure 6.1 Composed diagnostic classifier coverage.

6.1.1. COMPOSED DIAGNOSTIC CLASSIFICATION

The methodology of the generic fault diagnosis method for the diagnosis of any stuck-at or constant faults is based on the causal relations in a fuzzy Bayesian belief network using the relation direction probabilities as described in this section. The result of FBBNs that are causal relations is then visualized using the graphs. The first nine sections (6.1.1.1. to 6.1.1.9) show the generic steps to generate the RDPs that construct the offline fault library. Sections 6.1.1.10. and 6.1.1.11. explains the offline and online modes of fault diagnosis. The graphs are constructed from nodes (indices) and edges (arcs). Figure 6.2 shows the direction of the arcs with the direction of the dependency between each pair of nodes based on the conditional probability values.

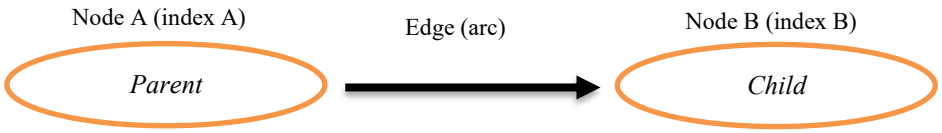


Figure 6.2 Visualized causal relations using the graphs.

Figure 6.3 shows the overview on the steps of generating RDP tables based on FBBN causal relations totally as a part of generic composed fault diagnosis classifier introduced in this thesis. These steps include preparation of the Relational Data Table (RDT), division of the attributes to subdomains by generating Subdomain Label Tables (SLTs), generating Weighted Fuzzy Relational Data Table (WFRDT) for each sample time based on the fuzzy theory, calculation of the joint/intersection probabilities with results in Intersection Triangular Top Matrix (ITTM) and calculation of probability values of subdomains in Subdomain Probability Vector (SPV) table, calculation of the Mutual Information (MI) with results in Subdomains Relation Table (SRT), calculation of the conditional probability of subdomains based on the Bayesian inference with results in Conditional Probability Table (CPT), and ordering elements of the CPT based on the conditional probability values with results in Relation Direction Probability (RDP) table. The next sections describe these steps in detail.

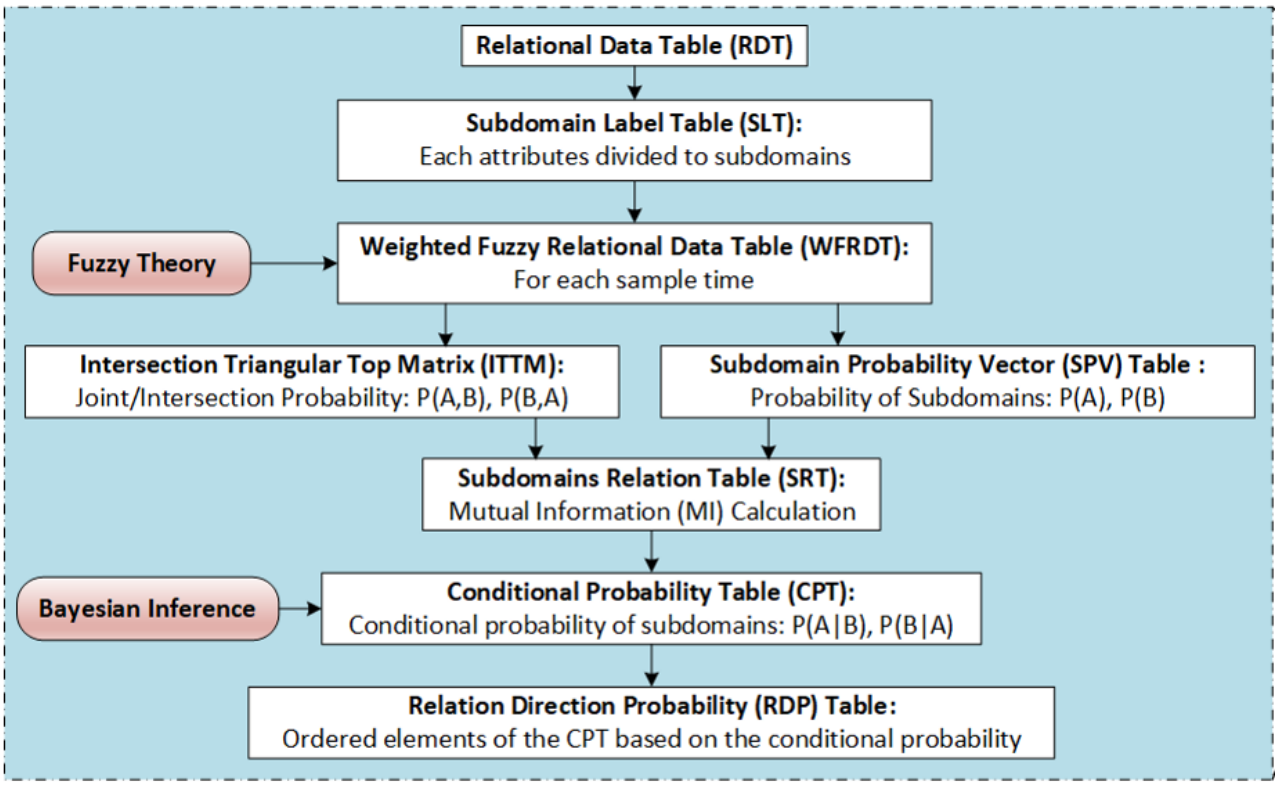


Figure 6.3 Overview of generating RDP tables based on FBBN causal relations.

6.1.1.1. RELATIONAL DATA TABLE (RDT)

In the first step, a Relational Data Table (RDT) of all random variables of the system is created as a basis of the recorded data that must be used for the fault diagnosis. The data samples include information of all variables or attributes as tuples [106]. Table 6.1 shows the overall dataset including samples with their attributes' values over time.

Table 6.1 Relational Data Table (RDT)

Samples	*Attribute*$_1$	*Attribute*$_2$	*Attribute*$_3$	*Attribute*$_i$
S_1	$Value_{11}$	$Value_{12}$	$Value_{13}$	$Value_{1i}$
S_2	$Value_{21}$	$Value_{22}$	$Value_{23}$	$Value_{2i}$
S_n	$Value_{n1}$	$Value_{n2}$	$Value_{n3}$	$Value_{ni}$

In this table, $RDT = \{S_1, S_2, S_3, \ldots, S_n\}$, where S_i is the data sample as a tuple of values for the *i-th* time instance. $S_i = \{Value_{i1}, Value_{i2}, \ldots, Value_{im}\}$, where the values are captured the information for each sample time. Attribute or domain is variable with its measured values over time.

6.1.1.2. SUBDOMAIN LABEL TABLE (SLT)

A domain is a set of ranged values for a variable that this vast range can be divided into small ranges as a set of subdomains. Figure 6.4 shows a sample domain with its subdomains with the values along the Y-axis over time along the X-axis.

$Attribute\text{-}ith = \{Subdomain_1, Subdomain_2, \ldots, Subdomain_p\}$, where subdomains can be a subset of continuous or discrete values.

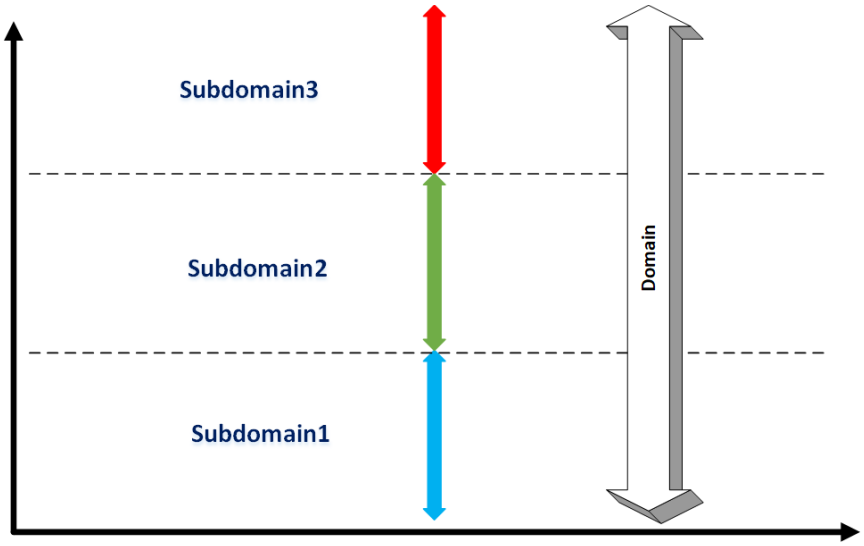


Figure 6.4 An example domain and its subdomains.

In this step, the attributes will be classified into several subdomains. The classification of continuous values can be done using the fuzzy theory [101]. The values with continuous changes will be classified using the fuzzy functions, while those with discrete changes will be classified based on their discrete values. Once all the attributes were classified into subdomains, the Subdomain Label Table (SLT) is generated that includes all attributes and their subdomains. Table 6.2 indicates the SLT. All the subdomains of each attribute will then be saved in the Subdomain Label Vector (SLV). SLV is a vector of all the subdomains extracted from the SLT table.

Table 6.2 Subdomain Label Table (SLT)

No.	Attributes	Subdomains		
1	$Attribute_1$	$Subdomain_{11}$	$Subdomain_{12}$	$Subdomain_{1m}$
2	$Attribute_2$	$Subdomain_{21}$	$Subdomain_{22}$	$Subdomain_{2e}$
i	$Attribute_i$	$Subdomain_{i1}$	$Subdomain_{i2}$	$Subdomain_{if}$

6.1.1.3. WEIGHTED FUZZY RELATIONAL DATA TABLE (WFRDT)

Once the subdomains are defined, the probability of each single sample in a subdomain as weight (W) is calculated based on the Degree of Membership (DM) using the Membership Function (MF). The membership function gets the value and produces the degree of membership values with a range of [0,1]. There are various types of fuzzy membership functions: triangular, trapezoidal, Gaussian, and bell-shaped membership functions. The Trapezoidal has been used in this thesis as follows:

$$Degree\ of\ Membership(x:a,b,c,d) = \begin{cases} 0 & x < a \\ \dfrac{(x-a)}{(b-a)} & a \le x \le b \\ 1 & b \le x \le c \\ \dfrac{(d-x)}{(d-c)} & c \le x \le d \\ 0 & x \ge d \end{cases} \quad (6.1)$$

Table 6.3, Weighted Fuzzy Relational Data Table (WFRDT), shows the DM values or W values extracted from the equation above for different subdomains based on Table 6.1. All weights of all samples for each subdomain are summed up to calculate the total weight for each subdomain. The total weight is placed in the last row of the WFRDT table.

Table 6.3 Weighted Fuzzy Relational Data Table (WFRDT)

	$Attribute_1$			**$Attribute_2$**			**$Attribute_i$**		
No.of Samples	$Subdomain_{11}$	$Subdomain_{12}$	$Subdomain_{1m}$	$Subdomain_{21}$	$Subdomain_{22}$	$Subdomain_{2e}$	$Subdomain_{i1}$	$Subdomain_{i2}$	$Subdomain_{if}$
1	w_{111}	w_{112}	w_{11m}	w_{121}	w_{122}	w_{12e}	w_{1i1}	w_{1i2}	w_{1if}
2	w_{211}	w_{212}	w_{21m}	w_{221}	w_{222}	w_{22e}	w_{2i1}	w_{2i2}	w_{2if}
n	w_{n11}	w_{n12}	w_{n1m}	w_{n21}	w_{n22}	w_{n2e}	w_{ni1}	w_{ni2}	w_{nif}
Total Weight	$W_{11}=\sum_{111}^{n11}W$	$W_{12}=\sum_{112}^{n12}W$	$W_{1m}=\sum_{11m}^{n1m}W$	$W_{21}=\sum_{121}^{n21}W$	$W_{22}=\sum_{122}^{n22}W$	$W_{2e}=\sum_{12e}^{n2e}W$	$W_{i1}=\sum_{1i1}^{ni1}W$	$W_{i2}=\sum_{1i2}^{ni2}W$	$W_{if}=\sum_{1if}^{nif}W$

6.1.1.4. MUTUAL INFORMATION AND SUBDOMAIN PROBABILITY VECTOR (SPV)

MI is a statistical measure rooted in information and probability theory. MI of two random variables is a statistical measurement of the mutual dependence of two random variables [106]. MI measures information regarding one random variable by observing the other one [32]. There are many definitions of random variables. For example, G. Zeng [33] has classified MI definitions into two categories: 1) definitions with random variables and 2) ensembles. The fuzzy theory can be used an appropriate likelihood density function [37]. Intan et al. [106] defined the MI between two fuzzy sets of *A* and *B* as follows:

$$MI(A,B) = MI(B,A) = P(A,B)\log_2\left(\frac{P(A,B)}{P(A)\times P(B)}\right) \tag{6.2}$$

Where $P(A)\neq 0$ and $P(B)\neq 0$. *P(A)* and *P(B)* are the probability values of fuzzy sets *A* and *B,* and *P(A,B)* is the joint probability value of fuzzy sets *A* and *B* or the intersection between *A* and *B*. The equation above determines a correlation measure and if the value of one variable is known, an amount of information (-or- mutual information) from the other variable will be also known, i.e., *MI(A,B)>0* refers to a positive correlation that describes the fuzzy sets *A* and *B* have a mutual dependency, then knowing *A* gives information about *B*; and *MI(A,B)≤0* describes that *A* and *B* are independent, then knowing *A* does not give any information about *B*.

$$P(A) = \frac{\sum_{k=1}^{n} A(d_{kj})}{n} \quad and \quad P(B) = \frac{\sum_{k=1}^{n} B(d_{ki})}{n} \tag{6.3}$$

Where *R* is the number of records and $A(d_{kj}), B(d_{ki}) \in [0,1]$ are membership degrees of d_{kj} and d_{ki} for fuzzy sets *A* and *B*, respectively [106]. The probability of the subdomains, i.e., *P(A), P(B),* therefore can be calculated by the total weights and using the equation below. Where *n* is the number of samples (No. of Records) and W_{if} is the total weight for the $subdomain_{if}$. Table 6.4 shows the Subdomain Probability Vector (SPV).

$$P(Subdomain_{if}) = \frac{\sum_{k=1}^{n} W_{if}}{n} \tag{6.4}$$

Table 6.4 Subdomain Probability Vector Table (SPV)

	$Attribute_1$			$Attribute_2$			$Attribute_i$		
Subdomains	$Subdomain_{11}$	$Subdomain_{12}$	$Subdomain_{1m}$	$Subdomain_{21}$	$Subdomain_{22}$	$Subdomain_{2e}$	$Subdomain_{i1}$	$Subdomain_{i2}$	$Subdomain_{if}$
Probability of Subdomain	$P_{11}=\frac{W_{11}}{n}$	$P_{12}=\frac{W_{12}}{n}$	$P_{1m}=\frac{W_{1m}}{n}$	$P_{21}=\frac{W_{21}}{n}$	$P_{22}=\frac{W_{22}}{n}$	$P_{2e}=\frac{W_{2e}}{n}$	$P_{i1}=\frac{W_{i1}}{n}$	$P_{i2}=\frac{W_{i2}}{n}$	$P_{if}=\frac{W_{if}}{n}$

6.1.1.5. INTERSECTION TRIANGULAR TOP MATRIX (ITTM)

The joint probability between two subdomains is calculated in this section. A joint probability is a statistical measure for two events occurring at the same time instance. If event *A* changes the probability of event *B*, then they are dependent; otherwise, they are independent. The probability value for the independent events is equal to zero. Therefore, the dependent events are determined. In this thesis, to calculate the dependent subdomains, the algorithm compares the fuzzy weights of

each time sample in pairs for the first pair of subdomains and calculates the mean of minimum weight values of the pair over the whole sample times. This process will be repeated over all subdomains and over all time samples. Finally, the algorithm calculates the intersection (joint) probability of that pair (of subdomains), respectively. The algorithm generates a triangular top matrix of intersection (joint) probabilities of subdomains called Intersection Triangular Top Matrix (ITTM). Therefore, if *P(A,B)* is the probability of two subsets *A* and *B*, the Intersection probability is calculated using the following equation [106]:

$$P(A,B) = P(A \cap B) = \frac{\sum_{k=1}^{n} \min\left(A(d_{kj}), B(d_{ki})\right)}{n} = \frac{\sum_{k=1}^{n} \min\left(W_{Subdomain_{kj}}, W_{Subdomain_{ki}}\right)}{n} \quad (6.5)$$

Here $W_{Subdomain kj}$ and $W_{Subdomain ki}$ are the fuzzy weights of subdomains *A* and *B, respectively*. Note that *P(A,B)* is equal to *P(B,A)*. Therefore, this is not essential to measure both the top and down triangular. In this chapter, we only consider the values of the top triangular matrix. Table 6.5 indicates the Intersection Triangular Top Matrix (ITTM).

Table 6.5 Intersection Triangular Top Matrix (ITTM)

Subdomains \ Subdomains	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	i-1	i
1																		
2									P(2,9)									
3																		
4																		
5																		
6																		
7												P(7,12)						
8																		
9																		
10																		
11																		
12																		
13															P(13,15)			
14																		
15																		
...																		
i-1																		
i																		

6.1.1.6. SUBDOMAINS RELATION TABLE (SRT)

After calculation of *P(A), P(B),* and *P(A,B)* for every subdomain pair, equation (6.2) will get the value for the MI. The positive MI value shows a pairwise dependency between two subdomains of a pair which shows two subdomains contain mutual information, and the negative MI value indicates a pairwise independency. The fault diagnosis classifier in this thesis assumes the binary value of 1 for the positive MI values and the binary value of 0 for the negative MI values. The binary results will then be placed in a top triangular matrix named Subdomains Relation Table (SRT), as shown in Table 6.6.

Table 6.6 Subdomains Relation Table (SRT)

Subdomains \ Subdomains	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...	i
1																		
2																		
3								1				1					1	
4										1								
5																		
6														1				
7																		
8											1							
9															1		1	
10																		
11																		
12																		
13																1		
14																		
15																		
16																		
...																		
i																		

6.1.1.7. CONDITIONAL PROBABILITY TABLE (CPT)

In this step, the conditional probabilities for only pairs that had positive mutual information are measured. The conditional probability of fuzzy event *A* given *B* based on the Bayesian inference is denoted *by P(A|B)* [106]. In other works in the literature, the conditional probability indicates the posterior probability of a fault given observed fault symptoms which requires expert knowledge; however, in this thesis, it indicates the direction of dependency of different subdomains (independent from expert knowledge). The left side of the equation below is the posterior probability and the right side includes prior conditional probability and prior probability values.

$$P(A \mid B) = \frac{P(A,B)}{P(B)} = \frac{\sum_{k=1}^{n} \min\left(A(d_{kj}), B(d_{ki})\right)}{\sum_{k=1}^{n} B(d_{ki})} \tag{6.6}$$

This $P(A|B)$ corresponds to $P(subdomain_i \mid subdomain_j)$ in the method described in this thesis.

$$P(subdomain_{ki} \mid subdomain_{kj}) = \frac{\sum_{1}^{n} \min\left(W_{Subdomain_{kj}}, W_{Subdomain_{ki}}\right)}{\sum_{1}^{n} W_{Subdomain_{ki}}} \tag{6.7}$$

Also, the conditional probabilities of fuzzy event *B* given *A* are denoted by $P(B|A)$ [106].

$$P(B \mid A) = \frac{P(B,A)}{P(A)} = \frac{\sum_{k=1}^{n} \min\left(A(d_{kj}), B(d_{ki})\right)}{\sum_{k=1}^{n} A(d_{kj})} \tag{6.8}$$

This conditional probability corresponds to $P(subdomain_j \mid subdomain_i)$ in the method described in this thesis.

$$P(subdomain_{kj} \mid subdomain_{ki}) = \frac{\sum_{1}^{n} \min\left(W_{Subdomain_{kj}}, W_{Subdomain_{ki}}\right)}{\sum_{1}^{n} W_{Subdomain_{kj}}} \quad (6.9)$$

The results from the above equations are later will be stored in a matrix called Conditional Probabilities Table (CPT) with the following rules:

- P(A|B)>P(B|A) indicates the direction of dependency between A and B is from B to A. Then, P(B|A) will be eliminated, and P(A|B) will be stored in Table 6.7.
- P(B|A)>P(A|B) indicates the direction of dependency between A and B is from A to B. Then, P(A|B) will be eliminated, and P(B|A) will be stored in Table 6.7.

In Table 6.7, the highlighted elements of the matrix with the yellow color show the conditional probability of each pair of subdomains, e.g., *P(Subdomain9|Subdomain2)* and *P(Subdomain2|Subdomain9)*. Then, the conditional probability with the greater value will be kept and saved in the CPT table as the green elements of the matrix, and the conditional probability with the lower value will be deleted from the CPT table.

Table 6.7 Conditional Probabilities Table (CPT)

Subdomains \ Subdomains	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1																		
2									P(Subdomain2 | Subdomain9)									
3																		
4																		
5																		
6																		
7																		
8																		
9		P(Subdomain9 | Subdomain2)																
10																		
11																		
12																		
13																		
14																		
15																		
16																		
17																		
18																		

6.1.1.8. RELATION DIRECTION PROBABILITY (RDP)

In this section, the Relation Direction Probability (RDP) table as shown in Table 6.8 which is the ordered result from the CPT indicates all the relations and their features, i.e., the direction between dependent subdomains and the conditional probability of the transmission. The RDP table includes the parents' and childs' columns generated based on the subdomains and the conditional probabilities of these pairs listed in the CPT table.

Table 6.8 Relation Direction Probability (RDP)

Number of relations	*Parents*	*Childs*	*Conditional Probabilies*
1	$Subdomain_i$	$Subdomain_j$	$P(Subdomain_j \mid Subdomain_i)$
2	$Subdomain_k$	$Subdomain_w$	$P(Subdomain_w \mid Subdomain_k)$
n	$Subdomain_n$	$Subdomain_m$	$P(Subdomain_m \mid Subdomain_n)$

All existed elements in the CPT table are ordered in this table in which the conditional probability of $P\ (Subdomain_j \mid Subdomain_i)$ presents that there is a relation between $Subdomain_j$ and $Subdomain_i$, and the direction is from $Subdomain_i$ (Parent Node) to $Subdomain_j$ (Child Node) with the probability of $P(Subdomain_j \mid Subdomain_i)$.

6.1.1.9. Causal Relation in FBBN using the Relation Direction Probabilities

As mentioned, the FBBN shows the causal relations between each pair of subdomains that are extracted from the RDP table. The conditional probabilities indicate the direction of the dependency in each pair of nodes. Figure 6.5 is an example that shows $Subdomain_j$ (Child Node) is related to $Subdomain_i$ (Parent Node).

Figure 6.5 FBBN Causal relations based on the Relation Direction Probabilities

6.1.1.10. Fault Diagnosis Classification based on Causal Relations in FBBNs

This thesis shows that the FBBN causal relations based on the RDPs can determine all the cause-effect relationships among every subdomain in case of faults. This capacity is used to diagnose stuck-at fault types for several components with various stuck-at values at different time instances. This research considers the constant faults because the constant values will be in a subdomain from a whole range of values for a parameter. Therefore, the diagnosis method can relate the faulty value with the fault. In this section, an overview of the overall methodology is described.

This diagnosis method has two modes: the online mode and the offline mode. The offline mode includes the generation phase of the reference libraries, including various faulty conditions (fault objects) for diagnosing different real fault cases with random/real faulty values. Figure 6.6 describes the schematic of the FDFD strategy of this thesis based on the causal relations in a fuzzy Bayesian belief network using the relation direction probabilities.

This strategy includes two main modules of failure detection and fault diagnosis within two parallel parts of offline training and online failure detection and fault diagnosis. First, the model inputs are inserted into the real plant at runtime and into the plant simulation equipped with the fault injection blocks to be preprocessed and create faulty and fault-free databases. Data from the fault-free database (offline training mode) and the output of the real plant (online failure detection and fault diagnosis) of this strategy are fed through the fault detection module to check whether the residuals are within the thresholds. The alarm reports failures based on the residuals. Residuals are the deviations of the model parameters at a specific range of operating conditions as an indication of failure. Once the failure is detected and announced by the alarm, the fault diagnosis module will be activated, and it starts with the creation of the real case object (*RealCase*) and its specification, including the RDP table.

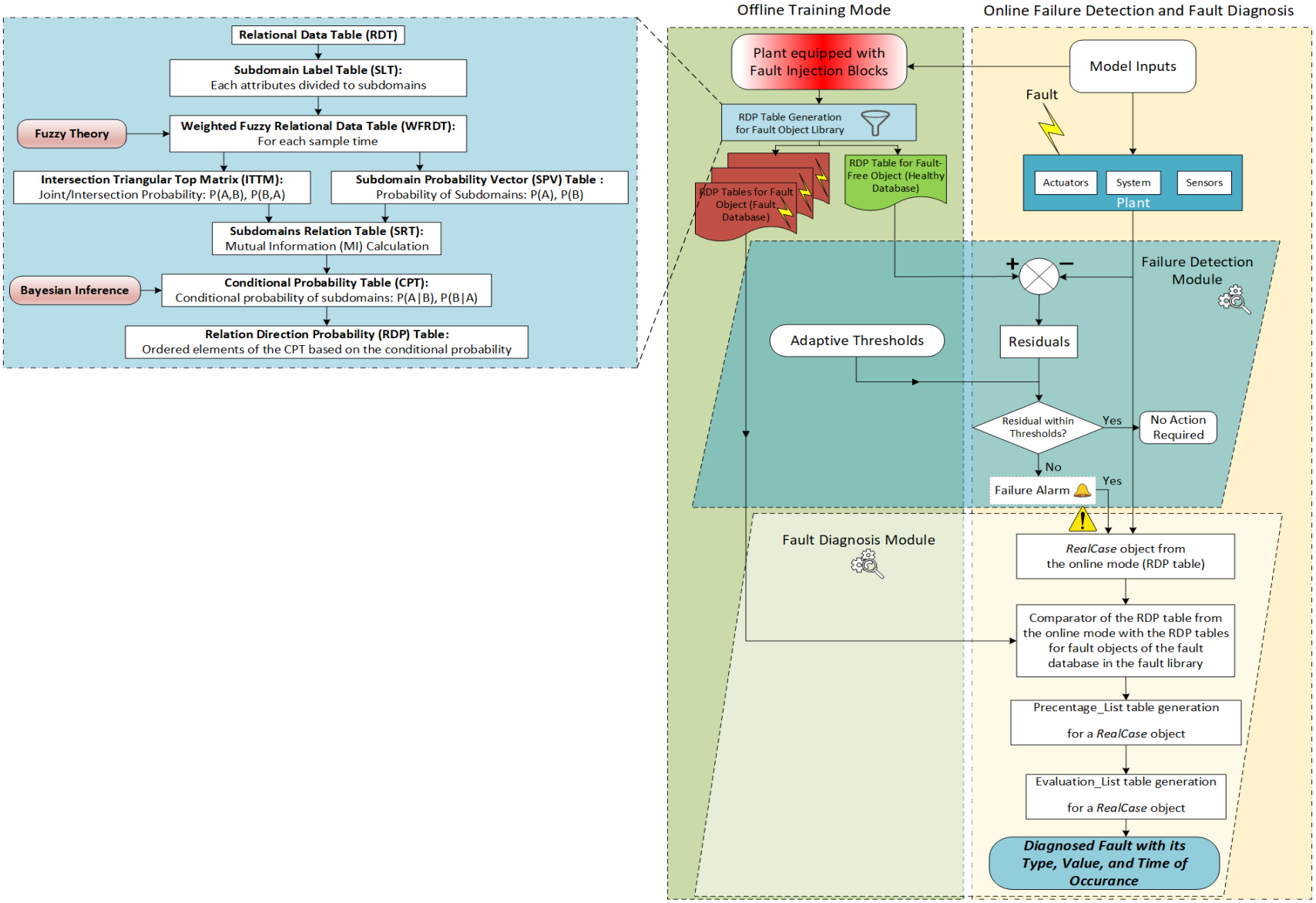


Figure 6.6 Schematic of the FDFD strategy of this thesis based on the causal relations in a fuzzy Bayesian belief network using the relation direction probabilities.

6.1.1.11. OFFLINE TRAINING MODE

The offline training mode includes a library of different fault modes. In this library, a fault object for each type of fault is created that is named: *Fault_Object$_i$, i= {1, n}*. The *n* value depends on the total number of fault types (*p*) and the time instances vector (*t*). Table 6.9 shows an overview of the offline library of the fault cases. The fault object is defined as a class in MATLAB as Function 1 that contains four properties: type of fault (Type), time of fault injection (Time), the system parameters' values for each fault case (Data), and the RDP table. This library is generated for every fault type that includes all subdomains of a domain and for different fault injection times. To include all the subdomains, a representative for each subdomain and for each time interval is defined. The time interval is a vector of time values for the total time domain.

Function 1. Fault Class Definition

```
classdef Fault
  properties
   Type;
   Time;
   Data;
   RDP;
  end
end
```

Table 6.9 Offline Library of Fault Cases

No. of Faults	1	2	3	n-1	n
Objects for Different Fault cases	$Fault\_Object_1$	$Fault\_Object_2$	$Fault\_Object_3$	$Fault\_Object_{n-1}$	$Fault\_Object_n$

6.1.1.12. ONLINE DIAGNOSTIC MODE

The online mode represents the real case scenario simulated in this research using the fault injection framework. Time, type, and value in this fault injection are the properties of the real scenario with random values within a range. The RDP table in the online mode contains all the relations of subdomains of the real case (*RealCase*). For the diagnosis of the real case, the *RealCase* diagnosis class is defined in Function 2 with three properties, including Type, Time, and Value of random case. This class also includes two more properties: Precentage_List and Evaluation_List.

Function 2. RealCase Diagnosis Class Definition

```
classdef RealCase Diagnosis
  properties
   Type;
   Time;
   Value;
   Precentage_List; // the size of this list is equal to n or the number of fault cases in the offline library
   Evaluation_List; // The size of this list is equal to x
   end
end
```

In the diagnosis, the RDP table of the *RealCase* is compared to the RDP tables available in the fault library (Offline Mode's Library). In this comparison, the relations' specifications in the RDP table of *RealCase* object, i.e., parent nodes, child nodes, and the pairs' directions, are compared to the respective values in the RDP table of each fault object in the offline library. Therefore, the fault diagnosis classifier can calculate the number of common pairs and the percentages of similarity. The percentages of similarity for each *RealCase* can be formulated as below:

$$Percentage\ of\ Similarity_i = \frac{Number\ of\ similar\ Pairs\ in\ both\ RDP\ Tables}{Total\ Number\ of\ Pairs\ in\ Fault\ Case_i} \times 100, \quad 1 < i < n \qquad (6.10)$$

The result from the equation above for *fault case*$_i$ will be stored in the *element*$_i$ of a list called Precentage_List. This list includes *n* number of percentages of similarity for a *RealCase* compared to the *n* fault objects in the offline library. Table 6.10 shows the Precentage_List for a *RealCase* object. After that, the *x* number of top-ranked similar fault cases based on the highest percentages in the Precentage_List will be distinguished by the fault diagnosis classifier as the most relevant results, stored in the Evaluation_List.

Table 6.10 Precentage_List for a *RealCase* object

No. of Fault Object in the Offline Library	**1**	**2**	**…**	**i**
Calculated Percentage of Similarity between the RealCase fault object and each Fault Object in the Offline Fault Library	Percentage of Similarity$_1$	Percentage of Similarity$_2$	…	Percentage of Similarity$_i$

The Evaluation list in Table 6.11 consists of *x* elements, and every element *j (1<j<x)* has three properties: fault type and value, fault time, and percentage. The type and the time are allocated from the *fault case*$_i$ in the offline library. The percentage is the percentage of similarity that is actually the belief and the higher the fault belief means the higher the possibility of the corresponding fault. The diagnosis algorithm sorts the evaluation list by the rank order of values in the percentage column. "ranking" in statistics is the data transformation that the number or order of values are justified by their rank when they are sorted. Therefore, the elements with the higher percentages of similarity will be placed at the top ranks, and the lower percentages of similarity will be placed at the lower ranks.

Table 6.11 Evaluation_List for a *RealCase* object

No.	**Type**	**Time**	**Percentage**
1	Offline_FaultType$_1$	Offline_FaultTime$_1$	Highest_Precentage$_1$
2	Offline_FaultType$_2$	Offline_FaultTime$_2$	Highest _Precentage$_2$
3	Offline_FaultType$_3$	Offline_FaultTime$_3$	Highest _Precentage$_3$
j	Offline_FaultType$_j$	Offline_FaultTime$_j$	Highest _Precentage$_j$

The type and time of element *j* of this list are the type and time of the *fault case*$_i$ object in the offline library. Finally, the comparison results of the type and the time values of the Evaluation_List with the type and the time of the *RealCase* object which can determine the belief of the fault diagnosis method in this thesis based on the causal relation in fuzzy Bayesian belief network using the relation direction probabilities.

6.1.2. IMPLEMENTATION OF COMPOSED DIAGNOSTIC CLASSIFICATION BASED ON EXAMPLE SYSTEM MODEL

As mentioned in section 6.1.1.10.3, the RDP tables are the essential data required for the online diagnosis phase. This section describes an example scenario for the introduced diagnostic method in section 6.1.1. for a use-case of a DCV and heating system in MATLAB/Simulink. Here, an example fault type is selected in the DCV and heating system model's fault injection framework to show the diagnosis methodology's detailed description. The selected fault type is the CO_2 sensor fault with a stuck-at value of 700 ppm with the fault injection time of 18.000 seconds. The simulation is run for 86400 seconds or one typical winter day.

6.1.2.1. RELATIONAL DATA TABLE (RDT) IN SYSTEM MODEL

The Data Preparation step includes extracting the data from signal values of Simulink-Model and initializing the RDT. An output from the system model in MATLAB/Simulink prepares the relational data table required for the introduced diagnosis methodology in this chapter. An example RDT based on the DCV and heating system model is shown in Table 6.12. The attributes can be domains with continuous or discrete values. In this table, the samples for 86400 seconds or one-day simulation time can be recorded. Therefore, RDT = $\{S_1, S_2, S_3, \ldots, S_{86400}\}$.

Table 6.12 Use-Case Relational Data Table (RDT) for CO_2 Sensor Fault in 18000 seconds

Seconds (samples)	Daily Temperature	Occupancy number	Room temperature	Room CO_2 Concentration	Heater Status	Damper Status
1	7.00	0	19.99	400	0	0
2	7.00	0	19.98	400	0	0
3	7.00	0	19.97	400	0	0
4	7.00	0	19.96	400	0	0
5	7.00	0	19.95	400	0	0
…	…	…	…	…	…	…
17999	11.82	6	19.55	579.06	1	1
18000	11.82	6	19.55	700	1	1
18001	11.82	6	19.55	700	1	1
18002	11.82	6	19.55	700	1	1
18003	11.82	2	19.55	700	1	1
…	…	…	…	…	…	…
86399	6.99	0	16.85	700	1	1
86400	7.00	0	16.85	700	1	1

6.1.2.2. SUBDOMAINS LABEL TABLE (SLT) IN SYSTEM MODEL

Attributes and subdomains preparation step includes defining the labels of attributes and subdomains of the system for analysis and defining fuzzy sets over these attributes. For defining attributes, they are firstly divided into two types of continuous and discrete. The continuous attributes have continuous changes in their values, such as room temperature, daily temperature, room CO_2 concentration, and occupancy parameters. The discrete attributes are heater status and damper status. There is also a simulation clock as a discrete attribute required for the evaluation step.

Once the continuous and discrete attributes and subdomains were defined in the previous section, the SLT is created that is shown in Table 6.13. The subdomains are also named nodes. For example, subdomains for the continuous daily temperature attribute are three subdomains of low daily temperature, middle daily temperature, and high daily temperature. These subdomains are used to create the fuzzy sets for the fuzzy function. Eighteen subdomains based on seven attributes can facilitate the conditional probability measurement of the BBN. The subdomain index is considered as a reference to the subdomain title. Figure 6.7 shows three example subdomains for the attribute of room temperature.

Table 6.13 Use-Case Subdomain Label Table (SLT)

No.	Attributes	Subdomains	Subdomains	Subdomains
1	Daily Temperature	Low_Daily_Temperature (No. 1)	Middle_Daily_Temperature (No.2)	High_Daily_Temperature (No.3)
2	Occupants Number	Low_Occupancy (No.4)	Normal_Occupancy (No.5)	High_Occupancy (No.6)
3	Room Temperature	Lower_than_Threshold_RoomTemperature (No.7)	Within_Threshold_RoomTemperature (No.8)	Upper_than_Threshold_RoomTemperature (No.9)
4	Heater Status	Heater_Status_On (No.10)	Heater_Status_Off (No.11)	---------
5	Damper Status	Damper_Status_Open (No.12)	Damper_Status_Close (No.13)	---------
6	Simulation Clock	Healthy_Mode (No.14)	Faulty_Mode (No.15)	---------
7	Room CO2 Concentration	Lower_than_Threshold_CO2Value (No.16)	Within_Threshold_CO2Value (No.17)	Upper_than_Threshold_CO2Value (No.18)

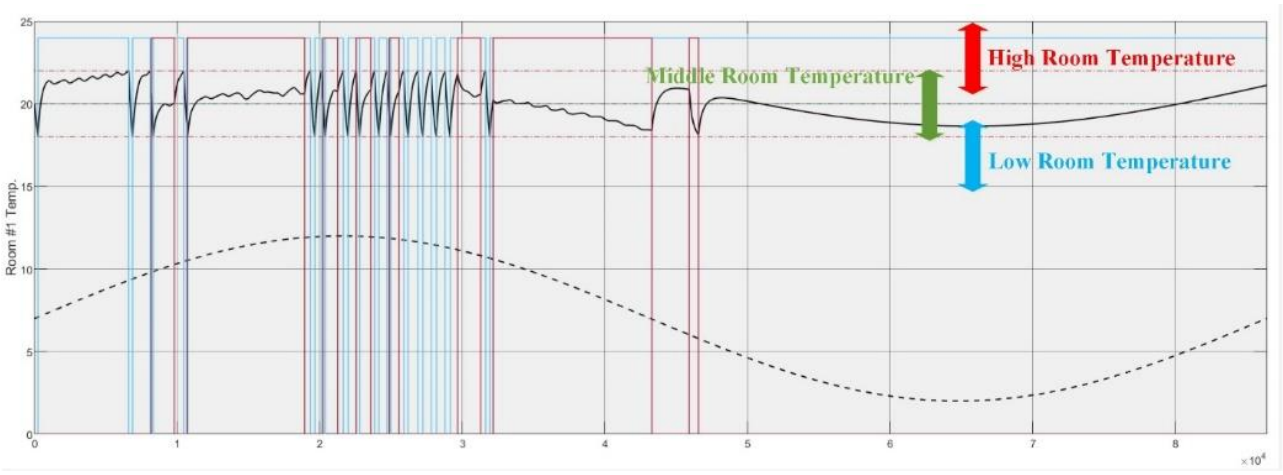


Figure 6.7 Illustration of Subdomains and fuzzy sets Definitions for Room Temperature.

Fuzzy Rules in System Model

In this section, fuzzy rules or membership functions are defined to generate the probability weights based on the degree of membership values from the fuzzy function. Therefore, the conditional probability of each subdomain will be measured by the membership degree of each sample of the subdomain. The fuzzy membership functions are explained here that get the degree of membership value for each x input value.

- **Room Temperature Fuzzy Membership Function**

The total range of the room temperature has been considered between [0-40]. This range can be divided into three subdomains of low room temperature with [0-17.5] range, middle room temperature with [17.5-22.5] range, and high room temperature with [22.5-40] range. Twenty degrees of centigrade is the nominal temperature value of the system. Equations (17-19) show the

fuzzy membership functions respective to each subdomain. Figure 6.8 shows the overall room temperature fuzzy membership function with the related fuzzy function of each subdomain in three various colors of blue for lew temperature values, green for middle-temperature values, and red for high-temperature values.

$$Low\_Room\_Temperature\_Fuzzy\_Membership\_Function(x)=\begin{cases}1 & x\le 17.5\\ \dfrac{(19.5-x)}{(19.5-17.5)} & 17.5\le x\le 19.5\\ 0 & x\ge 19.5\end{cases} \quad (6.11)$$

$$Middle\_Room\_Temperature\_Fuzzy\_Membership\_Function(x)=\begin{cases}0 & x<17.5\\ \dfrac{(x-17.5)}{(19.5-17.5)} & 17.5\le x\le 19.5\\ 1 & 19.5\le x\le 20.5\\ \dfrac{(22.5-x)}{(22.5-20.5)} & 20.5\le x\le 22.5\\ 0 & x\ge 22.5\end{cases} \quad (6.12)$$

$$High\_Room\_Temperature\_Fuzzy\_Membership\_Function(x)=\begin{cases}0 & x<20.5\\ \dfrac{(x-20.5)}{(22.5-20.5)} & 20.5\le x\le 22.5\\ 1 & x\ge 22.5\end{cases} \quad (6.13)$$

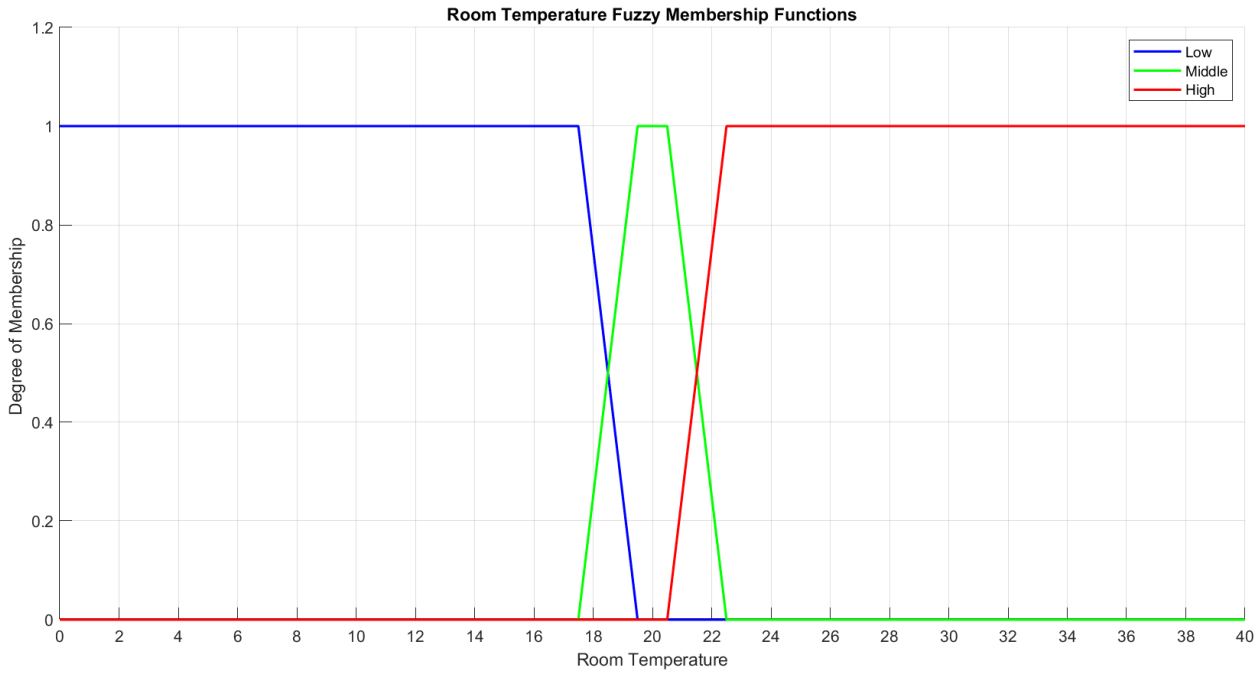


Figure 6.8 Room Temperature Fuzzy Membership Functions

- **Daily Temperature Fuzzy Membership Function**

The total range of the daily temperature has been considered between [0-14]. This range is divided into three subdomains the low daily temperature with [0-5] range, middle daily temperature with [5-9] range, and high daily temperature with [9-14] range. Below the fuzzy membership functions are shown, and their illustration is in Figure 6.9.

$$Low\_Daily\_Temperature\_Fuzzy\_Membership\_Function(x)=\begin{cases}1 & x\le 5\\ \dfrac{(6.825-x)}{(6.825-5)} & 5\le x\le 6.825\\ 0 & x\ge 6.825\end{cases} \quad (6.14)$$

$$Middle\_Daily\_Temperature\_Fuzzy\_Membership\_Function(x)=\begin{cases}0 & x<5\\ \dfrac{(x-5)}{(6.825-5)} & 5\le x\le 6.825\\ 1 & 6.825\le x\le 7.175\\ \dfrac{(9-x)}{(9-7.175)} & 7.175\le x\le 9\\ 0 & x\ge 9\end{cases} \quad (6.15)$$

$$High\_Daily\_Temperature\_Fuzzy\_Membership\_Function(x)=\begin{cases}0 & x<20.5\\ \dfrac{(x-7.175)}{(9-7.175)} & 7.175\le x\le 9\\ 1 & x\ge 9\end{cases} \quad (6.16)$$

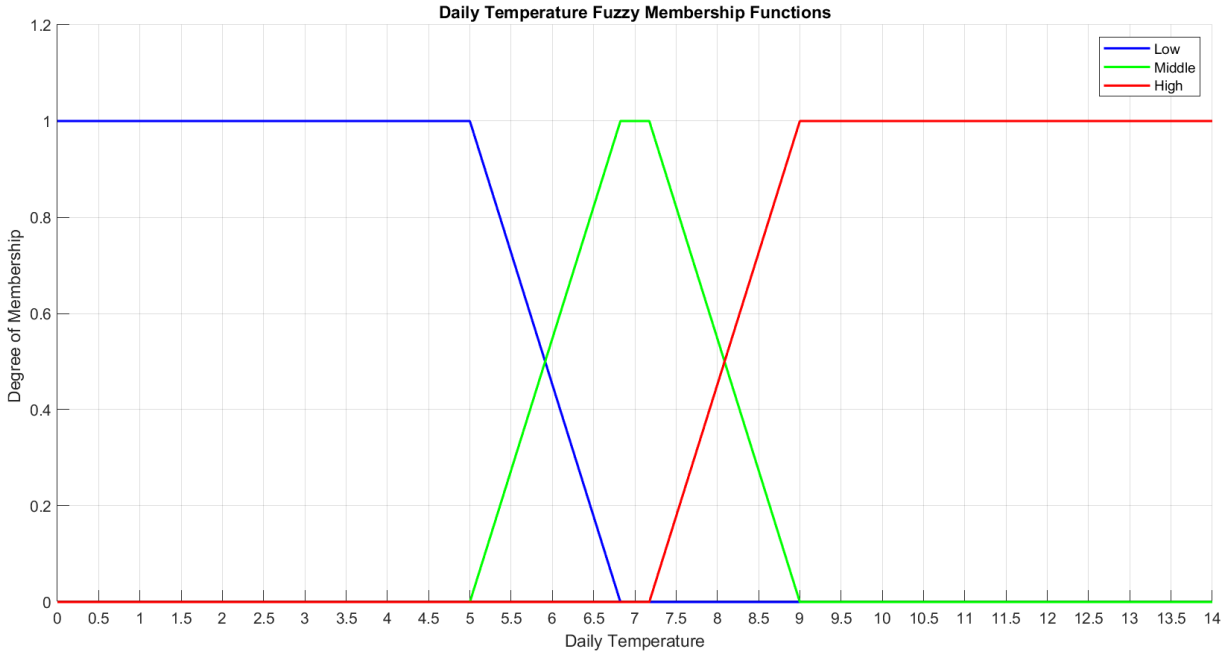


Figure 6.9 Daily Temperature Fuzzy Membership Functions

- **CO_2 Concentration Fuzzy Membership Function**

The CO_2 concentration values are considered in a range of [0-1200]. The CO_2 concentration variable is divided into three subdomains of the low CO_2 concentration with [0-400] range, middle CO_2 concentration with [400-800] range, and CO_2 concentration with [800-1200] range. Figure 6.10 illustrates the fuzzy membership function of the CO_2 concentration.

$$Low\_CO_2\_Concentration\_Fuzzy\_Membership\_Function(x)=\begin{cases}1 & x\le 400\\ \dfrac{(585-x)}{(585-400)} & 400\le x\le 585\\ 0 & x\ge 585\end{cases} \quad (6.17)$$

$$Middle\_CO_2\_Concentration\_Fuzzy\_Membership\_Function(x) = \begin{cases} 0 & x < 400 \\ \dfrac{(x-400)}{(585-400)} & 400 \le x \le 585 \\ 1 & 585 \le x \le 615 \\ \dfrac{(800-x)}{(800-615)} & 615 \le x \le 800 \\ 0 & x \ge 800 \end{cases} \quad (6.18)$$

$$High\_CO_2\_Concentration\_Fuzzy\_Membership\_Function(x) = \begin{cases} 0 & x < 615 \\ \dfrac{(x-615)}{(800-615)} & 615 \le x \le 800 \\ 1 & x \ge 800 \end{cases} \quad (6.19)$$

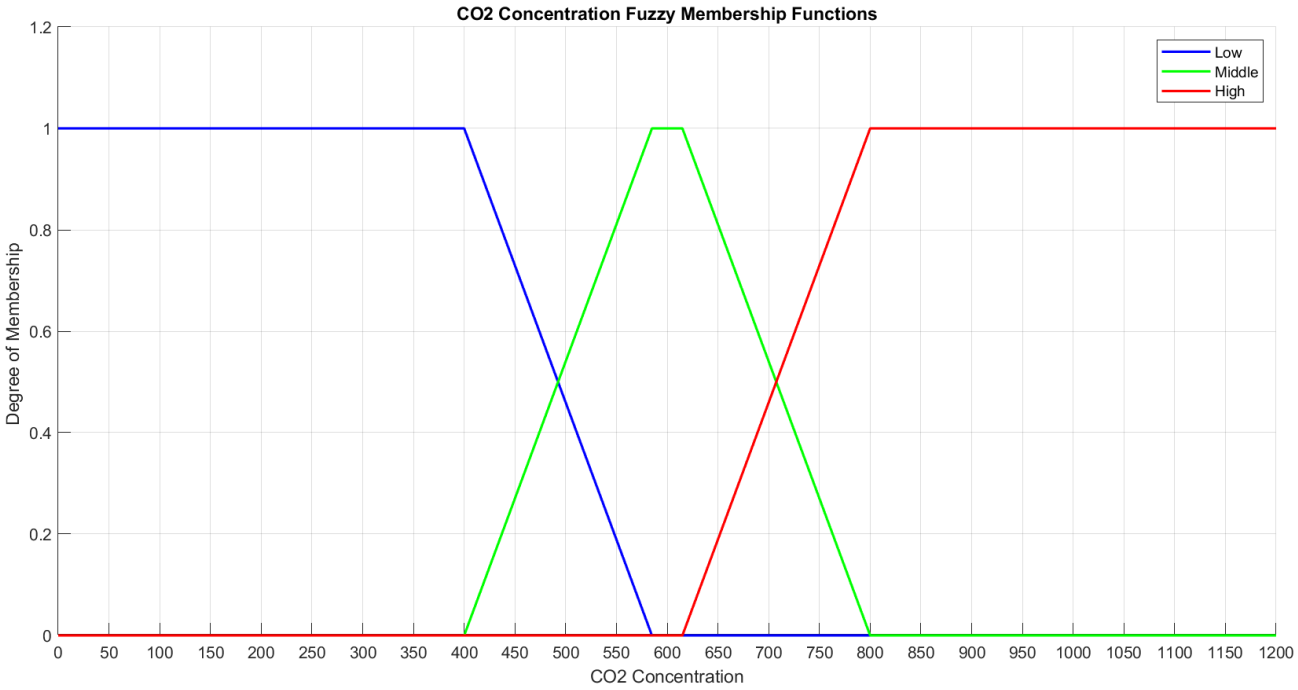


Figure 6.10 CO_2 Concentration Fuzzy Membership Functions

- **Occupancy Fuzzy Membership Function**

The number of occupants in this system model differs in a range of [0-6] people based on an example scenario. The occupancy with less than three people is considered low occupancy, equal to and between three and four people as the middle occupancy, and equal to and between five and six people represents high occupancy in a day. These variations are illustrated in the fuzzy membership degree functions in equations below and Figure 6.11.

$$Low\_Occupancy\_Fuzzy\_Membership\_Function(x) = \begin{cases} 1 & x \le 2 \\ \dfrac{(3-x)}{(3-2)} & 2 \le x \le 3 \\ 0 & x \ge 3 \end{cases} \quad (6.20)$$

$$Middle\_Occupancy\_Fuzzy\_Membership\_Function(x) = \begin{cases} 0 & x < 2 \\ \frac{(x-2)}{(3-2)} & 2 \le x \le 3 \\ 1 & 3 \le x \le 4 \\ \frac{(5-x)}{(5-4)} & 4 \le x \le 5 \\ 0 & x \ge 5 \end{cases} \quad (6.21)$$

$$High\_Occupancy\_Fuzzy\_Membership\_Function(x) = \begin{cases} 0 & x < 4 \\ \frac{(x-4)}{(5-4)} & 4 \le x \le 5 \\ 1 & x \ge 5 \end{cases} \quad (6.22)$$

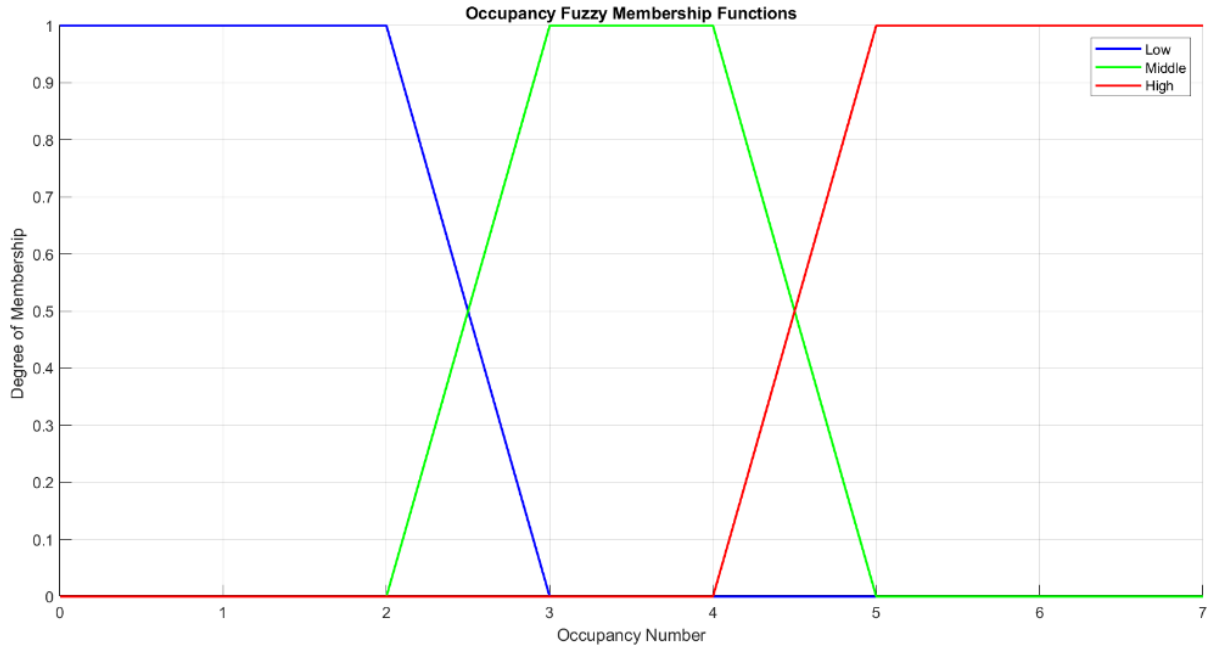


Figure 6.11 Occupancy Fuzzy Membership Functions

6.1.2.3. Weighted Fuzzy Relational Data Table (WFRDT) in System Model

In this step, a WFRDT based on RDT is generated that is shown in Table 6.14. For this, a matrix of 86401 rows and 18 columns is created. An example fault type is the CO_2 sensor fault with a stuck-at value of 700 ppm with an example fault injection time of 18.000 seconds is considered. Every table column is a subdomain and contains fuzzy weights for continuous attributes and normal weights of occurrence for discrete attributes. For each subdomain, the fuzzy weights based on the output from each sample's degree of membership functions are measured. Each sample is measured data in every second of simulation time. In the last row of this table, the total weights over one column are calculated.

Table 6.14 Use-Case Weighted Fuzzy Relational Data Table (WFRDT) for CO_2 Sensor Fault

Attributes	Daily Temperature			Occupancy			Room Temperature			Heater Status		Damper Status		Simulation Clock		Room CO_2 Concentration		
Subdomains / No. Samples	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	1	0	1	0	0	0	1	0	0	1	0	1	1	0	1	0	0
2	0	1	0	1	0	0	0	1	0	0	1	0	1	1	0	1	0	0
3	0	1	0	1	0	0	0	1	0	0	1	0	1	1	0	1	0	0
4	0	1	0	1	0	0	0	1	0	0	1	0	1	1	0	1	0	0
5	0	1	0	1	0	0	0	1	0	0	1	0	1	1	0	1	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
17999	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0.03	0.96	0
18000	0	0	1	0	0	1	0	1	0	1	0	1	0	0	1	0	0.54	0.45
18001	0	0	1	1	0	0	0	1	0	1	0	1	0	0	1	0	0.54	0.45
18002	0	0	1	1	0	0	0	1	0	1	0	1	0	0	1	0	0.54	0.45
18003	0	0	1	1	0	0	0	1	0	1	0	1	0	0	1	0	0.54	0.45
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
86399	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0.54	0.45
86400	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0.54	0.45
Total Weight	37133.79	12132.41	37133.79	61199	18001	7200	53381.54	24902.60	8115.84	85676	724	73790	12610	18000	68400	6038.38	48643.65	31717.95

6.1.2.4. MUTUAL INFORMATION AND SUBDOMAIN PROBABILITY VECTOR (SPV) IN SYSTEM MODEL

As mentioned in section 6.1.1.4, the probability of the subdomains can be measured by the total weights for each subdomain divided by several samples using the equation (6.4). Therefore, the Subdomain Probability Vector (SPV) is calculated, and the result is shown in the table below.

Table 6.15 Use-Case Subdomain Probability Vector (SPV) in Percent for CO_2 Sensor Fault

Attribute	*Daily Temperature*			*Occupancy*			*Room Temperature*			*Heater Status*		*Damper Status*		*Simulation Clock*		*Room CO_2 Concentration*		
Subdomain	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
Probability	*42.98*	*14.04*	*42.98*	*70.83*	*20.83*	*8.33*	*61.78*	*28.82*	*9.39*	*99.16*	*0.84*	*85.41*	*14.59*	*20.83*	*79.17*	*6.99*	*56.30*	*36.71*

6.1.2.5. INTERSECTION TRIANGULAR TOP MATRIX (ITTM) IN SYSTEM MODEL

In this step, a triangular top matrix is generated to store the intersection probabilities of subdomain pairs. The intersection probability of $P(A,B)=P(B,A)$. Therefore, it makes sense to calculate the probabilities upper than the main diagonal of the matrix. Therefore, the following matrix with 18 rows and 18 columns using equation (6.9).

Table 6.16 Use-Case Intersection Triangular Top Matrix (ITTM) for CO_2 Sensor Fault (in Percent)

Subdomains / Subdomains	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	0.99	0	42.98	0	0	42.98	0	0	42.98	0	42.98	0	0	42.98	0	24.72	21.23
2	0	0	2.99	10.53	3.51	0	10.76	2.04	2.49	13.6	0.35	10.53	3.51	3.51	10.53	3.51	8.18	7.07
3	0	0	0	17.32	17.32	8.33	8.28	28.03	7.92	42.49	0.49	31.90	11.08	17.32	25.66	3.74	28.12	12.87
4	0	0	0	0	0	0	50.69	14.20	5.94	70.48	0.35	62.50	8.33	8.33	62.50	6.46	35.65	28.72
5	0	0	0	0	0	0	9.17	8.84	2.82	20.35	0.49	15.83	5.01	8.33	12.50	0.30	14.55	5.99
6	0	0	0	0	0	0	1.93	5.78	0.63	8.33	0	7.08	1.25	4.17	4.17	0.23	6.10	2.01
7	0	0	0	0	0	0	0	4.76	0	61.53	0.26	61.49	0.29	1.07	60.71	0.50	35.66	29.98
8	0	0	0	0	0	0	0	0	3.05	28.35	0.48	23.41	5.41	10.37	18.45	2.67	20.58	10.77
9	0	0	0	0	0	0	0	0	0	9.29	0.10	0.50	8.89	9.39	0	4.67	5.26	0.31
10	0	0	0	0	0	0	0	0	0	0	0	85.12	14.04	20.00	79.17	6.64	55.87	36.66
11	0	0	0	0	0	0	0	0	0	0	0	0.28	0.56	0.84	0	0.35	0.43	0.05
12	0	0	0	0	0	0	0	0	0	0	0	0	0	6.24	79.17	0.33	48.57	36.50
13	0	0	0	0	0	0	0	0	0	0	0	0	0	14.59	0	6.66	7.73	0.21
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.99	13.51	0.34
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	42.79	36.37
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.66	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	36.71
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.1.2.6. SUBDOMAINS RELATION TABLE (SRT) IN SYSTEM MODEL

In this section, a top triangular matrix is generated based on the MI calculations named SRT to find the correlation between the subdomains of a pair in Table 6.17. For the pairs that have a positive amount of correlation measurement, the matrix element gets the value 1, and for the negative result of the MI equation, it gets the binary value of 0. As mentioned in section 4.6, the positive values indicate the relation or dependency between the subdomains of a pair (pairwise correlation) [32].

Table 6.17 Use-Case Subdomains Relation Table (SRT) for CO_2 Sensor Fault

Subdomains / *Subdomains*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
1	*0*	*0*	*0*	*1*	*0*	*0*	*1*	*0*	*0*	*1*	*0*	*1*	*0*	*0*	*1*	*0*	*1*	*1*
2	*0*	*0*	*0*	*1*	*1*	*0*	*1*	*0*	*1*	*0*	*1*	*0*	*1*	*1*	*0*	*1*	*1*	*1*
3	*0*	*0*	*0*	*0*	*1*	*1*	*0*	*1*	*1*	*0*	*1*	*0*	*1*	*1*	*0*	*1*	*1*	*0*
4	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*0*	*1*	*0*	*1*	*0*	*0*	*1*	*1*	*0*	*1*
5	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*1*	*0*	*1*	*0*	*1*	*1*	*0*	*0*	*1*	*0*
6	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*1*	*0*	*0*	*1*	*1*	*0*	*0*	*1*	*0*
7	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*1*	*0*	*0*	*1*	*0*	*1*	*1*
8	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*1*	*0*	*1*	*1*	*0*	*1*	*1*	*1*
9	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*1*	*1*	*0*	*1*	*0*	*0*
10	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*0*	*1*	*0*	*1*	*1*
11	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*1*	*0*	*1*	*0*	*0*
12	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*1*	*1*
13	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*1*	*0*	*0*
14	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*1*	*0*
15	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*
16	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
17	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*
18	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*

6.1.2.7. CONDITIONAL PROBABILITY TABLE (CPT) IN SYSTEM MODEL

The correlation among subdomains of each pair yields the direction of the transition arcs by calculation the conditional probability value between the pairs. In This step, every conditional probability of each pair of *A* and *B* as *P(A/B)* and *P(B|A)* are calculated. For this, a Top-Down triangular matrix including all the conditional probabilities of all corresponding subdomains for each pair is generated that is named CPT shown in Table 6.18. Then, the conditional probabilities for each pair of subdomains from both sides from *A* to *B* and from *B* to *A* are compared, and the direction with the corresponding probability with a higher probability value is kept. The direction with the corresponding probability with a lower probability value is eliminated. For example, the conditional probability value of 0.8315 that is the conditional probability of $subdomain_3$ given $subdomain_5$ (*P(*$subdomain_3$/$subdomain_5$*)=0.8315*), was more significant than the conditional probability of $subdomain_5$ given $subdomain_3$ that is already eliminated and its value is replaced with 0.

Table 6.18 Use-Case Conditional Probabilities Table (CPT) for CO_2 Sensor Fault (In Percent)

Subdomains / Subdomains	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	57.83
2	0	0	0	0	0	0	0	0	26.53	0	41.71	0	0	0	0	50.24	0	0
3	0	0	0	0	83.15	100	0	97.24	84.30	0	58.29	0	75.94	83.15	0	53.49	0	0
4	100	75	0	0	0	0	82.04	0	0	0	0	0	0	0	0	92.47	0	78.22
5	0	25	0	0	0	0	0	0	30.06	0	58.29	0	34.31	40.01	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	100	76.59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	63.33	81.65
8	0	0	0	0	42.44	69.31	0	0	32.45	0	56.87	0	37.05	49.78	0	38.23	0	0
9	0	0	0	0	0	0	0	0	0	0	12.49	0	0	0	0	66.80	0	0
10	100	0	0	99.51	0	100	99.58	0	0	0	0	99.67	0	0	100	0	99.23	99.85
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	100	0	0	88.24	0	0	99.52	0	0	0	0	0	0	0	100	0	86.27	99.43
13	0	25	0	0	0	15.04	0	0	94.67	0	66.44	0	0	0	0	95.26	0	0
14	0	25	0	0	0	49.99	0	0	100	0	100	0	100	0	0	100	0	0
15	100	0	0	88.24	0	0	98.27	0	0	0	0	0	0	0	0	0	0	99.08
16	0	0	0	0	0	0	0	0	0	0	41.71	0	0	0	0	0	0	0
17	57.51	58.22	65.42	0	69.82	73.19	0	71.40	0	0	0	0	0	64.84	0	0	0	100
18	0	50.32	0	0	0	0	0	37.35	0	0	0	0	0	0	0	0	0	0

6.1.2.8. Relation Direction Probability (RDP) in System Model

In this step, the Relation Direction Probability table is based on the filtered results in the CPT matrix. The RDP table shows the direction of the dependency between subdomains of each pair from the parent node to the child node and respective conditional probability values as a result for the example CO_2 Sensor Fault is shown in Table 6.19.

Table 6.19 Use-Case Relation Direction Probability (RDP) for CO_2 Sensor Fault

No. of Relations	*Parents*	*Childs*	*Conditional Probabilies*
1	*Upper_than_Threshold_CO2Value*	*Low_Daily_Temperature*	57.82
2	*Upper_than_Threshold_RoomTemperature*	*Middle_Daily_Temperature*	26.53
3	*Heater_Status_Off*	*Middle_Daily_Temperature*	41.71
4	*Lower_than_Threshold_CO2Value*	*Middle_Daily_Temperature*	50.23
5	*Normal_Occupancy*	*High_Daily_Temperature*	83.15
6	*High_Occupancy*	*High_Daily_Temperature*	100
7	*Within_Threshold_RoomTemperature*	*High_Daily_Temperature*	97.24
8	*Upper_than_Threshold_RoomTemperature*	*High_Daily_Temperature*	84.29
9	*Heater_Status_Off*	*High_Daily_Temperature*	58.28
10	*Damper_Status_Close*	*High_Daily_Temperature*	75.94
…	…	…	…
70	*Upper_than_Threshold_CO2Value*	*Within_Threshold_CO2Value*	100
71	*Middle_Daily_Temperature*	*Upper_than_Threshold_CO2Value*	50.31
72	*Within_Threshold_RoomTemperature*	*Upper_than_Threshold_CO2Value*	37.35

6.1.2.9. Causal Relation in FBBN using the Relation Direction Probabilities in System Model

As mentioned in section 4.9, the FBBN shows the causal relations between each pair of subdomains extracted from the RDP table. Figure 6.12 is an example that shows Subdomain of High_Daily_Temperature (Child Node) is related to Subdomain of Normal_Occupancy (Parent Node) with the conditional probability value of 0.8295.

Figure 6.12 An Example FBBN Causal relations based on the Relation Direction Probabilities.

6.1.2.10. Offline Mode of the Fault Diagnosis in System Model

This section shows the implementation result of the diagnosis method introduced in this thesis for fault diagnosis and evaluation of the proposed method for the DCV and heating system model. The fault diagnosis phase is constructed from two offline and online modules. The evaluation approach is the same with Figure 6.6; however, a sample RealCase fault object with a known RDP table, fault type, and fault type is considered in the evaluation phase. In the real and normal operation,

only the RDP table is available and is generated, and the fault type and the fault time are the diagnosis result. In the offline mode of fault diagnosis, a library of fault cases is created. The offline library includes ten types of faults stored in a vector called *Fault_Injection_Type_Vector*, and 17 time instances for the fault injection stored in a vector called *Fault_Injection_Time_Vector*. Therefore, the offline library has 170 injected fault objects for all mentioned fault types and injection times combinations.

Function 3. Offline Mode Fault Injection Type and Time Vectors

```
Fault_Injection_Type_Vector=["CO2SensorLow","CO2SensorMiddle","CO2SensorHigh","DamperActuatorOff","Da
mperActuatorOn","TemperatureSensorLow","TemperatureSensorMiddle","TemperatureSensorHigh",
"HeaterActuatorOff","HeaterActuatorOn"];// This vector has 10 elements

Fault_Injection_Time_Vector = 5000:5000:86400; // This vector has 17 elements
```

For each injected $fault\_case_i$, the Type, Time, Data, and RDP values will then be stored in the i-th element of the Offline Library, where $1 > i > 170$. The implemented offline library of the example system model is depicted as below:

Table 6.20 Implemented offline library of 170 fault cases in the DCV and heating system

No. of Fault Object	***Fault_object1***		***Fault_object2***		***Fault_object3***		**…**	***Fault_object169***		***Fault_object170***	
Details	Type	*"CO2SensorLow"*	*Type*	*"CO2SensorMiddle"*	*Type*	*"CO2 Sensor High"*	…	*Type*	*"HeaterActuator Off"*	*Type*	*"Heater Actuator On"*
	Time	*5000*	*Time*	*5000*	*Time*	*5000*	…	*Time*	*85000*	*Time*	*85000*
	Data	*86400 x10 double*	*Data*	*86400x10 double*	*Data*	*86400 x10 double*	…	*Data*	*86400x10 double*	*Data*	*86400x10 double*
	RDP	*144x3 string*	*RDP*	*144x3 string*	*RDP*	*144x3 string*	…	*RDP*	*144x3 string*	*RDP*	*144x3 string*

6.1.2.11. ONLINE MODE OF THE FAULT DIAGNOSIS IN SYSTEM MODEL

In the online mode of the diagnosis method, a random *RealCase* fault object with its specifications, i.e., Type, Time, Value, is injected into the system, the Precentage_List, and Evaluation_List are stored as an example scenario of the reality to test and monitor the response of the diagnosis method introduced in this chapter. For this, a fault type from the *Fault_Injection_Type_Vector* with a random time of fault occurrence and its value is selected.

Function 4. Online Mode Fault Injection Type Vector

```
Fault_Injection_Type_Vector=["CO2Sensor","DamperActuator","TemperatureSensor","HeaterActuator"];
```

Function 5. Random Fault Injection using *Randi* Function

```
RealTime_Fault = randi(86400);

switch Fault_Mode
case "CO2Sensor"
CO2_FaultInjectionValue = randi([300,850],1);
case "DamperActuator"
Damper_ FaultInjectionValue = randi([1,2],1)-1;

case "TemperatureSensor"
TempSensor_ FaultInjectionValue = randi([10,30],1);
case "HeaterActuator"
Heater_ FaultInjectionValue = randi([1,2],1)-1;
otherwise
end
```

A *RealCase* example has shown in Table 21. In this *RealCase* object example, a *stuck_at_off* fault mode in the heater actuator at 70393 seconds is simulated as a fault injection time.

Table 6.21 *RealCase* object Example for the DCV and heating system

RealCase_Object$_1$	
Type	*"Heater_Actuator"*
Time	*70393*
Value	*0*
Precentage_List	*170x1 double*
Evaluation_List	*20x3 string*

As mentioned in section 6.1.1.12, the Percentage_List shows the percentage of similarity. The generation of the Evaluation_List is the final step of fault diagnosis. The diagnosis method introduced in this thesis can diagnose the fault with its type, value, and time of fault occurrence. For this, the Evaluation_List is created based on comparing the percentages of similarity in the Precentage_List of the *RealCase* Example with every fault case available in the offline library. The Percentage_List table is eliminated in this thesis as it includes all the percentage of similarity values with every 170 fault objects available in the fault library. The fault diagnosis classifier sorts the evaluation list by the values in the percentage column. Therefore, the elements with the higher percentages of similarity will be placed at the top ranks, and the lower percentages of similarity will be placed at the lower ranks. Finally, the comparison results of the type and the time values of the Evaluation_List with the type and the time of the *RealCase* object which was initiated from the fault injection can determine the accuracy of the fault diagnosis method in this thesis based on the causal relations in fuzzy Bayesian belief network using the relation direction probabilities. In the example test scenario, the top 20 highest ranks of diagnosed fault cases have been considered for the fault diagnosis, and the first row of the table below is the diagnosis result with the highest rank having the same type and time of the injected fault the *RealCase* Example. It is also clear that the diagnosed time of fault occurrence is very close to the actual value with an excellent estimation. A more exact result is possible with an enormous offline library with more example fault scenarios for more fault

injections time samples. The Evaluation_List table for the *RealCase* example is shown in Table 6.22.

Table 6.22 Evaluation_List Table for the *RealCase* Example

No.	*Type in offline Library*	*Time in offline Library*	*Percentage in Precentage_List*	
1	*HeaterActuatorOff*	*70000*	*51.38*	→ The Diagnosed Case in the First Rank
2	*HeaterActuatorOff*	*65000*	*50*	→ Second Rank
3	*HeaterActuatorOff*	*60000*	*49.30*	→ Third Rank
4	*HeaterActuatorOff*	*75000*	*49.30*	
5	*HeaterActuatorOff*	*55000*	*47.91*	→ Fourth Rank
6	*HeaterActuatorOff*	*50000*	*45.83*	→ Fifth Rank
7	*HeaterActuatorOff*	*45000*	*45.13*	
8	*TemperatureSensorLow*	*70000*	*45.13*	
9	*HeaterActuatorOff*	*80000*	*45.13*	
10	*TemperatureSensorLow*	*65000*	*43.75*	
11	*TemperatureSensorHigh*	*70000*	*43.75*	
12	*TemperatureSensorLow*	*75000*	*43.75*	
13	*TemperatureSensorLow*	*60000*	*43.05*	
14	*TemperatureSensorHigh*	*65000*	*42.36*	
15	*TemperatureSensorHigh*	*75000*	*42.36*	
16	*HeaterActuatorOff*	*40000*	*41.66*	
17	*TemperatureSensorLow*	*55000*	*41.66*	
18	*TemperatureSensorHigh*	*60000*	*41.66*	
19	*TemperatureSensorHigh*	*55000*	*40.97*	
20	*TemperatureSensorLow*	*80000*	*40.97*	

6.2. DATA-DRIVEN-BASED FAULT DIAGNOSIS BASED ON MULTICLASS CLASSIFICATION

This thesis provides also a pure data-driven fault diagnosis method that is a data-driven fault classifier to evaluate the performance of the composed diagnostic classifier, which is the combination of the data-driven and knowledge-driven diagnostic methods, introduced in section 6.1.

6.2.1. SYSTEM ARCHITECTURE

The Deep Learning toolbox allows the users to create and train networks to classify, regress, and forecast assignments in time series. This section describes the overall architecture of the designed deep neural network. The study of fault detection and diagnosis for common faults in the demand-controlled ventilation and heating system is based on established Long-short-term Memory (LSTM) networks to perform the classification and regression for time series [85]. MATLAB provides a platform for implementing neural networks with a wide range of toolboxes, which assist in creating and tuning functions concerning the implementation requirements [203]. The neural network tool in MATLAB can do tasks such as input-output and curve fitting, pattern recognition and classification, data clustering, and time series analysis. The main steps of a neural network design are data collection, creating the network, configuring the network, initializing the weights and biases, training the network, validating the network (post-training analysis), and using the network.

The available samples must be divided into training and testing samples. The training samples are presented to the network during the training phase, and the network will be adjusted based on the error. The training samples are used to measure network generalization, which halts training when generalization stops improving.

The deep learning toolbox in MATLAB supplies a framework for designing and implementing deep neural networks with algorithms, pre-trained models, and applications besides visualization tools, editing network architectures, monitoring the training progress. A deep network can be produced using the `***deepNetworkDesigner***` function in MATLAB. This section describes the main steps of developing a deep neural network for FDD, including data acquisition, data preprocessing, network model design, network model evaluation, and optimization of the best-chosen deep neural network. Figure 6.13 shows a pictorial overview of the FDD based on the deep neural networks.

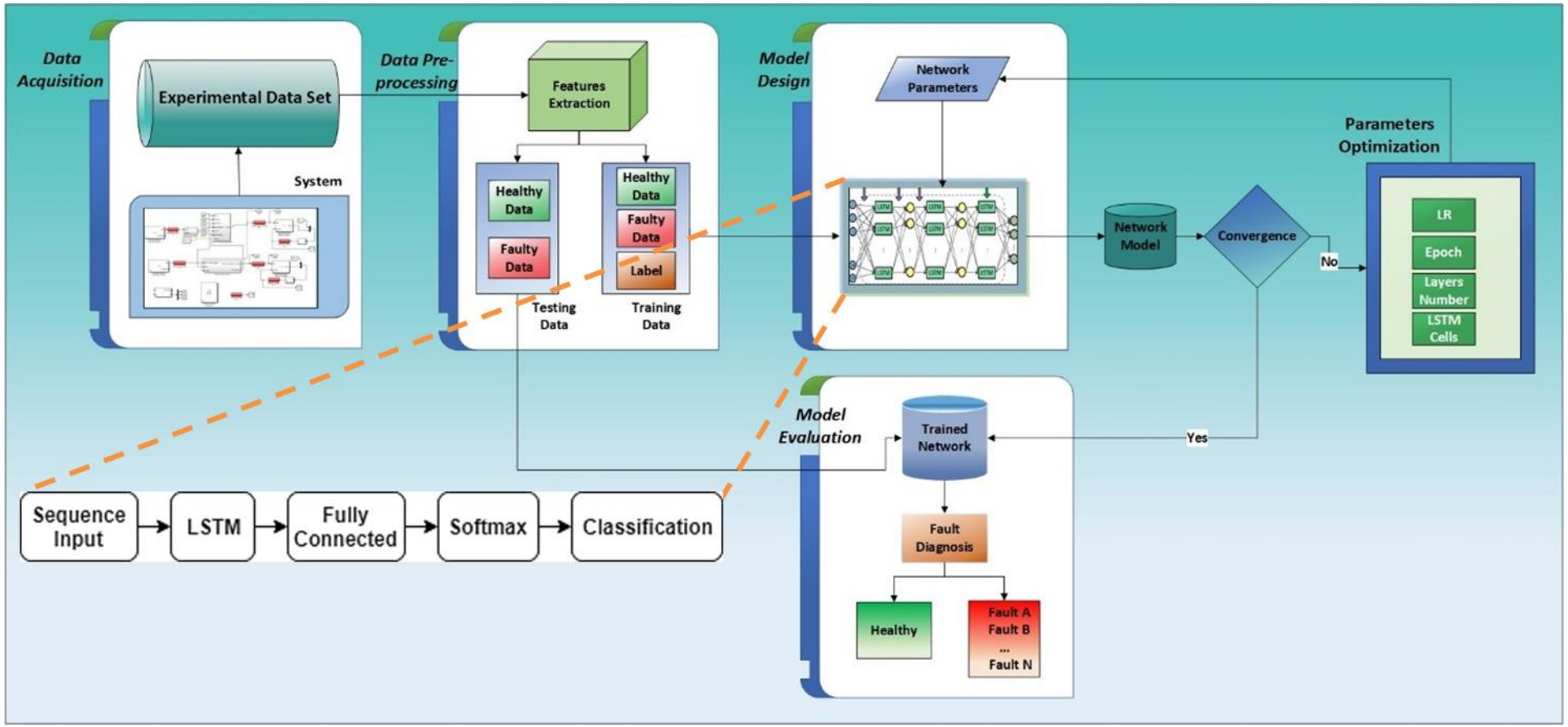


Figure 6.13 A pictorial overview of FDD based on deep neural networks [85].

6.2.1.1. DATA ACQUISITION

In the first step, the model or experimental data are collected. The data includes variables gathered from the status of sensors and actuators such as rooms temperature, outside environment temperature, occupancy of the rooms, CO_2 concentration, damper status, and heater actuator status.In this thesis, we assumed that the actuator variables can get a binary value of zero or one. State zero for the heater actuator describes an off position, and state one shows the on position. State zero for the damper actuator illustrates the closed position, and state one shows the open position of the damper. This step creates a complete collection of data in simulation time respective to each system state, i.e., fault-free (healthy) operation and different modes of faulty operations. The fault injection framework was used to obtain training data for different faults, as discussed in chapter 5. Table 6.23 denotes the abbreviations (labels) of each operating mode as a fault-free mode or occurrence of a specific type of fault.

Table 6.23 Fault modes abbreviations.

Fault Modes	Description
HM	Healthy Mode
BF (CO2)	Battery Fault in CO2 sensor
RF	Routing Fault
DF (OFF)	Damper Fault in closed status with value 0
GF (CO2)	Gain Fault in CO2 sensor
OF (T)	Offset Fault in temperature sensor
HF (OFF)	Heater Fault in off mode with value 0
OF (CO2)	Offset Fault in CO2 sensor
DF (ON)	Damper Fault in open status with value 1
GF (T)	Gain Fault in temperature sensor
BF (T)	Battery Fault in temperature sensor

The total simulation time in this study is 86400 seconds and illustrates the system behavior during one winter day where the sampling time is one second. Table 6.24 shows the collected values for an example gain fault in the temperature sensor scenario.

Table 6.24 Data Acquisition

Time	Stair_Temperature	Occupants	Daily_Temperature	Faulty sensor reading(Gain)	Heater Status(On/Off)	Damper (Open/Close)	Real Room temperature	CO2 Value
0	13.5	0	7	60	0	0	20	400
1	13.5	0	7.0003	59.9714	0	0	19.9904	400
2	13.5	0	7.0007	59.9430	0	0	19.9810	400
3	13.5	0	7.0010	59.9146	0	0	19.9715	400
4	13.5	0	7.0014	59.8863	0	0	19.9621	400
5	13.5	0	7.0018	59.8581	0	0	19.9527	400
6	13.5	0	7.0021	59.8299	0	0	19.9433	400
7	13.5	0	7.0025	59.4518	0	23.5	19.8172	400
8	13.5	0	7.0029	59.0757	0	23.5	19.6919	400
9	13.5	0	7.0032	58.7039	0	23.5	19.5679	400
10	13.5	0	7.0036	58.3364	0	23.5	19.4454	400
11	13.5	0	7.0039	57.9731	0	23.5	19.3243	400
12	13.5	0	7.0043	57.6140	0	23.5	19.2046	400
13	13.5	0	7.0047	57.2590	0	23.5	19.0863	400
14	13.5	0	7.0050	56.9080	0	23.5	18.9693	400
15	13.5	0	7.0054	56.5611	0	23.5	18.8537	400
16	13.5	0	7.0058	56.2181	0	23.5	18.7393	400
17	13.5	0	7.0061	55.8790	0	23.5	18.6263	400
18	13.5	0	7.0065	55.5438	0	23.5	18.5146	400
19	13.5	0	7.0069	55.2124	0	23.5	18.4041	400

6.2.1.2. DATA PREPROCESSING

Data preprocessing is an inevitable step that converts the raw data to the straightforward, practical, and meaningful representation of data as the deep neural network algorithms can learn from the processed high-quality data to solve a problem. The preprocessing includes the format conversion to the correct format, scaling the data, and feature extraction based on the created labels. The labels of the data are: 'Gain Fault in Temperature Sensor Room1 ','Offset Fault in Temperature Sensor Room1', 'Damper Fault in Room1(OFF)', 'Damper Fault in Room1(ON)', 'Heater Fault in Room1(OFF)', 'Battery Fault in Temperature Sensor Room1', 'Offset Fault in CO2 Sensor Room1', 'Gain Fault in CO2 Sensor Room1', 'Healthy Mode in Room1', 'Communication (Routing) Fault in Room1', 'Battery Fault in CO2 Sensor Room1'. Finally, the collected data is divided into training and testing data for the network training and network evaluation.

6.2.1.3. DEEP NEURAL NETWORK DESIGN

This section discusses the main parts of designing a deep neural network which is an optimization problem. Figure 6.14 shows these parts.

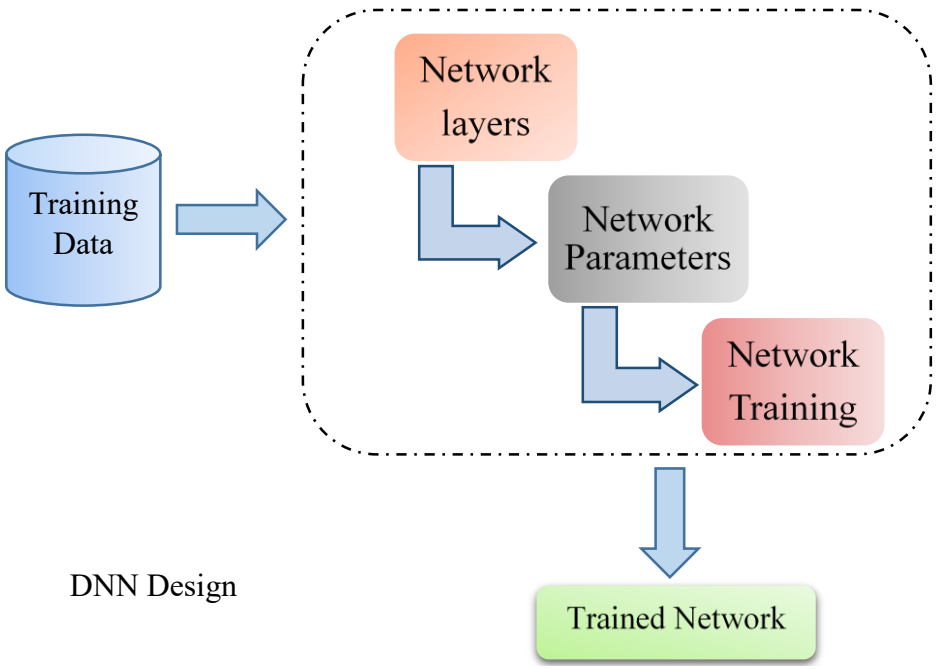


Figure 6.14 A Pictorial Overview of Deep Neural Network Design Network [85].

The training data is fed into a white box, including the following parts.

a) Network Layers

A deep neural network refers to a neural network with more than two hidden layers. The layers selected in this study are the sequence layer that transmits time-series data to the network.

- The first layer is the input sequence layer.
- The second layer which is an LSTM layer specifies the subtype of the sequence layers and learns long-term dependencies between time steps in time series and sequential data [85].
- The third layer is fully connected to multiply the inputs by a weight matrix added to the bias vector.

- The fourth layer is the SoftMax layer to calculate the probabilities respective to each prediction.
- The fifth layer is a classification output layer with classified labels as the results.

"An LSTM network is a type of Recurrent Neural Network (RNN) that can learn long-term dependencies between time steps of sequence data" [203]. A set of LSTM cells can be used to train networks known as Long-Short-Term Memory Networks to perform classification and regression. Figure 6.15 shows an example of LSTM. In this figure, a flow of a time series X with C features with a length of S is shown.

$\mathbf{h}_t$ (output in time step t) and $\mathbf{c}_t$ (information learned in time step t) describe the **hidden state** and the **cell state**, respectively. However, more details are described in [203]. Each cell also contains several components, i.e., input gate, forget gate, cell candidate, and output gate. The values of these components are calculated using an activation function, e.g., sigmoid in this thesis, and the concatenations of the input weights, the recurrent weights, and the bias of each component.

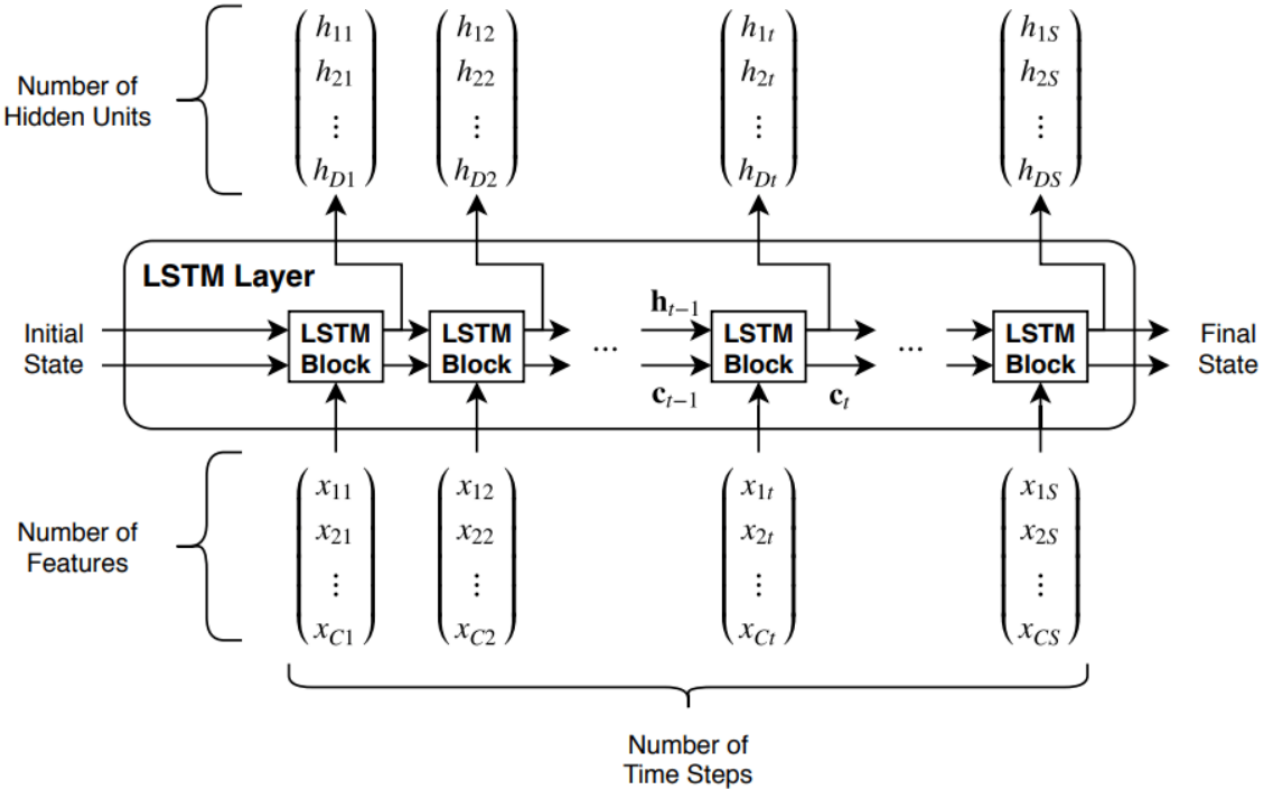


Figure 6.15 LSTM example [203].

An LSTM layer learns a long-term dependency between time steps of a sequence of information. Figure 6.16 shows the architecture of a straightforward LSTM network for classification. The network starts with an input layer of a sequence followed by a layer of LSTM. The network finishes with a fully connected layer, a softmax layer, and a classification output layer to predict class labels.

Figure 6.16 LSTM classification network architecture [203].

It is possible to generate a layered array comprising a sequence input layer, an LSTM layer, a fully linked layer, a softmax layer, and an output classification layer to generate a sequence-to-label LSTM network. The sequence input layer size is set to the amount of input information characteristics and the number of classes is set to the size of the fully linked layer. Then, the number of hidden units and the last output mode for the LSTM layer are specified. These are some steps to follow to construct a suitable LSTM network. Especially when working with lengthy sequences, noisy data, multi-step forecasts, and various input and output variables, time series forecasting is challenging. For time series forecasting, deep neural network techniques give many promises, such as automatic learning of temporal reliance and automatic processing of temporal constructions.

b) Network Parameters

Setting up network parameters or training options is the next step after defining the network layers. The *trainingOptions* function is a function to define these parameters and train the deep neural network. The "*MiniBatchSize*," "*MaxEpochs*," and "*InitialLearnRate*" are some examples of these parameters. The solver uses "*MiniBatchSize*" to update the parameters in each iteration. The term *iteration* refers to each network parameter update by the solver, and the term *epoch* refers to a full pass through the entire data set. The "*MaxEpochs*" specifies the maximum number of epochs. The "*InitialLearnRate*" shows the learning rate. The hardware resources are defined in the execution environment.

c) Network Training

Network training is the last step in the design of a neural network based on training data. The monitoring of network training is essential during the training process as it shows the improvement of the accuracy of the designed network.

6.2.1.4. DEEP NEURAL NETWORK OPTIMIZATION

The goal of this step is to optimize the deep neural network by tuning the optimal network parameters. Parameters include *learning rate, number of hidden layers, number of hidden nodes, and number of epochs*. The accuracy of the network changes by changing each of these parameters. Therefore, the optimum point is determined by monitoring the accuracy when a parameter change is applied while having nothing to do with the rest of the parameters. The training results of the diagnostic model for different parameters are discussed as the following [85].

a) Effect of Learning Rate

Learning rate is a tuning parameter in the optimization of the DNN that controls how quickly the model is adapted to the problem by defining the step size of each iteration while moving toward a minimum of a loss function. Too low learning rate requires more training epochs and smaller weight change in each update to reach the minimum point of the loss function and too high learning rate causes drastic changes which may lead to divergent results. The value of the learning rate in each training greatly influences the performance of the network. In this study, instead of using a fixed learning rate throughout the training process, a higher learning rate can be set in the

first training and be gradually reduced to find the optimized point. Therefore, the network is trained based on a wide range of learning rates such as 0.1, 0.05, 0.025, 0.01, 0.005, 0.0005, and 0. On the other hand, the initial settings of the other network parameters had been set to the fixed values as follows. The number of the hidden units is set to 180, the number of output classes is set to 11, the mini-batch size is set to 150, and the maximum epoch is set to 150. Figure 6.17 shows the model accuracy with different learning rates. According to these results, a high value of the learning rate causes a low network accuracy, and when the learning rate value decreases, the accuracy increases gradually. The best accuracy for the DNN model is 93% at a learning rate of 0.005. With more reduction in the learning rate, the accuracy tends to decrease because the maximum number of epochs is not enough for the lower learning rate to reach the solution. Also, the training time increases with low learning rates because more time is then needed to meet the convergence.

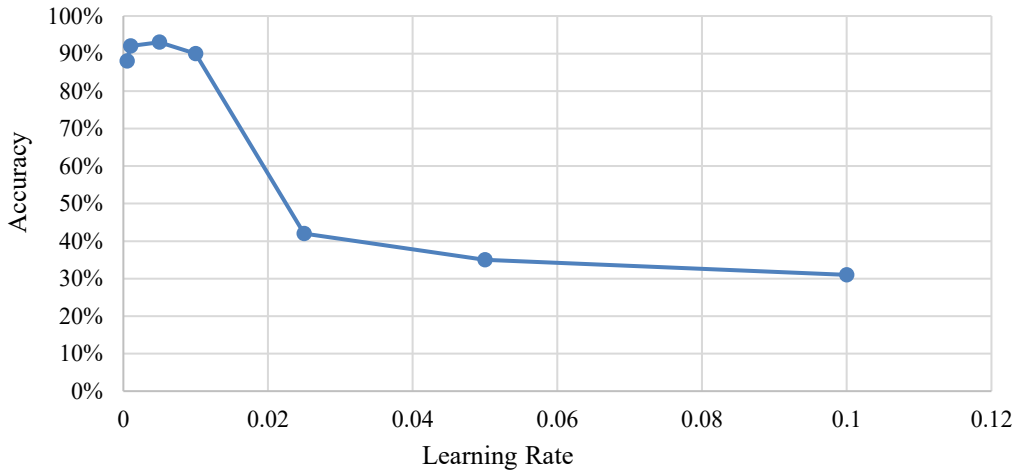


Figure 6.17 Model accuracy with different learning rates.

b) Effect of Number of Hidden Layers

The number of hidden layers is also an essential factor affecting training performance that describes the main difference between the traditional NN and DNN. The DNN model is trained with the following parameters. During the tuning, the number of hidden layers increases gradually with the values of 3, 4, 5, 6, 7 while the other parameters remain the same in the training, e.g., the number of output classes is 11, the maximum number of epochs is 200, the minimum batch size is 150, and the learning rate is 0.001. Figure 6.18 demonstrates the model accuracy with a different number of hidden layers. According to the results in this figure, it can be observed that the accuracy decreases gradually when increasing the depth of the network. The accuracy decreases from 94%, with three hidden layers to 30% with seven hidden layers. Also, the required training time increases by increasing the number of hidden layers. Therefore, three hidden layers are selected as a suitable depth of the designed network model.

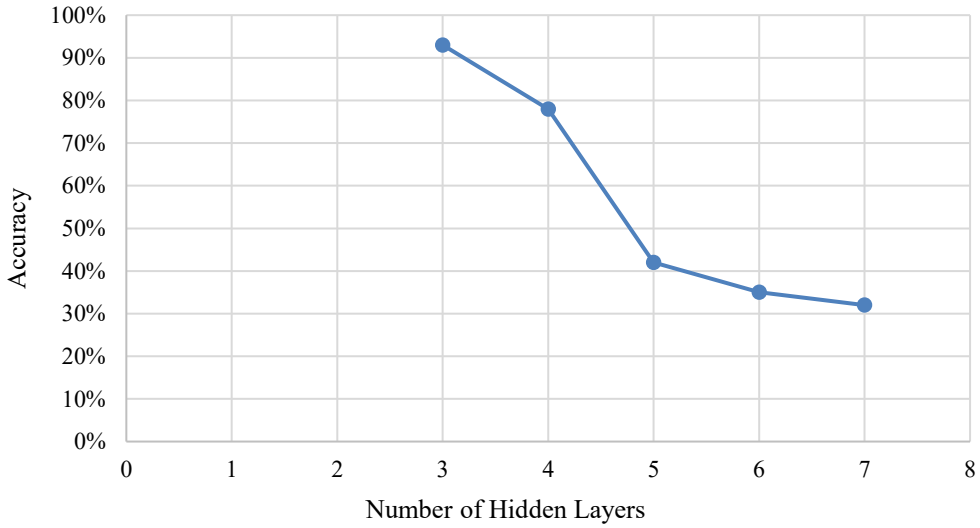


Figure 6.18 Model accuracy with different numbers of hidden layers.

c) Effect of Number of Hidden Layer Units

Hidden layer nodes (or units) should be considered in the network training stage because of their significant influence on the training progress. In general, the number of nodes of the input layer is set based on the number of training data features, whereas the nodes of the output layer are equal to the output classes. Figure 6.19 illustrates the accuracy of the model with a different number of hidden units. This thesis considers tuning the number of hidden layer units ranging as follows: 25, 50, 100, 150, 200, 250, 300, 350, 400. The other experimental parameters in these setups are the number of classes of 11, the maximum epochs of 250, the minimum batch size of 150, and the learning rate of 0.001. The results in Figure 6.19 show that the training accuracy is poor with small hidden nodes. Increasing the number of nodes improves the accuracy to exceed 80% with 50 nodes, which seems insufficient. By increasing the number of nodes, it is observed that the accuracy improves and reaches the best value with 250 nodes (accuracy around 98%). However, more increase of nodes is not helpful anymore. The number of nodes is related to training data samples. However, the training time increases by increasing the number of hidden nodes.

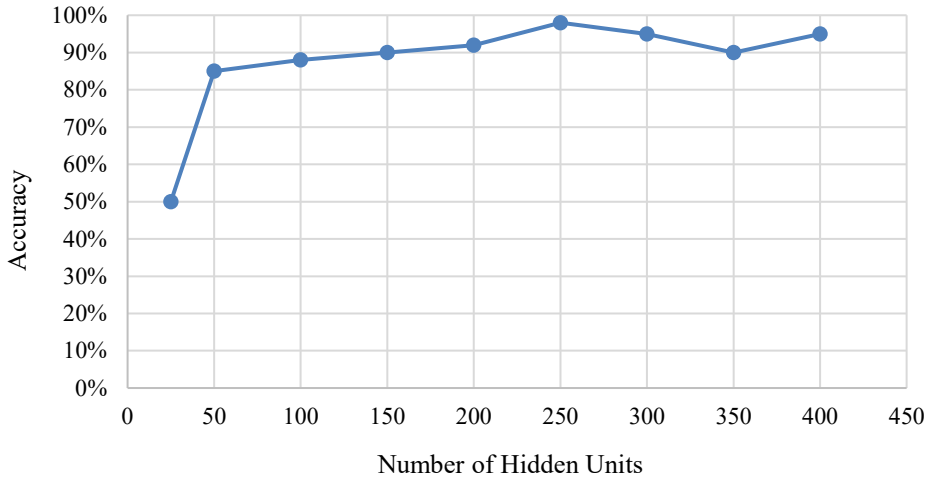


Figure 6.19 Model accuracy with different numbers of hidden units.

d) Effect of Number of Epochs

The last parameter recognized as an essential factor in the DNN model accuracy is the number of epochs. The best parameters of the previous experiments are selected in tuning the number of training epochs. The other setup parameters are the number of the hidden units of 250, the learning rate of 0.005, the number of hidden layers 3, the output classes of 11, and the mini-batch size of 150. Figure 6.20 describes the accuracy of the DNN model over different numbers of epochs. In the beginning, the number of epochs is 25 and then increases gradually in the next training. The results show the effect of increasing the number of epochs on the performance, where the accuracy is improved by increasing the epoch number to be around 85% with 150 epochs. When the number of epochs exceeds 250, the accuracy of the DNN model improved to reach around 95%. The best-observed value of the accuracy is 98% with 490 epochs, and a further number of epochs cause an increase in the training time.

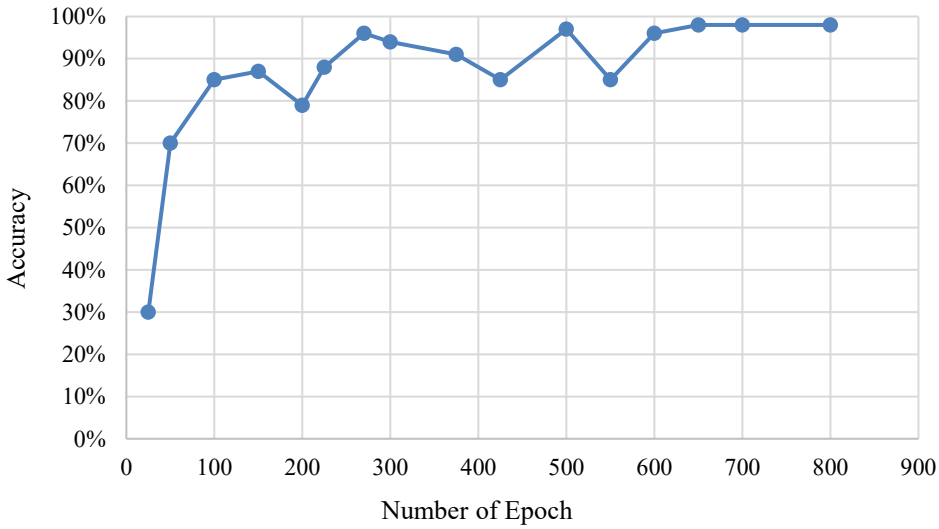
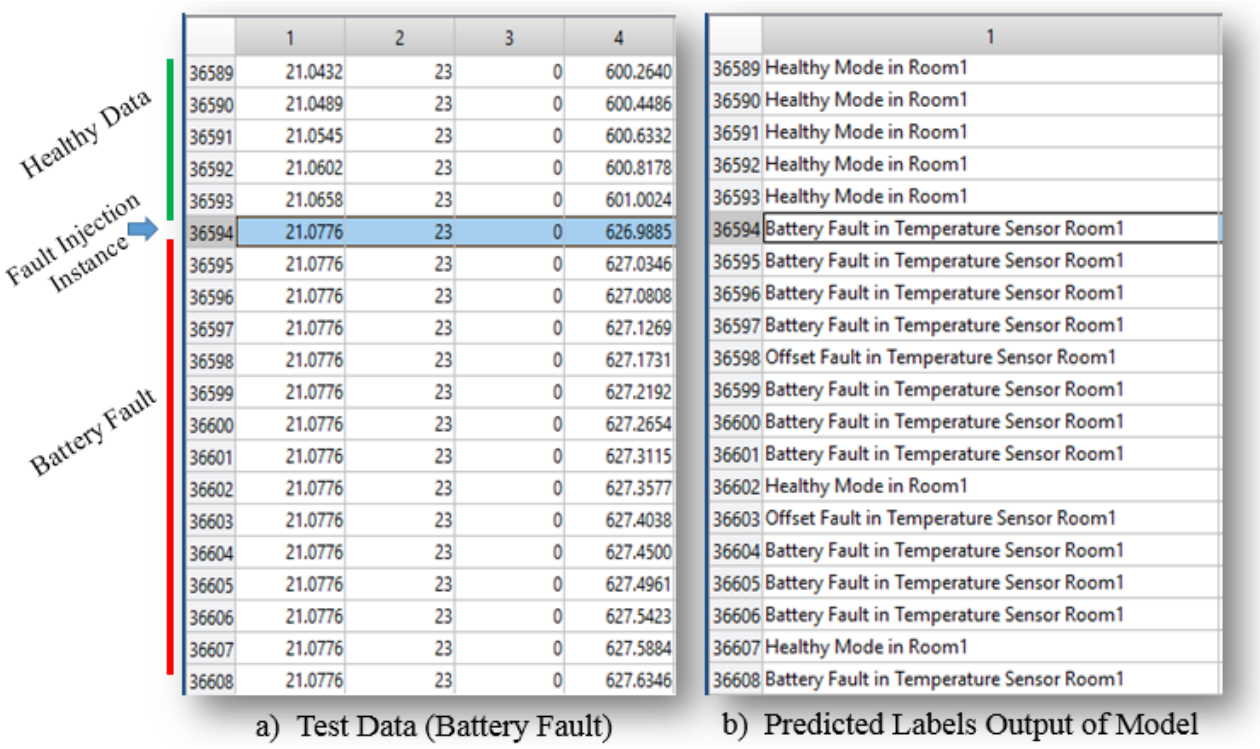


Figure 6.20 Model accuracy with different numbers of the epoch.

6.2.1.5. DEEP NEURAL NETWORK RESULT

After training the proposed deep neural network, the model gives the predicted output as a result. Figure 6.21 shows a sample of model prediction results for a battery fault as an example.



	1	2	3	4
36589	21.0432	23	0	600.2640
36590	21.0489	23	0	600.4486
36591	21.0545	23	0	600.6332
36592	21.0602	23	0	600.8178
36593	21.0658	23	0	601.0024
36594	21.0776	23	0	626.9885
36595	21.0776	23	0	627.0346
36596	21.0776	23	0	627.0808
36597	21.0776	23	0	627.1269
36598	21.0776	23	0	627.1731
36599	21.0776	23	0	627.2192
36600	21.0776	23	0	627.2654
36601	21.0776	23	0	627.3115
36602	21.0776	23	0	627.3577
36603	21.0776	23	0	627.4038
36604	21.0776	23	0	627.4500
36605	21.0776	23	0	627.4961
36606	21.0776	23	0	627.5423
36607	21.0776	23	0	627.5884
36608	21.0776	23	0	627.6346

a) Test Data (Battery Fault)

	1
36589	Healthy Mode in Room1
36590	Healthy Mode in Room1
36591	Healthy Mode in Room1
36592	Healthy Mode in Room1
36593	Healthy Mode in Room1
36594	Battery Fault in Temperature Sensor Room1
36595	Battery Fault in Temperature Sensor Room1
36596	Battery Fault in Temperature Sensor Room1
36597	Battery Fault in Temperature Sensor Room1
36598	Offset Fault in Temperature Sensor Room1
36599	Battery Fault in Temperature Sensor Room1
36600	Battery Fault in Temperature Sensor Room1
36601	Battery Fault in Temperature Sensor Room1
36602	Healthy Mode in Room1
36603	Offset Fault in Temperature Sensor Room1
36604	Battery Fault in Temperature Sensor Room1
36605	Battery Fault in Temperature Sensor Room1
36606	Battery Fault in Temperature Sensor Room1
36607	Healthy Mode in Room1
36608	Battery Fault in Temperature Sensor Room1

b) Predicted Labels Output of Model

Figure 6.21 Test data and predicted labels of the designed model.

7. EVALUATION AND RESULTS

This section discusses the evaluation and results of fault diagnosis for both diagnostic methods as classifiers discussed in this thesis. The first method is a composed diagnostic classification method introduced in this thesis based on the causal relations in fuzzy Bayesian networks using relation direction probabilities, and the second method is a data-driven classification method based on a deep neural network. Performance indicators are used to evaluate and compare different classification models [204]. Therefore, this section introduces the performance metrics and the results from these two methods will be compared and discussed to evaluate the effectiveness of each method. Statistical classification is visualized in Figure 7.1 to distinguish different classes of a set of data in a classification problem, e.g., True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). In these terms, the word "positive" is the output of the diagnosis algorithm that is predicted and diagnosed. TP shows the number of positive elements classified accurately. TP indicates that the injected faults are also correctly predicted. FP is a false alarm and describes the number of actual negative elements that are classified as positive. FP shows the incorrect diagnosis of the system as a healthy mode when it is faulty. TN indicates the number of negative examples classified accurately (correct rejection). TN describes the truly healthy mode that is not also diagnosed. FN is defined as the missed class of faults or the number of positive elements classified as negatives. FN shows the faults in the system known from the fault injection, but the diagnostic classifier did not successfully diagnose them.

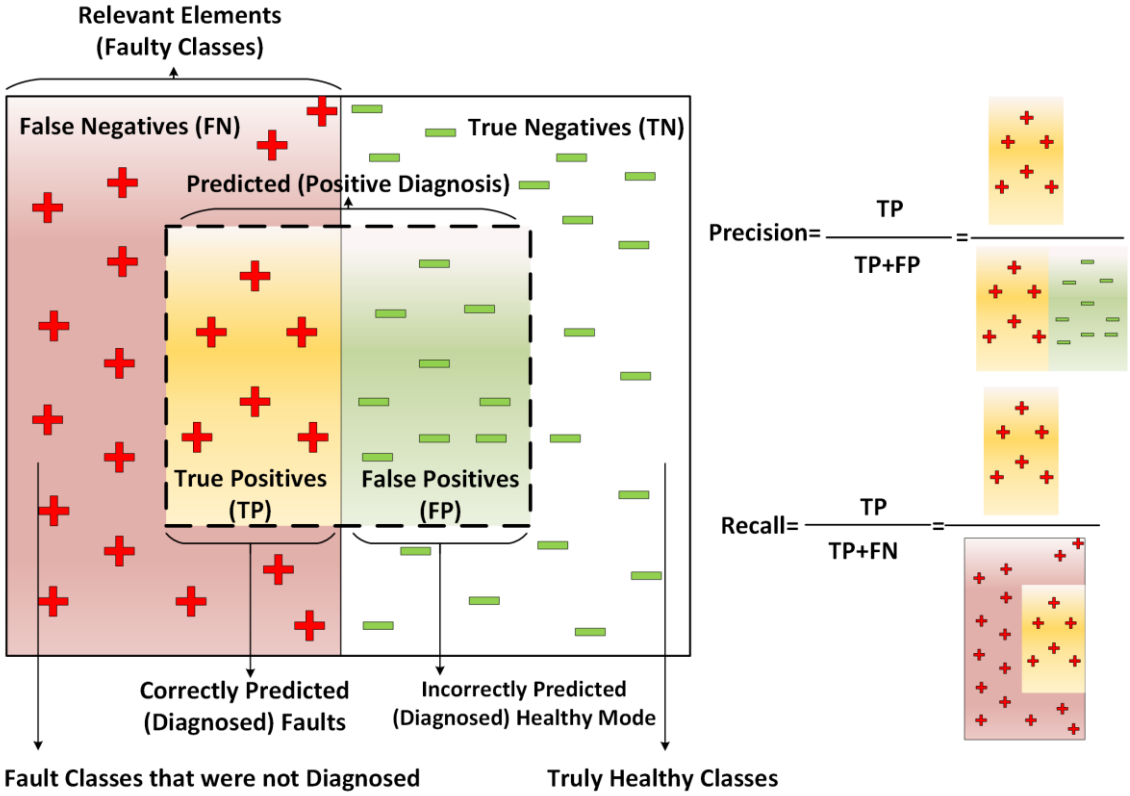


Figure 7.1 Statistical Classification and Performance Measures, e.g., Precision and Recall.

There are several evaluation metrics for classifiers. The diagnostic methods that rely on classifiers to solve classification problems use some parameters to measure the performance known as performance measures, e.g., precision, recall, F1, and accuracy, are used based on main classes of

TP, FP, FN, and TN. A confusion matrix is used to get insight and to visualize the performance of an algorithm. Confusion matrices distinguish between predicted and actual values and are applied to binary (or double-class-) and multiclass classifications problems [205]. The rows of the matrix indicate the samples in a predicted class and columns show the samples in an actual class (or vice versa). Figure 7.2 indicates an example of a confusion matrix for binary classification.

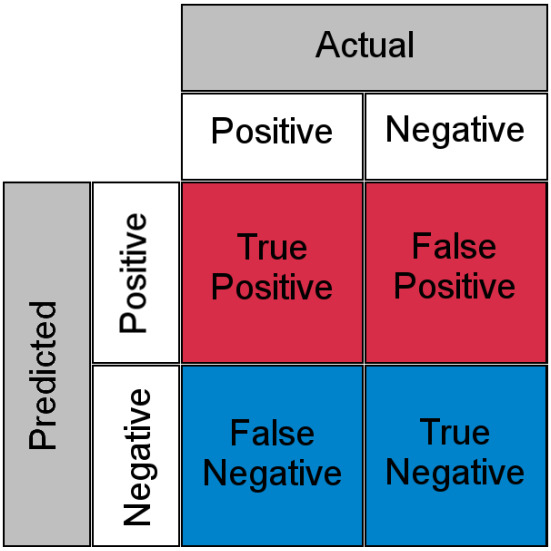


Figure 7.2 Confusion matrix for binary classification problem [206].

Precision or **Positive Predictive Value (PPV)** shows how many of the total diagnostic results are correct or how relevant the detected items are. The precision [207] is the fraction of the truly predicted elements to the total detected elements. The precision can be determined by the equation below.

$$Precision = \frac{TP}{TP + FP} \tag{7.1}$$

Recall or **sensitivity** or **hit rate** or **True Positive Rate (TPR)** shows how many relevant items are correctly diagnosed. The recall is the fraction of correctly detected items among all the items that should be detected [207]. It means the ability to find all relevant elements. Recall can be determined by the equation below.

$$Recall = \frac{TP}{TP + FN} \tag{7.2}$$

F-Measure or **F-score** combines precision and recall rates and is the harmonic mean of precision and recall instead of the arithmetic mean. **F1** is an F-Measure with evenly weighted recall and precision. The f-measure gives an estimation of the accuracy of the system under test [207]. The equation below determines the F1.

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{7.3}$$

Accuracy (ACC) is the proportion of correct predictions [208], as below:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{7.4}$$

7.1. Evaluation and Results for Composed Diagnostic Classifier based on Knowledge-driven and Data-driven Methods

During the evaluation phase, the total number of 110 fault cases have been considered calculated based on 22 fault injection values and five injection times. These fault cases have been injected in five different instances of time {17000,34000,51000,68000,85000}, And four fault types of }"CO2Sensor","DamperActuator","TemperatureSensor","HeaterActuator"{ with 22 fault values for these four fault types shown in the Function 7.1. The range of values for each signal is divided into three subdomains. Therefore, the value vectors are defined to ensure that sufficient fault samples from each subdomain are considered.

Function 7.1 Value Vectors of Fault Injection

```
Fault_Injection_Co2Value_Vector = [350,400,450,550,600,650,750,800,850];
Fault_Injection_DamperValue_Vector = [0,1];
Fault_Injection_TempValue_Vector = [16,17,18,19,20,21,22,23,24];
Fault_Injection_HeaterValue_Vector = [0,1];
```

The numbers of truly diagnosed faults considering the type and the time of faults in different cumulative ranks are depicted in Table 7.1. This table shows the number of correct diagnoses (TPs) categorized in different cumulative ranks for different fault types.

Table 7.1 Number of diagnoses in different cumulative ranks for different fault types

Fault Type	**Total number of injected fault cases**	**Number of Diagnoses (TPs) in Rank1**	**Number of Diagnoses (TPs) in Rank 1,2**	**Number of Diagnoses (TPs) in Rank1,2,3**	**Number of Diagnoses (TPs) in Rank 1,2,3,4**	**Number of Diagnoses (TPs) in Rank 1,2,3,4,5**
CO_2 Sensor	45	38	41	43	43	43
Damper Actuator	10	10	10	10	10	10
Temperature Sensor	45	35	40	41	41	42
Heater Actuator	10	10	10	10	10	10
Total Number	110	93	101	104	104	105

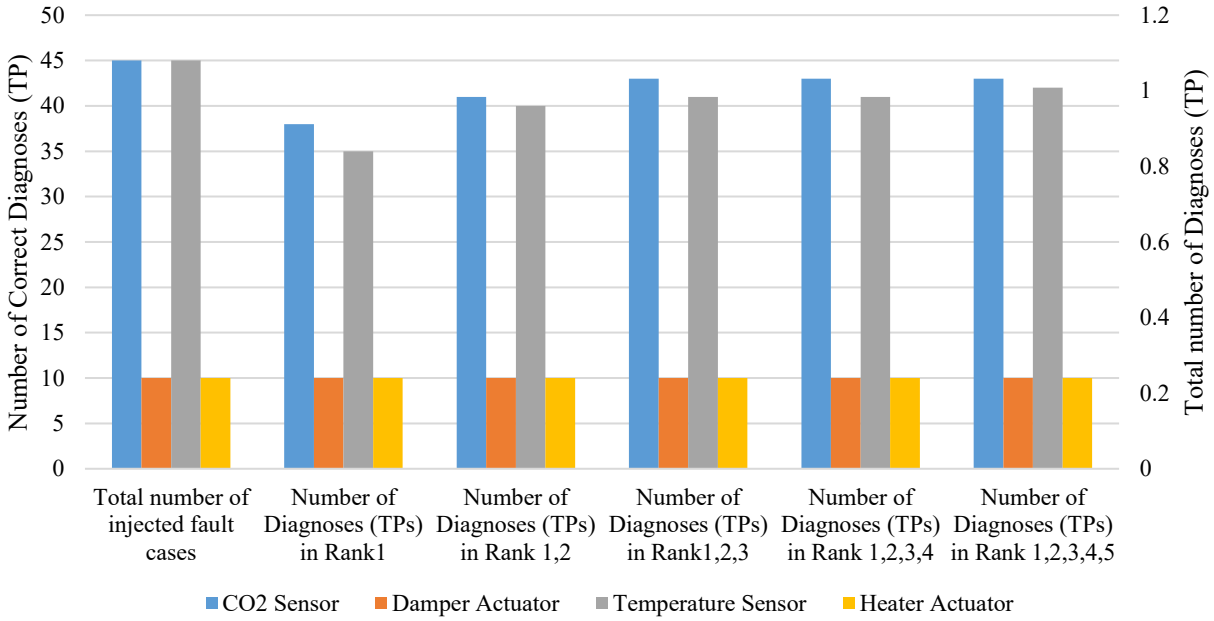


Figure 7.3 Number of correct diagnoses (TPs) categorized in different cumulative ranks for different fault types.

The percentages of diagnoses (TPs in %) are shown in Table 7.2.

Table 7.2 Percentages of diagnosis (TP)

Fault Type	**Percentage of Diagnoses (TPs) in Rank1**	**Percentage of Diagnoses (TPs) in Rank 1,2**	**Percentage of Diagnoses (TPs) in Rank 1,2,3**	**Percentage of Diagnoses (TPs) in Rank 1,2,3,4**	**Percentage of Diagnoses (TPs) in Rank 1,2,3,4,5**
CO_2 Sensor	84.44	91.11	95.55	95.55	95.55
Damper Actuator	100	100	100	100	100
Temperature Sensor	77.77	88.88	91.11	91.11	93.33
Heater Actuator	100	100	100	100	100
Average Percentage of Diagnosis	**90.55**	**95**	**96.66**	**96.66**	**97.22**

Figure 7.4 contains the diagram with the percentages of correct diagnoses (TPs in %) in different cumulative ranks for different fault types. This figure shows the composed diagnostic method can overall diagnose 97.22% of faults truly (TPs over the whole five cumulative ranks). From this value, 90.55% of faults were diagnosed at the first rank, 4.44% were diagnosed at the second rank, 1.66% were diagnosed at the third or fourth rank, and 0.56% were diagnosed at the fifth rank.

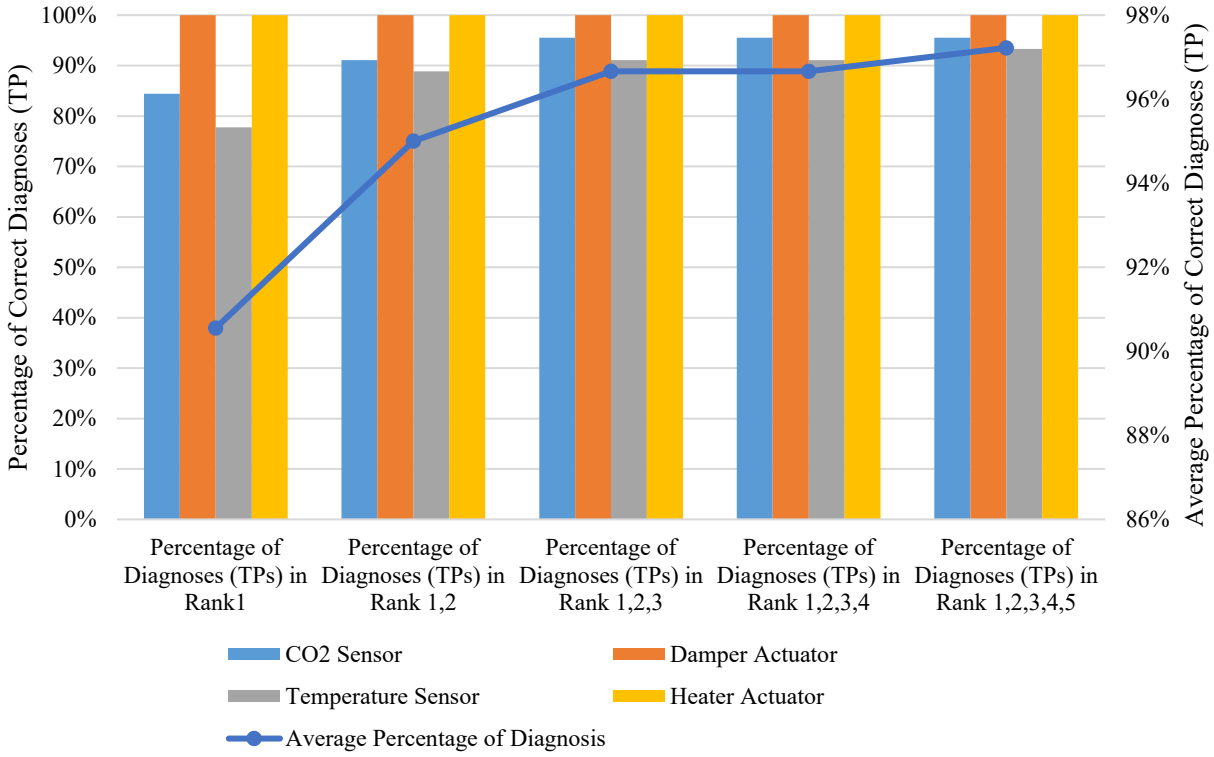


Figure 7.4 Percentage of correct diagnoses (TPs) in different Ranks for different fault types.

The accuracy of the diagnosis method is measured as the number of truly diagnosed faults considering the type and time of fault to the total population based on equation (7.4) that is depicted in Table 7.3. This table shows the cumulative accuracy increases when only considering the diagnosis results from the first rank to the top five ranks.

Table 7.3 Accuracy (ACC) of fault diagnosis

Fault Type	Accuracy for Rank 1	Accuracy for Rank 1, 2	Accuracy for Rank 1, 2, 3	Accuracy for Rank 1, 2, 3, 4	Accuracy for Rank 1, 2, 3, 4, 5
CO_2 Sensor	84.78	91.30	95.65	95.65	95.65
Damper Actuator	100	100	100	100	100
Temperature Sensor	78.26	89.13	91.30	91.30	93.47
Heater Actuator	100	100	100	100	100
Average Accuracy	90.76	95.10	96.73	96.73	97.28

Figure 7.5 shows the diagram of the fault diagnosis accuracy of detections in different cumulative ranks for different fault types. As shown, including more diagnosis ranks increases the overall accuracy and the accuracy of each fault type. The diagnosis method's average accuracy shows the accuracy increases from 90.76% to 97.28% when considering the top five ranks instead of only the first rank.

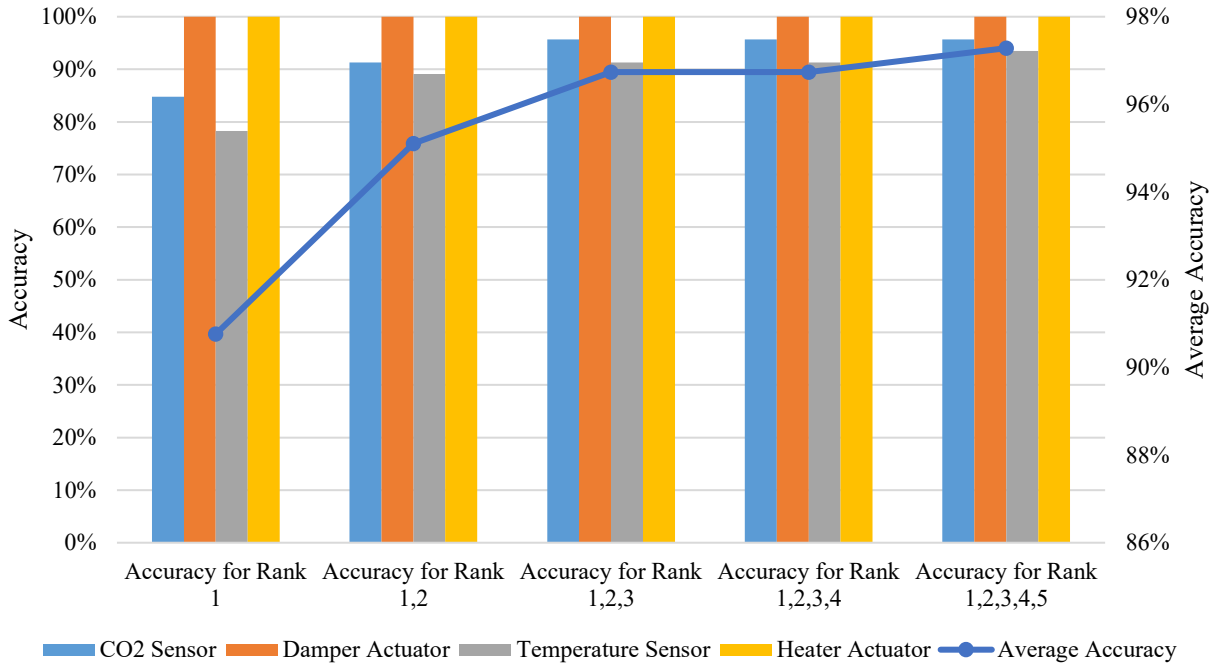


Figure 7.5 Fault Diagnosis Accuracy in different Ranks for different Fault Types.

Recall is calculated for the diagnosis method introduced in this thesis based on equation 7.2, and the results are shown in Table 7.4.

Table 7.4 Recall (TPR)

Fault Type	Recall for Rank 1	Recall for Rank 1, 2	Recall for Rank 1, 2, 3	Recall for Rank 1, 2, 3, 4	Recall for Rank 1, 2, 3, 4, 5
CO_2 Sensor	84.44	91.11	95.55	95.55	95.55
Damper Actuator	100	100	100	100	100
Temperature Sensor	77.77	88.88	91.11	91.11	93.33
Heater Actuator	100	100	100	100	100
Average Recall	90.55	95	96.66	96.66	97.22

Figure 7.6 shows the diagram of the recall performance metric of fault diagnosis in different cumulative ranks for different fault types. As shown, including more diagnosis ranks in a cumulative manner increases the overall and specific recall value for each fault type. The average recall of the diagnosis method shows the recall increases from 90.55% to 97.22% when considering the top five ranks instead of only the first rank.

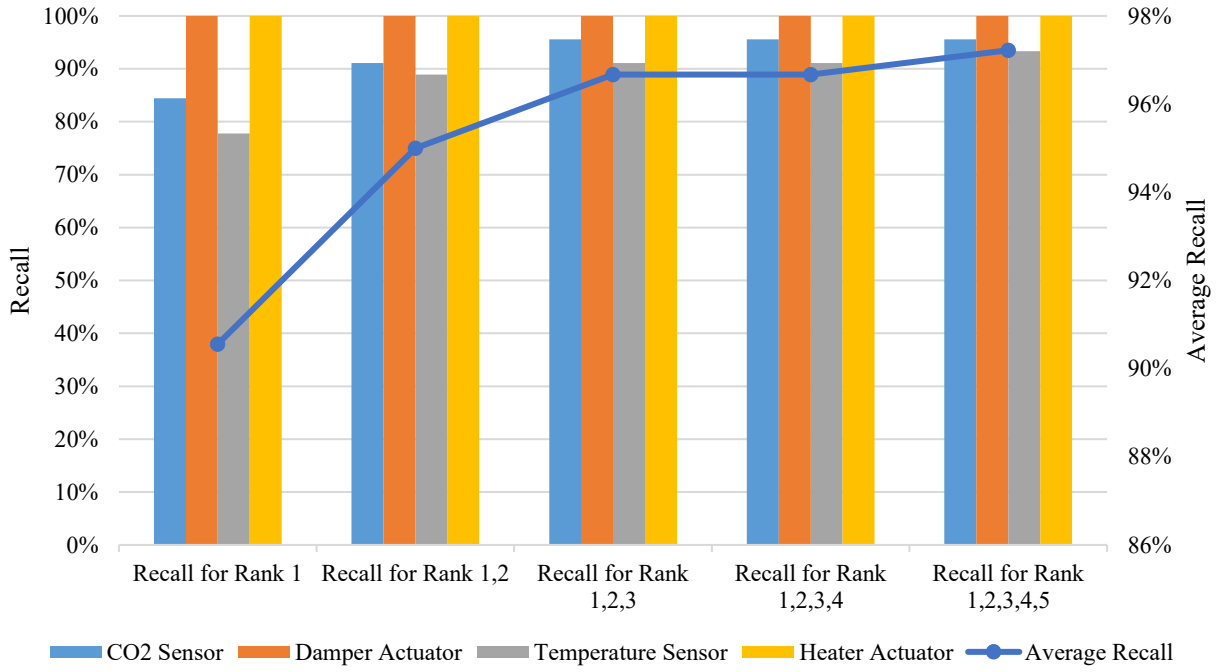


Figure 7.6 Recall (TPR) in different cumulative ranks for different Fault types.

Precision is calculated for the diagnosis method introduced in this thesis based on equation 7.1, and the results are shown in Table 7.5.

Table 7.5 Precision (PPV)

Fault Type	Precision in Rank 1	Precision in Rank 1,2	Precision in Rank 1,2,3	Precision in Rank 1,2,3,4	Precision in Rank 1,2,3,4,5
CO_2 Sensor	100	100	100	100	100
Damper Actuator	100	100	100	100	100
Temperature Sensor	100	100	100	100	100
Heater Actuator	100	100	100	100	100
Average Precision	100	100	100	100	100

Figure 7.7 shows the diagram of the precision. As shown, the average precision of the diagnosis method shows a distinguished result of 100% for every fault type for every component.

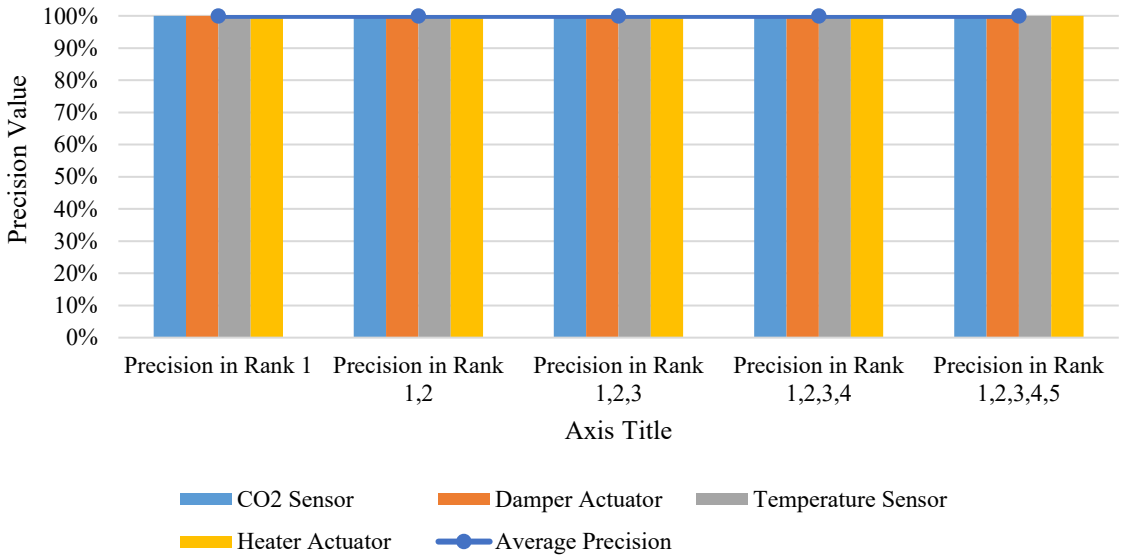


Figure 7.7 Precision (PPV) in different cumulative ranks for different Fault types.

F1 is calculated for the diagnosis method introduced in this thesis based on equation 7.3, and the results are shown in Table 7.6.

Table 7.6 F1 in different cumulative ranks for different Fault types

Fault Type	F1 for Rank 1	F1 for Rank 1, 2	F1 for Rank 1, 2, 3	F1 for Rank 1, 2, 3, 4	F1 for Rank 1, 2, 3, 4, 5
CO_2 Sensor	91.56	95.34	97.72	97.72	97.72
Damper Actuator	100	100	100	100	100
Temperature Sensor	87.50	94.11	95.34	95.34	96.55
Heater Actuator	100	100	100	100	100
Average F1	94.76	97.36	98.26	98.26	98.56

Figure 7.8 shows the diagram of the F1 in different cumulative ranks for different fault types. As shown, including more diagnosis ranks increases the overall F1 value and the F1 of each fault type. The average F1 or F-score of the diagnosis method shows the F1 increases from 94.76% to 98.56% when considering the top five ranks instead of only the first rank.

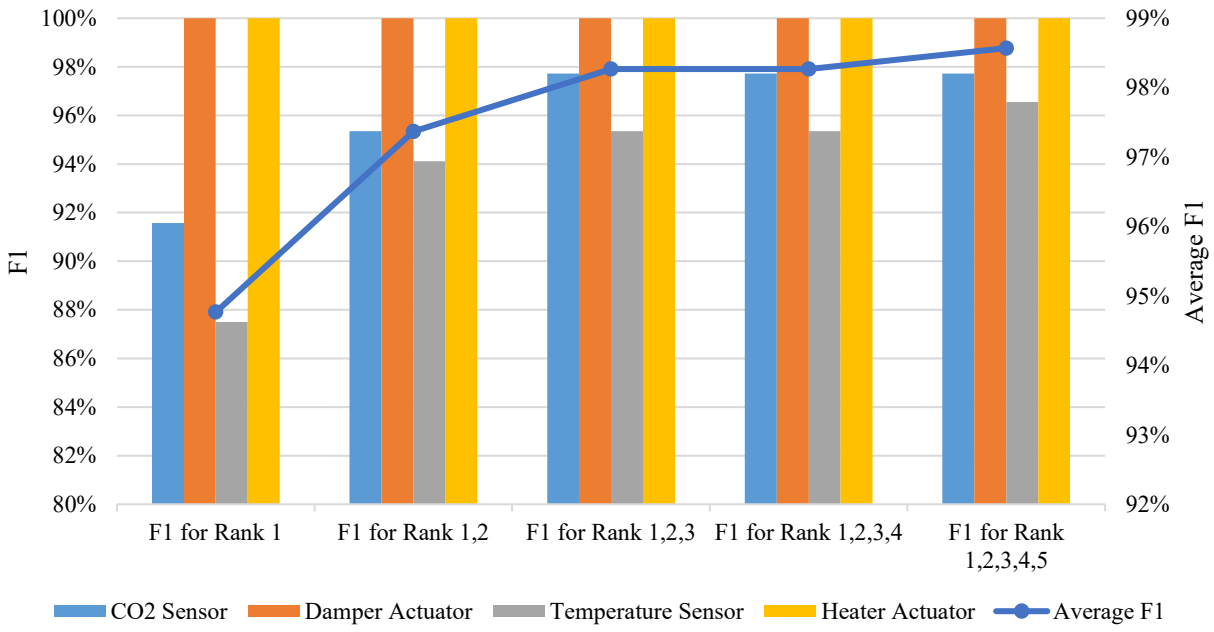


Figure 7.8 F1 in different cumulative ranks for different Fault types.

The overview of the performance metrics of the fault diagnosis method introduced in this thesis indicates the average values increases when considering the top five ranks instead of only the first rank. For example, the diagnosis result with considering the top five ranks is always better than considering only the top two ranks, which is summarized in the table below.

Table 7.7 Overview of performance metrics of fault diagnosis method in this thesis

Fault Labels	Relevant Elements / Diagnosis	TP	FP	TN	FN	TotPop	Precision	Recall	F-Score or F1	Accuracy
CO_2 Sensor Rank 1	45	38	0	1	7	46	100	84.44	91.56	84.78
CO_2 Sensor Rank 1,2	45	41	0	1	4	46	100	91.11	95.34	91.30
CO_2 Sensor Rank 1,2,3	45	43	0	1	2	46	100	95.55	97.72	95.65
CO_2 Sensor Rank 1,2,3,4	45	43	0	1	2	46	100	95.55	97.72	95.65
CO_2 Sensor Rank 1,2,3,4,5	45	43	0	1	2	46	100	95.55	97.72	95.65
Damper Actuator Rank 1	10	10	0	1	0	11	100	100	100	100
Damper Actuator Rank 1,2	10	10	0	1	0	11	100	100	100	100
Damper Actuator Rank 1,2,3	10	10	0	1	0	11	100	100	100	100
Damper Actuator Rank 1,2,3,4	10	10	0	1	0	11	100	100	100	100
Damper Actuator Rank 1,2,3,4,5	10	10	0	1	0	11	100	100	100	100
Temperature Sensor Rank 1	45	35	0	1	10	46	100	77.77	87.50	78.26
Temperature Sensor Rank 1,2	45	40	0	1	5	46	100	88.88	94.11	89.13
Temperature Sensor Rank 1,2,3	45	41	0	1	4	46	100	91.11	95.34	91.30
Temperature Sensor Rank 1,2,3,4	45	41	0	1	4	46	100	91.11	95.34	91.30
Temperature Sensor Rank 1,2,3,4,5	45	42	0	1	3	46	100	93.33	96.55	93.47
Heater Actuator Rank 1	10	10	0	1	0	11	100	100	100	100
Heater Actuator Rank 1,2	10	10	0	1	0	11	100	100	100	100
Heater Actuator Rank 1,2,3	10	10	0	1	0	11	100	100	100	100
Heater Actuator Rank 1,2,3,4	10	10	0	1	0	11	100	100	100	100
Heater Actuator Rank 1,2,3,4,5	10	10	0	1	0	11	100	100	100	100

7.2. Evaluation and Results for Data-driven-based Fault Diagnosis Based on Multiclass Classification

This section presents and discusses the evaluation results for the fault diagnosis based on deep neural networks. The evaluation strategy evaluates the proposed deep neural network based on different faults types described in Table 6.23. Once the training step is finished, the final network model with the optimized parameters is ready for evaluation. The setup parameters are the number of the hidden units of 250, the learning rate of 0.005, the number of hidden layers 3, the output classes of 11, and the mini-batch size of 150. The evaluation results of the deep neural network are calculated based on testing data. Testing data consists of 10 types of faults injected at different times and the healthy data.

In a deep neural network, multi-class confusion matrices are used. A multi-class confusion matrix, unlike a binary confusion matrix, has no positive or negative classes. Figure 7.9 shows a multi-class confusion matrix [207], considering that the confusion matrix in this method includes more than two categories. The grey diagonal shows the correct predictions and the white diagonal represents the incorrect predictions. *Confusionmat* computes confusion matrix for classification problem in MATLAB [209].

True class \ Predicted class	A	B	C	D
A	AA	AB	AC	AD
B	BA	BB	BC	BD
C	CA	CB	CC	CD
D	DA	DB	DC	DD

Figure 7.9 Confusion matrix for the multi-class classification problem, with A, B, C, and D classes [210].

The equivalent confusion matrix of a multi-class confusion matrix to the confusion matrix in Figure 7.2 is shown below. In Figure 7.10, let assume that class B is the target class; then, to obtain the equivalent TN, all the elements shown with the TN label must be summed up. Similarly, for obtaining the equivalent FP and FN, all the elements shown with FP and FN labels must be summed up. The TP value is the single element crossed from the B column and B row.

Predicted Class \ True Class	Classes	A	B	C	D
	A	TN	FN	TN	TN
	B	FP	TP	FP	FP
	C	TN	FN	TN	TN
	D	TN	FN	TN	TN

Figure 7.10 Confusion matrix for the multi-class classification problem equivalent to a double-class confusion matrix.

According to the confusion matrix in Figure 7.11, it can be observed that the network model can diagnose ten different types of faults and the healthy data with accuracy values ranging from 93% to 100%. The matrix shows the precision and the recall percentage of each class. The model's

overall accuracy is based on all classes (97.4%). This value is considered a promising result of FDD in the DCV and heating systems using the deep neural network, particularly the LSTM model.

Predicted Output \ Target Output	HM	BF CO_2	RF	DF OFF	GF CO_2	OF (T)	HF OFF	OF CO_2	DF ON	GF (T)	BF (T)	
HM	202629 38.1%	395 0.1%	1293 0.2%	1 0.0%	0 0.0%	3 0.0%	24 0.0%	1 0.0%	394 0.1%	1 0.0%	135 0.0%	98.9% 1.1%
BF CO_2	1884 0.4%	31605 5.9%	796 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	430 0.1%	91.0% 9.0%
RF	1 0.0%	0 0.0%	29911 5.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100.0% 0.0%
DF OFF	908 0.2%	0 0.0%	0 0.0%	31998 6.0%	4 0.0%	0 0.0%	0 0.0%	12 0.0%	0 0.0%	0 0.0%	0 0.0%	97.2% 2.8%
GF CO_2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	31995 6.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
OF (T)	14 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	31950 6.0%	994 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	96.9% 3.1%
HF OFF	333 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	30981 5.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.9% 1.1%
OF CO_2	2441 0.5%	0 0.0%	0 0.0%	1 0.0%	1 0.0%	1 0.0%	1 0.0%	31987 6.0%	40 0.0%	1 0.0%	48 0.0%	92.7% 7.3%
DF ON	45 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	31566 5.9%	0 0.0%	0 0.0%	99.9% 0.1%
GF (T)	22 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	46 0.0%	0 0.0%	0 0.0%	0 0.0%	31998 6.0%	75 0.0%	99.6% 0.4%
BF (T)	3722 0.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	31313 5.9%	89.4% 10.6%
	95.6% 4.4%	98.8% 1.2%	93.5% 6.5%	100.0% 0.0%	100.0% 0.0%	99.8% 0.2%	96.8% 3.2%	100.0% 0.0%	98.6% 1.4%	100.0% 0.0%	97.9% 2.1%	**97.4%** **2.6%**

Figure 7.11 Confusion matrix of designed diagnostic neural network.

Figure 7.12 shows precision values calculated for each class of faults [207] in fault diagnosis based on DL. This precision indicates correctly predicted samples or correct diagnosed faults among all of the predicted data. The lowest precision value belongs to the battery fault in the temperature sensor (0.894), and the highest precision value is 1 for the routing fault and gain fault in the CO_2 sensor. The average precision for all fault types is 0.967, while the average precision for the stuck-at faults, including damper fault in closed status, heater fault in off status, and damper fault in open status, is 0.986.

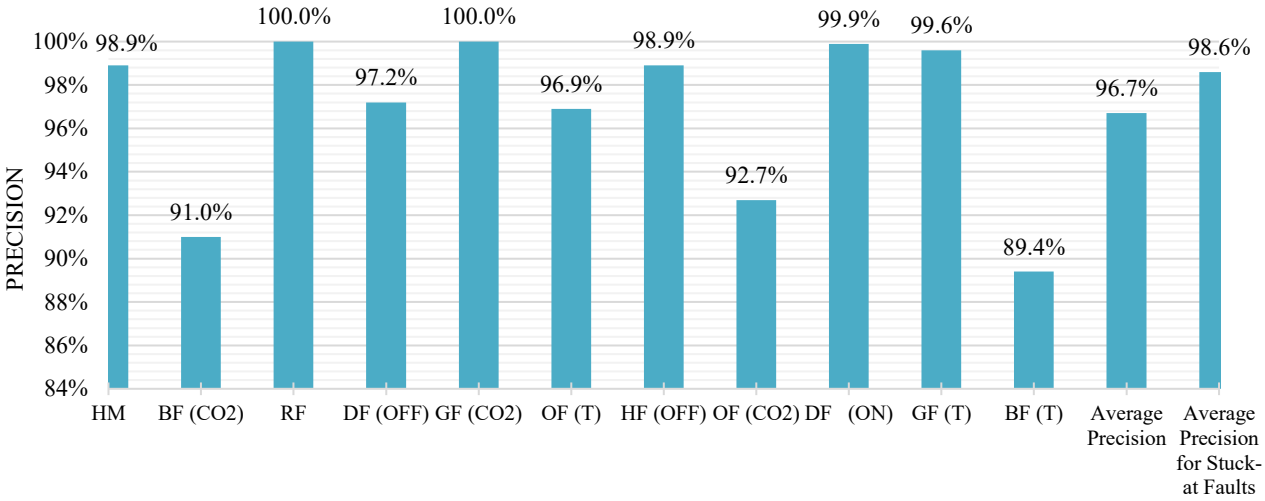


Figure 7.12 Precision for different fault classes in DL.

Figure 7.13 shows recall values calculated for each class of faults [207] in fault diagnosis based on DL. The lowest recall value belongs to the routing fault (0.935), and the highest precision value is 1 for the damper fault stuck at off status, offset fault in the CO_2 sensor, and gain fault in the temperature sensor. The average recall for all fault types is 0.982, while the average recall for the stuck-at faults, including damper fault in closed status, heater fault in off status, and damper fault in open status, is 0.984.

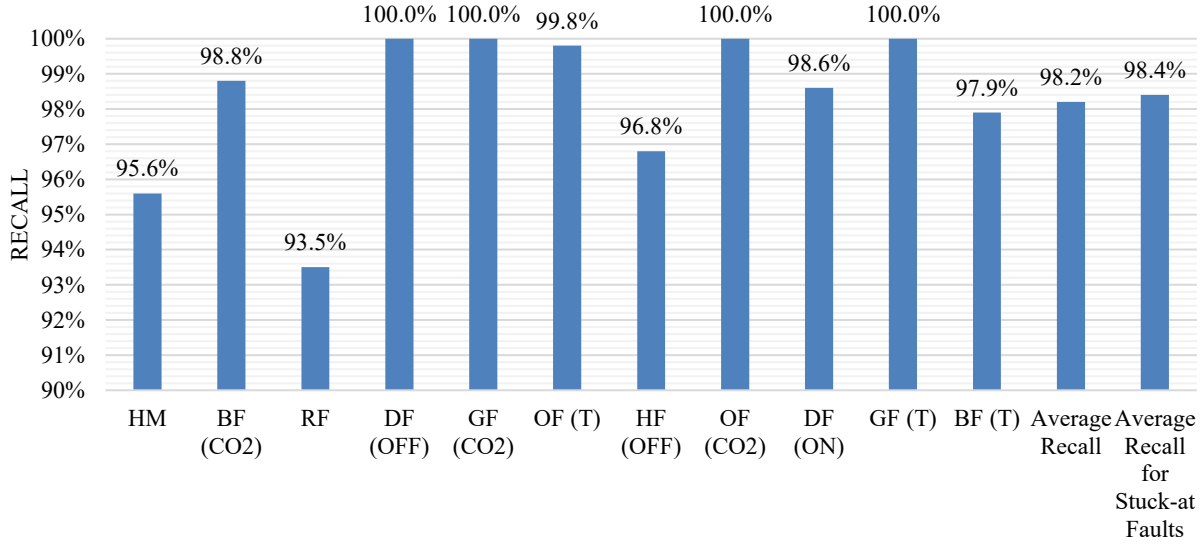


Figure 7.13 Recall for different fault classes in DL.

Figure 7.14 presents the F-score [207] values of each fault class. The lowest F-score value belongs to the battery fault in the temperature sensor (0.9345), and the highest F-score value is 1 for the gain fault in the CO_2 sensor. The average F-score for all fault types is 0.9746, while the average F-score for the stuck-at faults, including damper fault in closed status, heater fault in off status, and damper fault in open status, is 0.9855.

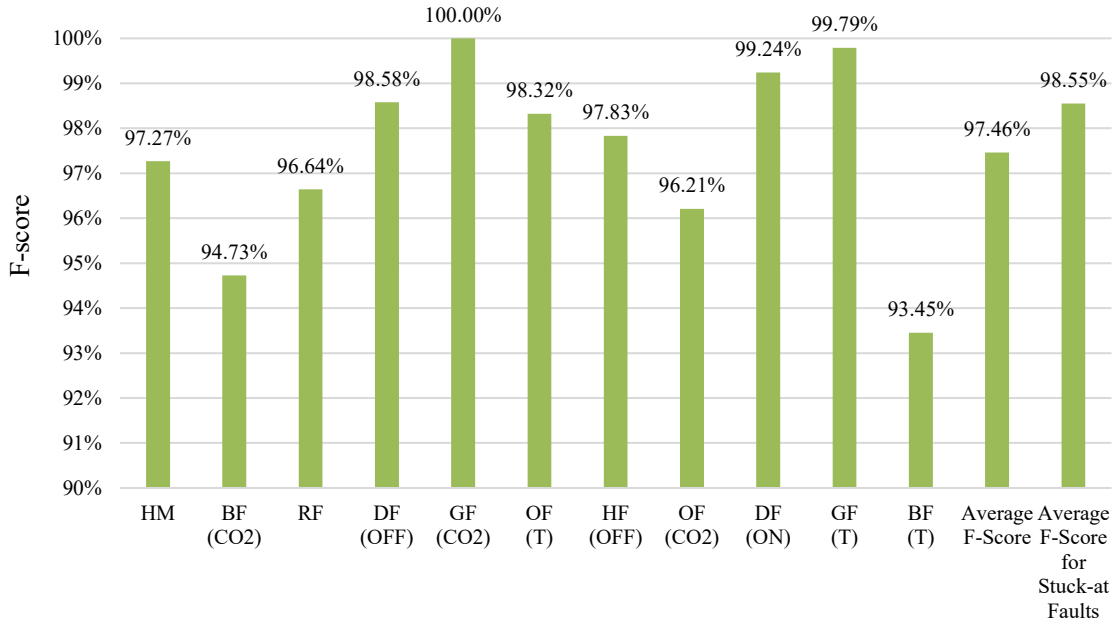


Figure 7.14 F-score for different fault classes in DL.

7.3. DISCUSSION ON RESULTS

This section compares the results of the composed diagnostic classifier based on the causal relations in an FBBN using RDPs to data-driven classification based on a deep neural network. Table 7.8 shows the comparison results, which shows that the performance of the composed diagnostic classifier is as good as data-driven classification based on deep neural network, while benefiting from the advantages of the knowledge-driven methods such as transparency of the process, understandability besides advantages of the data-driven methods such as automatic and low effort classification. In this table, the performance of the data-driven diagnostic classifier is recalculated for stuck-at faults that are comparable with the composed diagnostic classifier's performance.

The accuracy of the composed diagnostic classifier is measured at 97.28% compared to the overall accuracy of the data-driven diagnostic classifier (97.40%), which shows a reasonable accuracy.

The precision and F-score values show the superiority of the presented composed diagnostic classifier over the data-driven diagnostic classifier. The precision of the composed diagnostic classifier is 100% while it is 96.70% for the overall precision of the data-driven diagnostic classifier and 98.60% for the precision of the data-driven diagnostic classifier for stuck-at or constant faults that means the composed diagnostic classifier diagnosed items are more relevant than the data-driven diagnostic classifier. The F-score of the composed diagnostic classifier indicated the values of 98.56%. In comparison, 97.46% for the overall F-score of the data-driven diagnostic classifier and 98.55% for the F-score of the data-driven diagnostic classifier for stuck-at or constant faults that means the harmonic mean of precision and recall (instead of the arithmetic mean) for the composed diagnostic classifier is better than the data-driven diagnostic classifier.

Evaluation results show that the recall value for the data-driven diagnostic classifier (98.20%), specifically for only stuck-at or constant faults (98.40%), is comparable to the recall value of the composed diagnostic classifier (97.22%); however, the recall value of the data-driven diagnostic classifier is better that means the data-driven diagnostic classifier diagnosed more items from the relevant items than the composed diagnostic classifier.

Table 7.8 Comparison of performance metrics in this thesis.

Type of Classifier	Accuracy	Precision	Recall	F1 or F-score
Composed diagnostic classifier – **Overall**	97.28%	100%	97.22%	98.56%
Data-driven diagnostic classifier – **Overall**	97.40%	96.70%	98.20%	97.46%
Composed diagnostic classifier – For only actuator stuck-at faults	100%	100%	100%	100%
Data-driven diagnostic classifier – For only actuator stuck-at faults	NA	98.60%	98.40%	98.55%

The composed diagnostic classifiers are designed and implemented step-by-step and they are one of the white-box approaches, however, the data-driven approaches are mostly categorized as black-box approaches. On the other hand, the data-driven diagnostic classifiers need lower efforts but more computation capacities.

8. CONCLUSION AND FURTHER RESEARCH

This chapter gives an insight into the overall results presented in this dissertation. The related works and the state-of-the-art illustrated the research gaps. The research gaps are from different perspectives: a model which combines both DCV and heating systems for health monitoring and fault diagnosis, a composed fault diagnosis method based on the combination of the knowledge-driven and data-driven fault diagnosis methods.

For the first research gap, a DCV and a heating system are combined and they are analytically modeled and dynamically simulated. This simulation framework is a platform for health monitoring and testing and training failure detection and fault diagnosis methods based on the observed system behavior and output data. Therefore, this thesis introduced the theory and models a DCV and heating system, including three parts of the physical model, DCV and heating system, and modular composability model. This DCV and heating system involves numerous functions, inputs, and outputs and the simulation framework helps in-depth assessment of components functionality and finding effective parameters besides the adaptive thermal control. The results indicate that the developed DCV and heating system can simulate IAQ and comfort parameters in an example scenario of an office building in real-time. The control strategies in this study aim to keep the indoor temperature and the CO_2 concentrations of the office rooms around the setpoint (within the scalable thresholds), despite the variation of the different parameters, e.g., occupants, outside temperature pattern, heating system output power, status or size of air damper, and wind speed. The implementation part of the DCV and heating system is established scalable based on the modular composability scheme. The results showed that the behavior of the system through the simulation maps the analytical model of the system. The investigated system empowers the user to monitor and control the real-time system performance, the duty cycle for the heating system, the frequency of heater ON/OFF switch, and damper open/closed status to identify maintenance problems, failure detection, and fault diagnosis, especially in the design phase. The user can change various parameters and thresholds to monitor the system operation with desired values in the MATLAB/Simulink model workspace and find the optimum set points. The system model is established based on wired and wireless networks. A fault injection framework with a GUI allows to trace the behavior of the system in the presence of different kinds of faults and to understand the effect of different types of faults in components or communication networks on energy consumption and occupancy comfort based on the fault-error-failure propagation model in component and system level.

This thesis further solved the research problem of failure detection and fault diagnosis of the DCV and heating systems as a composed fault diagnosis method that is independent of the historical data, independent of the expert knowledge, and computing-resource efficient. The building blocks and GUIs are developed for evaluating the robustness against various faults and failures by artificially injecting different types of faults. The data produced in this framework is used to test various diagnostic classifiers, e.g., data-driven and composed diagnostic classifiers. The simulated model can be connected to the data server using SQL for the data logging. The developed fault diagnosis method in this thesis based on a composed diagnostic classifier formulates the fault diagnosis problem in a knowledge-driven diagnostic system based on fuzzy theory and Bayesian inference that is combined with data-driven classifiers when maintaining the advantages of the knowledge-driven based and data-driven methods. The composed diagnostic classifier uses conditional probability to show the direction of dependency of different subdomains against the literature that used it to calculate the posterior probability of a fault given observed fault symptoms (based on the expert knowledge) and relies on the causal relations in a fuzzy Bayesian belief network using relation direction probabilities to lower the fault diagnosis efforts and energy waste as well as

to increase the operational efficiency. This method also reveals the hidden dependencies in signals over time in case of faults and reveals the hidden dependencies of multiple fault implications, e.g., a damper stuck-at open fault causes over ventilation that causes room temperature decrease that means the increase of heating load that causes the heater stuck at ON which itself is another fault implication. This novel composed diagnostic classification method is developed for specific component fault types, e.g., stuck-at or constant faults and integrates causal relations using the relation direction probabilities and fuzzy theory into the Bayesian belief networks. The RDPs were used to update the fault library. This research considers four types of stuck-at or constant faults from different components: temperature sensor, CO_2 sensor, heater actuator, and damper actuator with various fault values in different time instances. The evaluation result for the FBBN shows that this method can overall diagnose 97.22% of faults truly (TPs over the whole five cumulative ranks). From 97.22%, 90.55% of faults were diagnosed at the first rank, 4.44% were diagnosed at the second rank, 1.66% were diagnosed at the third and fourth rank. The performance metrics are used to evaluate the effectiveness of the ranking of diagnostic results using probability values. The average accuracy of the diagnosis method shows the cumulative accuracy from only considering the diagnosis results from the first rank (90.76%) increased to 97.28% when the fault diagnosis classifier considers the top five ranks. The average recall of the diagnosis method shows the cumulative recall from only considering the diagnosis results from the first rank (90.55%) increased to 97.22% when the fault diagnosis classifier considers the top five ranks. The average precision of the diagnosis method shows the cumulative precision is always 100%. The average F-score or F1 of the diagnosis method shows the cumulative F-score from only considering the diagnosis results from the first rank (94.76%) increased to 98.57% when the fault diagnosis classifier considers the top five ranks. The overall overview of the performance metrics of the fault diagnosis method introduced in this thesis indicates the average values of the performance metrics with a cumulative basis from only considering the diagnosis results from the first rank increases to when the fault diagnosis classifier considers more top ranks, for example, the diagnosis result with considering the top five ranks is always better than considering only the top two ranks.

Also, data-driven classification is established using deep neural networks to compare and evaluate the performance of the presented composed diagnostic classifier. This framework based on artificial intelligence includes different steps of data acquisition, data preprocessing, network model design, model optimization, and network model evaluation. The evaluation results show that the performance of the introduced composed diagnostic classification based on the causal relations in an FBBN using RDPs is as good as data-driven classification based on deep neural networks. The accuracy of the novel composed diagnostic classifier is measured at 97.28% compared to the overall accuracy of the data-driven diagnostic classifier (97.40%), which shows a reasonable accuracy. The precision of the composed diagnostic classifier indicated the values of 100% while 96.70% for the overall precision of the data-driven diagnostic classifier and 98.60% for the precision of the data-driven diagnostic classifier for only stuck-at or constant faults that means the composed diagnostic classifier diagnosed items are more relevant than the data-driven diagnostic classifier. The F-score of the composed diagnostic classifier indicated the values of 98.56%. In comparison, 97.46% for the overall F-score of the data-driven diagnostic classifier and 98.55% for the F-score of the data-driven diagnostic classifier for only stuck-at or constant faults that means the harmonic mean of precision and recall (instead of the arithmetic mean) for the composed diagnostic classifier is better than data-driven diagnostic classifier. Evaluation results show the recall value for the data-driven diagnostic classifier (98.20%), specifically for only stuck-at or constant faults (98.40%), is comparable to the recall value of the composed diagnostic classifier (97.22%); however, the recall value of the data-driven diagnostic classifier is better that means the data-driven diagnostic classifier diagnosed more

items from the relevant items than the composed diagnostic classifier. The composed diagnostic classifiers are designed and implemented step-by-step and they are one of the white-box approaches, however, the data-driven approaches are mostly categorized as black-box approaches. On the other hand, the data-driven diagnostic classifiers need lower efforts but more computation capacities.

All the contributions mentioned above in this thesis are designed based on a modular composability scheme. The unique modules are used to create various complex building models. This thesis introduces the modular composability scheme using generic diagnostic components at the system level applicable in many areas, especially in cyber-physical systems. Therefore, the introduced model in this thesis can combine modules and understand the composition of diverse subsystems by having an entire perception of its components and combinations. That means the model, thanks to the modular composability algorithm, is scalable. The user can generate different types of buildings with various architectures with many rooms and floors. The system model, fault injection blocks, and diagnostic modules will be automatically extended. This thesis describes a step-by-step composability modeling. This study aimed to develop a generic composable model for the buildings with different architectures for system behavior monitoring and to test the FDFD models.

The future work will be an extension of the introduced fault diagnosis method based on the composed diagnostic classification with consideration of complex systems with many signals and more types of faults extra than the stuck-at faults, fault diagnosis of multiple faults in components, e.g., communication faults.

REFERENCES

[1] M. Maasoumy, “Controlling energy-efficient buildings in the context of smart grid: A cyber physical system approach,” PhD Thesis, University of California, Berkeley, 2013.

[2] International energy agency, “International energy agency-report: World energy outlook,” Paris, France, 2011.

[3] X. Li and R. Yao, “Modelling heating and cooling energy demand for building stock using a hybrid approach,” *Energy and Buildings*, vol. 235, p. 110740, 2021.

[4] EBPD, “On the energy performance of buildings,” pp. 13–34, 2010.

[5] Nikolina Šajn, *Energy efficiency of buildings: A nearly zero-energy future?*: European Parliamentary Research Service (EPRS).

[6] M. Economidou, V. Todeschi, P. Bertoldi, D. D. Agostino, P. Zangheri, and L. Castellazzi, “Review of 50 years of EU energy efficiency policies for buildings,” *Energy and Buildings*, p. 110322, 2020.

[7] U.S. Department of Energy, “Building Energy Data Book,” 2011. [Online]. Available: http:// buildingsdatabook.eere.energy.gov/

[8] P. H. Shaikh, N. B. M. Nor, P. Nallagownden, I. Elamvazuthi, and T. Ibrahim, “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, 2014.

[9] B. Merema, M. Delwati, M. Sourbron, and H. Breesch, “Demand controlled ventilation (DCV) in school and office buildings: Lessons learnt from case studies,” *Energy and Buildings*, vol. 172, pp. 349–360, 2018.

[10] M. Abarkan, N. K. M’Sirdi, and F. Errahimi, “Analysis and simulation of the energy behavior of a building equipped with RESin simscape,” *Energy procedia*, vol. 62, pp. 522–531, 2014.

[11] M. J. Brandemuehl and J. E. Braun, “The impact of demand-controlled and economizer ventilation strategies on energy use in buildings,” *ASHRAE Transactions*, vol. 105, p. 39, 1999.

[12] American Society of Heating, Refrigerating and Air-Conditioning Engineers, *62.1 User’s Manual: ANSI/ASHRAE Standard 62.1-2010: Ventilation for Acceptable Indoor Air Quality*: American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2010.

[13] E. Jeannette and T. Phillips, Eds., *Designing and testing demand controlled ventilation strategies*, 2006.

[14] L. Kajtár and L. Herczeg, “Influence of carbon-dioxide concentration on human well-being and intensity of mental work,” *QJ Hung. Meteorol. Serv*, vol. 116, pp. 145–169, 2012.

[15] S. Lazarova-Molnar, H. R. Shaker, and N. Mohamed, “Fault detection and diagnosis for smart buildings: State of the art, trends and challenges,” in *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*, pp. 1–7.

[16] Y. Zhao, T. Li, X. Zhang, and C. Zhang, “Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future,” *Renewable and Sustainable Energy Reviews*, vol. 109, pp. 85–101, 2019.

[17] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault detection and diagnosis in industrial systems*: Springer Science & Business Media, 2000.
[18] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, “Basic concepts and taxonomy of dependable and secure computing,” *IEEE Trans.Dependable and Secure Comput.*, vol. 1, no. 1, pp. 11–33, 2004, doi: 10.1109/TDSC.2004.2.
[19] A. Avizienis, J.-C. Laprie, and B. Randell, “Fundamental concepts of computer system dependability,” in *Workshop on Robot Dependability: Technological Challenge of Dependable Robots in Human Environments*, pp. 1–16.
[20] G. Gössler and J. Sifakis, “Composition for component-based modeling,” *Science of Computer Programming*, vol. 55, 1-3, pp. 161–183, 2005.
[21] M. D. Petty E. W. Eric, *A theory of simulation composability*.
[22] A. Zakinthinos and E. S. Lee, “The composability of non-interference,” *Journal of Computer Security*, vol. 3, no. 4, pp. 269–281, 1995.
[23] Z. Shi and W. O'Brien, “Development and implementation of automated fault detection and diagnostics for building systems: A review,” *Automation in Construction*, vol. 104, pp. 215–229, 2019, doi: 10.1016/j.autcon.2019.04.002.
[24] A. Behravan, A. Mallak, R. Obermaisser, D. H. Basavegowda, C. Weber, and M. Fathi, “Fault injection framework for fault diagnosis based on machine learning in heating and demand-controlled ventilation systems,” in *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI): Iran University of Science and Technology, Tehran, Iran, Dec. 22nd, 2017*, Tehran, 2017?, pp. 273–279.
[25] S. X. Ding, *Model-based fault diagnosis techniques: Design schemes, algorithms, and tools / Steven X. Ding*. Berlin: Springer, 2008.
[26] S. Katipamula and M. Brambley, “Review Article: Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems—A Review, Part I,” *HVAC&R Res.*, vol. 11, no. 1, pp. 3–25, 2005, doi: 10.1080/10789669.2005.10391123.
[27] A. Behravan, R. Obermaisser, and M. Abboush, “Fault Injection Framework for Demand-Controlled Ventilation and Heating Systems Based on Wireless Sensor and Actuator Networks: BEST PAPER AWARD,” in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, BC, Nov. 2018 - Nov. 2018, pp. 525–531.
[28] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.
[29] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, “A review of process fault detection and diagnosis,” *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 313–326, 2003, doi: 10.1016/S0098-1354(02)00161-8.
[30] M. van Steen and A. S. Tanenbaum, “Distributed systems principles and paradigms,” *Network*, vol. 2, p. 28, 2002.
[31] J.-C. Laprie, “Dependability of computer systems: concepts, limits, improvements,” in *Proceedings of Sixth International Symposium on Software Reliability Engineering. ISSRE'95*, pp. 2–11.

[32] “Mutual information,” [Online]. Available: https://en.wikipedia.org/wiki/Mutual_information
[33] G. Zeng, “A unified definition of mutual information with applications in machine learning,” *Mathematical Problems in Engineering*, vol. 2015, 2015.
[34] C. E. Shannon, “A mathematical theory of communication,” *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.
[35] S. Kullback, *Information theory and statistics*: Courier Corporation, 1997.
[36] T. M. Cover and J. A. Thomas, “Information theory and statistics,” *Elements of Information Theory*, vol. 1, no. 1, pp. 279–335, 1991.
[37] C.-Y. Chiu, C.-C. Lo, and Y.-X. Hsu, “Integrating bayesian theory and fuzzy logics with case-based reasoning for car-diagnosing problems,” in *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, pp. 344–348.
[38] *World energy outlook 2011*. Paris: IEA, International Energy Agency; OECD, 2011.
[39] A. I. Dounis and C. Caraiscos, “Advanced control systems engineering for energy and comfort management in a building environment—A review,” *Renewable and Sustainable Energy Reviews*, vol. 13, 6-7, pp. 1246–1261, 2009.
[40] V. Dimitriou, S. K. Firth, T. M. Hassan, T. Kane, and F. Fouchal, “Developing suitable thermal models for domestic buildings with Smart Home equipment,” 2014.
[41] C. Lapusan, R. Balan, O. Hancu, and C. Rad, “Rapid Control Prototyping in the Development of Home Energy Management Systems,” in *Applied Mechanics and Materials*, 2014, pp. 395–400.
[42] *The MathWorks Inc.*
[43] C. Lapusan, R. Balan, O. Hancu, and A. Plesa, “Development of a multi-room building thermodynamic model using Simscape library,” *Energy procedia*, vol. 85, pp. 320–328, 2016.
[44] S. Karmacharya, G. Putrus, C. Underwood, and K. Mahkamov, “Thermal modelling of the building and its HVAC system using Matlab/Simulink,” in *2012 2nd International Symposium On Environment Friendly Energies And Applications*, 2012, pp. 202–206.
[45] M. M. Gouda, S. Danaher, and C. P. Underwood, “Building thermal model reduction using nonlinear constrained optimization,” *Building and environment*, vol. 37, no. 12, pp. 1255–1265, 2002.
[46] J. A. Crabb, N. Murdoch, and J. M. Penman, “A simplified thermal response model,” *Building Services Engineering Research and Technology*, vol. 8, no. 1, pp. 13–19, 1987.
[47] T. Sulka and N. Jenkins, “Modelling of a housing estate with micro-combined heat and power for power flow studies,” *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, vol. 222, no. 7, pp. 721–729, 2008.
[48] N. Mendes, G. H. C. Oliveira, and H. X. de Araújo, “Building thermal performance analysis by using matlab/simulink,” in *Seventh International IBPSA Conference, Rio de Janeiro, Brazil*, 2001, pp. 473–480.
[49] S. Bertagnolio and J. Lebrun, “Simulation of a building and its HVAC system with an equation solver: application to benchmarking,” in *Building Simulation*, 2008, pp. 234–250.

[50] M. Embaye, R. K. Al-Dadah, and S. Mahmoud, “Effect of flow pulsation on energy consumption of a radiator in a centrally heated building,” *International Journal of Low-Carbon Technologies*, vol. 11, no. 1, pp. 119–129, 2014.
[51] A. Thavlov and H. W. Bindner, “Thermal models for intelligent heating of buildings,” in *4th International Conference on Applied Energy (ICAE 2012): Energy innovations for a sustainable world*, 2012.
[52] L. Danza *et al.,* “A simplified thermal model to control the energy fluxes and to improve the performance of buildings,” *Energy procedia*, vol. 101, pp. 97–104, 2016.
[53] L. R. Petzold, “Description of DASSL: a differential/algebraic system solver,” Sandia National Labs., Livermore, CA (USA), 1982.
[54] H. Bastida, C. E. Ugalde-Loo, M. Abeysekera, M. Qadrdan, and J. Wu, “Thermal Dynamic Modelling and Temperature Controller Design for a House,” *Energy procedia*, vol. 158, pp. 2800–2805, 2019.
[55] W. Mai and C. Y. Chung, “Economic MPC of aggregating commercial buildings for providing flexible power reserve,” *IEEE transactions on power systems*, vol. 30, no. 5, pp. 2685–2694, 2014.
[56] J. Li, J. Wall, and G. Platt, Eds., *Indoor air quality control of HVAC system*, 2010.
[57] A. Standard, “62.1,“Ventilation for Acceptable Indoor Air Quality”,” *American Society of Heating, Refrigerating and Air conditioning Engineers*, 2004.
[58] S. P. Mohanty, *Nanoelectronic mixed-signal system design*: McGraw-Hill Education New York, 2015.
[59] M. L. Rajaram, “Comparative analysis and implementation of high data rate wireless sensor network simulation frameworks,” University of North Texas.
[60] A. Pinto, M. D’Angelo, C. Fischione, E. Scholte, and A. Sangiovanni-Vincentelli, “Synthesis of embedded networks for building automation and control,” in *2008 American Control Conference*, 2008, pp. 920–925.
[61] A. Guinard, A. McGibney, and D. Pesch, “A wireless sensor network design tool to support building energy management,” in *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, 2009, pp. 25–30.
[62] Z. Song and X. Zhou, “Research and simulation of wireless sensor and actuator networked control system,” in *2013 25th Chinese Control and Decision Conference (CCDC)*, 2013, pp. 3995–3998.
[63] E. Callaway *et al.,* “Home networking with IEEE 802.15. 4: a developing standard for low-rate wireless personal area networks,” *IEEE Communications magazine*, vol. 40, no. 8, pp. 70–77, 2002.
[64] J.-J. Wang and S. Wang, “Wireless sensor networks for Home Appliance Energy Management based on ZigBee technology,” in *2010 International Conference on Machine Learning and Cybernetics*, pp. 1041–1046.
[65] N. H. A. Aziz, S. S. Sarnin, K. A. Othman, N. Wahab, N. Misman, and A. T. Hashim, “Supervisory data acquisition of temperature and humidity in oil palm tissue culture

laboratory," in *2010 Second International Conference on Computational Intelligence, Modelling and Simulation*, pp. 470–475.

[66] O. Hyncica, P. Kacz, P. Fiedler, Z. Bradac, P. Kucera, and R. Vrba, Eds., *The Zigbee experience*, 2006.

[67] R. Shoureshi and K. McLaughlin, "Microprocessor-Based Failure Detection of Heat Pumps," *IFAC Proceedings Volumes*, vol. 18, no. 9, pp. 155–160, 1985, doi: 10.1016/S1474-6670(17)60277-5.

[68] P. B. Usoro, I. C. Schick, and S. Negahdaripour, *HVAC SYSTEM FAULT DETECTION AND DIAGNOSIS* (in English (US)): IEEE.

[69] Z. Gao, C. Cecati, and S. X. Ding, "A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis With Model-Based and Signal-Based Approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, 2015, doi: 10.1109/TIE.2015.2417501.

[70] Q. Zhou, S. Wang, and Z. Ma, "A model-based fault detection and diagnosis strategy for HVAC systems," *International Journal of Energy Research*, vol. 33, no. 10, pp. 903–918, 2009, doi: 10.1002/er.1530.

[71] J. Schein, S. T. Bushby, N. S. Castro, and J. M. House, "A rule-based fault detection method for air handling units," *Energy and Buildings*, vol. 38, no. 12, pp. 1485–1492, 2006.

[72] S. H. Lee and F. W. Yik, "A study on the energy penalty of various air-side system faults in buildings," *Energy and Buildings*, vol. 42, no. 1, pp. 2–10, 2010.

[73] R. Sterling, G. Provan, J. Febres, D. O'Sullivan, P. Struss, and M. M. Keane, "Model-based fault detection and diagnosis of air handling units: A comparison of methodologies," *Energy procedia*, vol. 62, pp. 686–693, 2014.

[74] M. Basarkar, X. Pang, L. Wang, P. Haves, and T. Hong, "Modeling and simulation of HVAC faults in EnergyPlus," Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2011.

[75] E. A. Lee and S. A. Seshia, *Introduction to embedded systems: A cyber-physical systems approach*. Cambridge, Massachuetts, London, England: MIT Press, 2017.

[76] J. Boi-Ukeme, C. Ruiz-Martin, and G. Wainer, "Real-Time Fault Detection and Diagnosis of CPS Faults in DEVS," in *2020 IEEE 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Application (DependSys)*, pp. 57–64.

[77] E. Bartocci, N. Manjunath, L. Mariani, C. Mateis, and D. Ničković, "Automatic failure explanation in CPS models," in *International Conference on Software Engineering and Formal Methods*, pp. 69–86.

[78] S. Wu, "System-level monitoring and diagnosis of building HVAC system," UC Merced, 2013.

[79] S. R. West, Y. Guo, X. R. Wang, and J. Wall, "Automated fault detection and diagnosis of HVAC subsystems using statistical machine learning," in *12th International Conference of the International Building Performance Simulation Association*, 2011.

[80] M. C. Comstock, J. E. Braun, and R. Bernhard, *Development of analysis tools for the evaluation of fault detection and diagnostics in chillers*: Purdue University, 1999.

[81] J. Wen and S. Li, “Tools for evaluating fault detection and diagnostic methods for air-handling units,” *ASHRAE RP-1312 Final Report, American Society of Heating, Refrigerating and Air Conditioning Engineers Inc., Atlanta*, 2011.

[82] M. łgorzata Steinder and A. S. Sethi, “A survey of fault localization techniques in computer networks,” *Science of Computer Programming*, vol. 53, no. 2, pp. 165–194, 2004.

[83] E. Akgün and M. Demir, “Modeling Course Achievements of Elementary Education Teacher Candidates with Artificial Neural Networks,” *International Journal of Assessment Tools in Education*, pp. 491–509, 2018, doi: 10.21449/ijate.444073.

[84] M. Ahmed, N. Mehmood, A. Nadeem, A. Mehmood, and K. Rizwan, “Fall Detection System for the Elderly Based on the Classification of Shimmer Sensor Prototype Data,” *Healthcare informatics research*, vol. 23, no. 3, pp. 147–158, 2017, doi: 10.4258/hir.2017.23.3.147.

[85] A. Behravan, M. Abboush, and R. Obermaisser, “Deep Learning Application in Mechatronics Systems’ Fault Diagnosis, a Case Study of the Demand-Controlled Ventilation and Heating System,” in *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, Dubai, United Arab Emirates, Mar. 2019 - Apr. 2019, pp. 1–6.

[86] Wikipedia, *Deep Learning.* [Online]. Available: https://en.wikipedia.org/wiki/Deep_learning (accessed: Jan. 18 2020).

[87] A. L'Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, “Machine Learning With Big Data: Challenges and Approaches,” *IEEE Access*, vol. 5, pp. 7776–7797, 2017, doi: 10.1109/ACCESS.2017.2696365.

[88] G. Hong and B. S. Kim, “Development of a Data-Driven Predictive Model of Supply Air Temperature in an Air-Handling Unit for Conserving Energy,” *Energies*, vol. 11, no. 2, p. 407, 2018, doi: 10.3390/en11020407.

[89] M. Chokesuwattanaskul, “Optimization and neural network modeling for HVAC systems,” SIU SS SOT-PhD 2007-02, 2007.

[90] D. Lee, B. Lee, and J. W. Shin, “Fault detection and diagnosis with modelica language using deep belief network,” in *Proceedings of the 11th International Modelica Conference, Versailles, France, September 21-23, 2015*, 2015, pp. 615–623.

[91] A. Beghi, L. Cecchinato, C. Corazzol, M. Rampazzo, F. Simmini, and G. A. Susto, “A One-Class SVM Based Tool for Machine Learning Novelty Detection in HVAC Chiller Systems,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 1953–1958, 2014, doi: 10.3182/20140824-6-za-1003.02382.

[92] D. Dehestani, F. Eftekhari, Y. Guo, S. Ling, S. Su, and H. Nguyen, “Online Support Vector Machine Applicationfor Model Based Fault Detection and Isolationof HVAC System,” *IJMLC*, pp. 66–72, 2011, doi: 10.7763/IJMLC.2011.V1.10.

[93] S. Wang and Y. Chen, “Fault-tolerant control for outdoor ventilation air flow rate in buildings based on neural network,” *Building and environment*, vol. 37, no. 7, pp. 691–704, 2002, doi: 10.1016/S0360-1323(01)00076-2.

[94] Z. Yang, Y. Wang, and J. Lv, “Survey of modern fault diagnosis methods in networks,” in *2012 International Conference on Systems and Informatics (ICSAI2012)*, pp. 1640–1643.

[95] S. Qiu, A. M. Agogino, S. Song, J. Wu, and S. Sitarama, “A fusion of bayesian and fuzzy analysis for print faults diagnosis,” in *Computers and Their Applications*, pp. 229–232.
[96] H. Tang and S. Liu, “Basic Theory of Fuzzy Bayesian Networks and Its Application in Machinery Fault Diagnosis,” in *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*.
[97] J. Pearl, *Probabilistic reasoning in intelligent systems: Networks of plausible inference / Judea Pearl*. San Francisco: Morgan Kaufmann, 2014?
[98] W. Yu and C. Zhao, “Online fault diagnosis for industrial processes with Bayesian network-based probabilistic ensemble learning strategy,” *IEEE Trans. Automat. Sci. Eng.*, vol. 16, no. 4, pp. 1922–1932, 2019.
[99] M. Hu *et al.,* “A machine learning bayesian network for refrigerant charge faults of variable refrigerant flow air conditioning system,” *Energy and Buildings*, vol. 158, pp. 668–676, 2018.
[100] Di Peng, Z. Geng, and Q. Zhu, “A multilogic probabilistic signed directed graph fault diagnosis approach based on Bayesian inference,” *Industrial & Engineering Chemistry Research*, vol. 53, no. 23, pp. 9792–9804, 2014.
[101] C.-Y. Chiu, C.-C. Lo, and Y.-X. Hsu, “Integrating bayesian theory and fuzzy logics with case-based reasoning for car-diagnosing problems,” in *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, pp. 344–348.
[102] R. J. Kuo, C. L. Cha, S. H. Chou, C. W. Shih, and C. Y. Chiu, “Integration of ant algorithm and case based reasoning for knowledge management,” in *Proceedings of International Conference on IJIE*, pp. 10–12.
[103] L. A. Zadeh, “Information and control,” *Fuzzy sets*, vol. 8, no. 3, pp. 338–353, 1965.
[104] S. Gottwald, “An early approach toward graded identity and graded membership in set theory,” *Fuzzy Sets and Systems*, vol. 161, no. 18, pp. 2369–2379, 2010, doi: 10.1016/j.fss.2009.12.005.
[105] J. Y. Yao, J. Li, H. Li, and X. Wang, “Modeling system based on fuzzy dynamic Bayesian network for fault diagnosis and reliability prediction,” in *2015 Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–6.
[106] R. Intan and O. Y. Yuliana, “Fuzzy bayesian belief network for analyzing medical track record,” in *Advances in Intelligent Information and Database Systems*: Springer, 2010, pp. 279–290.
[107] J. Cheng, D. Bell, and W. Liu, “Learning Bayesian networks from data: An efficient approach based on information theory,” *On World Wide Web at http://www. cs. ualberta. ca/~ jcheng/bnpc. htm*, 1998.
[108] F. M. Mele, *A model-based approach to HVAC fault detection and diagnosis*.
[109] J. Shiozaki and F. Miyasaka, Eds., *A fault diagnosis tool for HVAC systems using qualitative reasoning algorithms*, 1999.
[110] Z. Shi, W. O'Brien, and H. B. Gunay, “Development of a distributed building fault detection, diagnostic, and evaluation system,” *ASHRAE Transactions*, vol. 124, no. 2, pp. 23–38, 2018.

[111] Yang Zhao, Fu Xiao, and Shengwei Wang, “An intelligent chiller fault detection and diagnosis methodology using Bayesian belief network,” *Energy and Buildings*, vol. 57, pp. 278–288, 2013, doi: 10.1016/j.enbuild.2012.11.007.

[112] Fu Xiao, Yang Zhao, Jin Wen, and Shengwei Wang, “Bayesian network based FDD strategy for variable air volume terminals,” *Automation in Construction*, vol. 41, pp. 106–118, 2014, doi: 10.1016/j.autcon.2013.10.019.

[113] Y. Zhao, J. Wen, F. Xiao, X. Yang, and S. Wang, “Diagnostic Bayesian networks for diagnosing air handling units faults – part I: Faults in dampers, fans, filters and sensors,” *Applied Thermal Engineering*, vol. 111, pp. 1272–1286, 2017, doi: 10.1016/j.applthermaleng.2015.09.121.

[114] A. Taal and L. Itard, “Fault detection and diagnosis for indoor air quality in DCV systems: Application of 4S3F method and effects of DBN probabilities,” *Building and environment*, vol. 174, p. 106632, 2020.

[115] Baoping Cai *et al.,* “Multi-source information fusion based fault diagnosis of ground-source heat pump using Bayesian network,” *Applied Energy*, vol. 114, pp. 1–9, 2014, doi: 10.1016/j.apenergy.2013.09.043.

[116] T. Gao, “Integrated building fault detection and diagnosis using data modeling and Bayesian networks,” Ecole nationale supérieure Mines-Télécom Lille Douai, 2020.

[117] M. Amin, “Fault detection and root cause diagnosis using dynamic Bayesian network,” Memorial University of Newfoundland, 2018.

[118] X. J. Luo, K. F. Fong, Y. J. Sun, and M. K. Leung, “Development of clustering-based sensor fault detection and diagnosis strategy for chilled water system,” *Energy and Buildings*, vol. 186, pp. 17–36, 2019.

[119] J. Leonard, *Systems engineering fundamentals: Supplementary text*: DIANE Publishing, 1999.

[120] OpenEI developed and maintained by the US National Renewable Energy Laboratory, *Definition: Analytical Modeling.* [Online]. Available: https://openei.org/wiki/Definition: Analytical_Modeling (accessed: May 13 2019).

[121] K. K. Andersen, H. Madsen, and L. H. Hansen, “Modelling the heat dynamics of a building using stochastic differential equations,” *Energy and Buildings*, vol. 31, no. 1, pp. 13–24, 2000.

[122] P. Bacher and H. Madsen, “Identifying suitable models for the heat dynamics of buildings,” *Energy and Buildings*, vol. 43, no. 7, pp. 1511–1522, 2011.

[123] A. K. Athienitis and W. O’Brien, *Modeling, design, and optimization of net-zero energy buildings*: Wiley Online Library, 2015.

[124] A. Foucquier, S. Robert, F. Suard, L. Stéphan, and A. Jay, “State of the art in building modelling and energy performances prediction: A review,” *Renewable and Sustainable Energy Reviews*, vol. 23, pp. 272–288, 2013.

[125] S. Katipamula and M. Brambley, “Review Article: Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems—A Review, Part II,” *HVAC&R Res.*, vol. 11, no. 2, pp. 169–187, 2005, doi: 10.1080/10789669.2005.10391133.

[126] James E. Braun, “Reducing energy costs and peak electrical demand through optimal control of building thermal storage: Summary for Policymakers,” in *ASHRAE transactions 1989*.

[127] J. Braun and N. Chaturvedi, “An Inverse Gray-Box Model for Transient Building Load Prediction,” *HVAC&R Res.*, vol. 8, no. 1, pp. 73–99, 2002, doi: 10.1080/10789669.2002.10391290.

[128] B. Sun, P. B. Luh, Q.-S. Jia, Z. O'Neill, and F. Song, “Building Energy Doctors: An SPC and Kalman Filter-Based Method for System-Level Fault Detection in HVAC Systems,” *IEEE Trans. Automat. Sci. Eng.*, vol. 11, no. 1, pp. 215–229, 2014, doi: 10.1109/TASE.2012.2226155.

[129] A. Persily, S. Emmerich, and M. Taylor, *Simulations of indoor air quality and ventilation impacts of demand controlled ventilation in commercial and institutional buildings*: US Department of Commerce, National Institute of Standards and Technology, 2003.

[130] C. Zayane, “Identification d’un modèle de comportement thermique de bâtiment à partir de sa courbe de charge,”

[131] M. Maasoumy, B. Moridian, M. Razmara, M. Shahbakhti, and A. Sangiovanni-Vincentelli, Eds., *Online simultaneous state estimation and parameter adaptation for building predictive control*, 2013.

[132] T. L. Bergman, F. P. Incropera, D. P. DeWitt, and A. S. Lavine, *Fundamentals of heat and mass transfer*: John Wiley & Sons, 2007.

[133] L. Evangelisti, C. Guattari, and P. Gori, “Energy retrofit strategies for residential building envelopes: An Italian case study of an early-50s building,” *Sustainability*, vol. 7, no. 8, pp. 10445–10460, 2015.

[134] M. G. Davies, “Optimum design of resistance and capacitance elements in modelling a sinusoidally excited building wall,” *Building and environment*, vol. 18, 1-2, pp. 19–37, 1983.

[135] Q. Zhou, S. Wang, X. Xu, and F. Xiao, “A grey-box model of next-day building thermal load prediction for energy-efficient control,” *International Journal of Energy Research*, vol. 32, no. 15, pp. 1418–1431, 2008.

[136] G. Horváth, *„Neural Networks in Systems Identification”,(Chapter 4. in: S. Ablameyko, L. Goras, M. Gori and V. Piuri (Eds.) Neural Networks in Measurement Systems) NATO ASI*: IOS Press,(Amsterdam, The Netherlands).

[137] T. Dewson, B. Day, and A. D. Irving, “Least squares parameter estimation of a reduced order thermal model of an experimental building,” *Building and environment*, vol. 28, no. 2, pp. 127–137, 1993.

[138] A. Malone, *The Windcatcher House: Architectural Record: Building for Social Change*: McGraw-Hill.

[139] S. Roaf, *The Windcatchers of Yazd, Phd*: Oxford Polytechnic.

[140] H. B. Awbi, *Ventilation of buildings,* 2nd ed.: Spon Press, 2003.

[141] D. W. Etheridge and M. Sandberg, *Building ventilation: theory and measurement*: John Wiley & Sons Chichester, 1996.

[142] B. BS5925, "Code of practice for ventilation principles and designing for natural ventilation," *British Standard Institution*, 1995.
[143] ANSI/ASHRAE, *Standard 140-2004, standard method of test for the evaluation of building energy analysis computer programs*.
[144] *Ventilation for Acceptable Indoor Air Quality,* ANSI/ASHRAE, 2016.
[145] P. Stabat, M. Caciolo, and D. Marchio, "Progress on single-sided ventilation techniques for buildings," *Advances in Building Energy Research*, vol. 6, no. 2, pp. 212–241, 2012.
[146] Chartered Institution of Building Services Engineers, *Natural ventilation in non-domestic buildings*. BRE Digest 399. London: Chartered Institution of Building Services Engineers, 1994.
[147] H. B. Awbi, "Air movement in naturally-ventilated buildings," *Renewable Energy*, vol. 8, 1-4, pp. 241–247, 1996.
[148] N. Pavelchak *et al.,* "Identification of factors that disrupt negative air pressurization of respiratory isolation rooms," *Infection Control & Hospital Epidemiology*, vol. 21, no. 3, pp. 191–195, 2000.
[149] J. Hou, Y. Sun, X. Kong, P. Wang, Q. Zhang, and J. Sundell, Eds., *Single and Multiple Zone Methods to Calculate Air Change Rate in Apartments*: Elsevier, 2015.
[150] A. ASHRAE and S. ASHRAE, "Standard 62.1-2016: Ventilation for Acceptable Indoor Air Quality, American Society of Heating Refrigeration and Air-Conditioning Engineers," *Inc., Atlanta*, 2016.
[151] X. Lu, T. Lu, and M. Viljanen, "Estimation of space air change rates and CO2 generation rates for mechanically-ventilated buildings," *Advances in Computer Science and Engineering*, pp. 237–260, 2011.
[152] A. ASTM, "D6245-12, Standard guide for using indoor carbon dioxide concentrations to evaluate indoor air quality and ventilation," *West Conshohocken, PA*, 2012.
[153] S. J. Emmerich and A. K. Persily, *State-of-the-art review of CO2 demand controlled ventilation technology and application*. NISTIR 672: Citeseer, 2001.
[154] A. Persily and L. de Jonge, "Carbon dioxide generation rates for building occupants," *Indoor air*, vol. 27, no. 5, pp. 868–879, 2017.
[155] ESDU TN 07007, *Incompressible flow through orifice plates – a review of the data in the literature*.
[156] P. Heiselberg, K. Svidt, and P. V. Nielsen, Eds., *Windows-measurements of air flow capacity*, 2000.
[157] D. J. Tritton, *Physical fluid dynamics*: Springer Science & Business Media, 2012.
[158] D. Etheridge, *Natural ventilation of buildings: theory, measurement and design*: John Wiley & Sons, 2011.
[159] S. K. Verma, K. Kumar, and H. Kaur, "Estimation of Coefficient of Pressure in High Rise Buildings Using Artificial Neutral Network," *Int. J. Eng. Res. Appl*, vol. 4, pp. 105–110, 2014.
[160] H. Awbi, Ed., *Basic concepts for natural ventilation of buildings*, 2010.

[161] C. Y. Shaw, “Methods for estimating air change rates and sizing mechanical ventilation systems for houses,” *Building Research Note*, vol. 237, p. 27, 1985.

[162] F. Hall, *Building Services and Equipment,* 3rd ed.: Routledge, 2014.

[163] A. Behravan, N. Tabassam, O. Al-Najjar, and R. Obermaisser, “Composability Modeling for the Use Case of Demand-controlled Ventilation and Heating System,” in *IEEE 2019 – 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 1998–2003.

[164] O. Balci, J. D. Arthur, and W. F. Ormsby, “Achieving reusability and composability with a simulation conceptual model,” *Journal of Simulation*, vol. 5, no. 3, pp. 157–165, 2011.

[165] P. K. Davis and R. H. Anderson, “Improving the composability of department of defense models and simulations,” RAND CORP SANTA MONICA CA, 2003.

[166] Y. Zhu, W. Wang, and D. Zhou, “Conceptual framework of composable simulation using multilevel model specification for complex systems,” in *2008 Asia Simulation Conference-7th International Conference on System Simulation and Scientific Computing*, 2008, pp. 84–88.

[167] P. Attie, E. Baranov, S. Bliudze, M. Jaber, and J. Sifakis, “A general framework for architecture composability,” *Formal Aspects of Computing*, vol. 28, no. 2, pp. 207–231, 2016.

[168] A. B. Bondi, “Characteristics of scalability and their impact on performance,” in *Proceedings of the 2nd international workshop on Software and performance*, 2000, pp. 195–203.

[169] M. D. Hill, “What is scalability?,” *ACM SIGARCH Computer Architecture News*, vol. 18, no. 4, pp. 18–21, 1990.

[170] I. Škrjanc, B. Zupančič, B. Furlan, and A. Krainer, “Theoretical and experimental fuzzy modelling of building thermal dynamic response,” *Building and environment*, vol. 36, no. 9, pp. 1023–1038, 2001.

[171] S. F. Barrett and D. J. Pack, “Microcontrollers fundamentals for engineers and scientists,” *Synthesis lectures on digital circuits and systems*, vol. 1, no. 1, pp. 1–124, 2005.

[172] A. Behravan, R. Obermaisser, and A. Nasari, “Thermal dynamic modeling and simulation of a heating system for a multi-zone office building equipped with demand controlled ventilation using MATLAB/Simulink,” in *2017 International Conference on Circuits, System and Simulation (ICCSS)*, 2017, pp. 103–108.

[173] J. Kuutti, P. Saarikko, and R. E. Sepponen, “Real time building zone occupancy detection and activity visualization utilizing a visitor counting sensor network,” in *2014 11th International Conference on Remote Engineering and Virtual Instrumentation (REV): 26-28 Feb. 2014, Porto, Portugal*, 2014.

[174] R. Waeber, H.-U. Wanner, and L. Carlucci, *Luftqualität in Innenräumen: Aussenluftverunreinigung, Quellen in Innenräumen, Gesundheit, Massnahmen*: Bundesamt für Umwelt, Wald und Landschaft, 1997.

[175] *Ventilation for non -residential buildings - Performance requirements for ventilation and room-conditioning systems,* European Standard DIN EN 13779: 2007-09.

[176] DIPL.-ING. DR. BOOS, R/DIPL.-ING. DAMBERGER, B/DIPL.-ING. DR. HUTTER, H-P/UNIV.-PROF. DR. KUNDI, M/DR. MOSHAMMER, H/DIPL.-ING. TAPPLER, P/DIPL.-

ING. TWRDIK, F/DR. WALLNER, P, *Bewertung der Innenraumluft - Physikalische Faktoren Kohlenstoffdioxid als Lüftungsparameter*. Aktualisierte Fassung.

[177] S. Alvarez, *Natural ventilation in buildings: a design handbook*: Earthscan, 1998.

[178] F. Xia, X. Kong, and Z. Xu, "Cyber-Physical Control Over Wireless Sensor and Actuator Networks with Packet Loss," in *Wireless networking based control*, S. K. Mazumder, Ed., New York, London: Springer, 2010, pp. 85–102.

[179] Y. Sadi, S. C. Ergen, and P. Park, "Minimum Energy Data Transmission for Wireless Networked Control Systems," *IEEE Trans. Wireless Commun.*, vol. 13, no. 4, pp. 2163–2175, 2014, doi: 10.1109/TWC.2014.0204.131204.

[180] A. W. Al-Dabbagh and T. Chen, "Design Considerations for Wireless Networked Control Systems," *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5547–5557, 2016, doi: 10.1109/TIE.2016.2564950.

[181] M. Ivanovich and D. Gustavson, "Intelligent buildings is now," *Heating, piping and air conditioning*, vol. 71, no. 5, pp. 73–79, 1999.

[182] A. Cervin, D. Henriksson, and M. Ohlin, "TrueTime 2.0 beta-Reference manual," *Department of Automatic Control, Lund University*, 2010.

[183] S. Balasubramaniyan, B. Subathra, R. C. Hemesh, S. Gurusamy, and S. Srinivasan, "On simulating processor schedules and network protocols within CPS using TrueTime," in *IEEE ICCIC 2015: 2015 IEEE International Conference on Computational Intelligence and Computing Research : 2015 December 10th to 12th : Vickram College of Engineering, Enathi, Madurai - 630 561 (Sivagangai Dist)., Tamil Nadu, India*, Madurai, India, 2016, pp. 1–6.

[184] T. Obaid, H. Rashed, A. Abou-Elnour, M. Rehan, M. M. Saleh, and M. Tarique, "ZigBee Technology and its application in Wireless Home Automation systems: a survey," *International Journal of Computer Networks & Communications*, vol. 6, no. 4, p. 115, 2014.

[185] V. Moravcevic, M. Tucic, R. Pavlovic, and A. Majdak, "An approach for uniform representation and control of ZigBee devices in home automation software," in *2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Berlin, Germany, Sep. 2015 - Sep. 2015, pp. 237–239.

[186] K. Sohraby, D. Minoli, and T. Znati, *Wireless sensor networks: Technology, protocols, and applications*. Hoboken, NJ: Wiley, 2007. [Online]. Available: http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10278611

[187] W. C. Craig, "Zigbee: Wireless control that simply works," *Zigbee Alliance ZigBee Alliance*, 2004.

[188] S. Kasputis and H. C. Ng, "Composable simulations," in *2000 Winter Simulation Conference Proceedings (Cat. No. 00CH37165)*, pp. 1577–1584.

[189] C. Kobyrn, "Will UML 2.0 be agile or awkward?," *Association for Computing Machinery. Communications of the ACM*, vol. 45, no. 1, p. 107, 2002.

[190] S. B. Gupta and A. Mittal, *Introduction to database management system*: Laxmi Publications, Ltd, 2009.

[191] M. Chatham, *Structured Query Language By Example-Volume I: Data Query Language*: Lulu. com, 2012.
[192] J. W. Lloyd, “Practical Advtanages of Declarative Programming,” in *GULP-PRODE (1)*, pp. 18–30.
[193] A. B. Sharma, L. Golubchik, and R. Govindan, “Sensor faults: Detection methods and prevalence in real-world datasets,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 3, p. 23, 2010.
[194] A. Behravan, R. Obermaisser, D. H. Basavegowda, and S. Meckel, “Automatic model-based fault detection and diagnosis using diagnostic directed acyclic graph for a demand-controlled ventilation and heating system in Simulink,” in *2018 Annual IEEE International Systems Conference (SysCon)*, Vancouver, BC, Apr. 2018 - Apr. 2018, pp. 1–7.
[195] J. Hyvarinen, “IEA Annex 25 Final Report,” *VTT, Espoo, Finland*, 1997.
[196] A. Mahapatro and P. M. Khilar, “Fault diagnosis in wireless sensor networks: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2000–2026, 2013.
[197] L. Paradis and Q. Han, “A survey of fault management in wireless sensor networks,” *Journal of Network and Systems Management*, vol. 15, no. 2, pp. 171–190, 2007.
[198] D. Raposo, A. Rodrigues, J. S. Silva, and F. Boavida, “A taxonomy of faults for wireless sensor networks,” *Journal of Network and Systems Management*, vol. 25, no. 3, pp. 591–611, 2017.
[199] T. Muhammed and R. A. Shaikh, “An analysis of fault detection strategies in wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 78, pp. 267–287, 2017.
[200] K. Ni *et al.,* “Sensor network data fault types,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 3, pp. 1–29, 2009.
[201] E. U. Warriach, M. Aiello, and K. Tei, “A machine learning approach for identifying and classifying faults in wireless sensor network,” in *2012 IEEE 15th International Conference on Computational Science and Engineering*, pp. 618–625.
[202] D. M. G. Raposo, “Monitoring Industrial Wireless Sensor Networks: A model to enhance Security and Reliability,” 00500:: Universidade de Coimbra, 2020.
[203] Mathworks, *Deep Leaning Toolbox.* [Online]. Available: https://de.mathworks.com/products/deep-learning.html (accessed: Jan. 18 2020).
[204] M. Grandini, E. Bagli, and G. Visani, “Metrics for multi-class classification: an overview,” *arXiv preprint arXiv:2008.05756*, 2020.
[205] A. Kulkarni, D. Chong, and F. A. Batarseh, “Foundations of data imbalance and solutions for a data democracy,” in *Data Democracy*: Elsevier, 2020, pp. 83–106.
[206] ThresholdTom, *A confusion matrix with predicted positives in red and predicted negatives in blue.* [Online]. Available: https://commons.wikimedia.org/wiki/File:ConfusionMatrixRedBlue.png (accessed: Sep. 7 2021).
[207] J. Makhoul, F. Kubala, R. Schwartz, R. Weischedel, and others, “Performance measures for information extraction,” in *Proceedings of DARPA broadcast news workshop*, 1999, pp. 249–252.

[208] S. M. Frank *et al.,* "Metrics and methods to assess building fault detection and diagnosis tools," National Renewable Energy Lab.(NREL), Golden, CO (United States).

[209] *Confusionmat: Confusion matrix for classification problem in MATLAB.* [Online]. Available: https://de.mathworks.com/help/stats/confusionmat.html (accessed: Sep. 11 2021).

[210] M. Viscaino, J. C. Maass, P. H. Delano, M. Torrente, C. Stott, and F. A. (. Cheein, *Confusion matrix for multi-class classification problem, with A, B, C and D as classes. PLOS ONE.* Figure. Accessed: Sep. 11 2021. [Online]. Available: https://plos.figshare.com/articles/figure/Confusion_matrix_for_multi-class_classification_problem_with_A_B_C_and_D_as_classes_/11975994/1