Simulating data center cooling systems: data-driven and physical modeling methods

Mikko Siltala

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 02.01.2020

Supervisor

Prof. Quan Zhou

Advisors

Ph.D. Jonas Gustafsson

M.Sc. Rickard Brännvall

Aalto University
School of Electrical
Engineering

Copyright © 2020 Mikko Siltala

Aalto University
School of Electrical Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Mikko Siltala

Title Simulating data center cooling systems: data-driven and physical modeling methods

Degree programme Automation and Electrical Engineering

Major Control, Robotics and Autonomous Systems **Code of major** ELEC3025

Supervisor Prof. Quan Zhou

Advisors Ph.D. Jonas Gustafsson, M.Sc. Rickard Brännvall

Date 02.01.2020 **Number of pages** 76+6 **Language** English

Abstract

As the amount and energy consumption of data centers (DC) in the world has increased, their energy efficiency has come under scrutiny. Currently the best way to improve the effectiveness of a data center is to improve the cooling system, as its energy use is essentially wasted energy. This has ignited the interest to utilize model-based control strategies for DC thermal management. However, there are no definitive solutions on how to create suitable simulation models that predict the thermal response of a data center cooling system.

In this thesis traditional physical modeling methods and new data-driven modeling methods are used to create two simulation models. The data-driven models are created using the long short-term memory (LSTM) neural network architecture using the machine learning tools TensorFlow and Keras. The physical models, which are based on the laws of physics, are created in Matlab and Simulink.

The models are validated using the Edge DC laboratory located at the RISE ICE Datacenter research facilities in Sweden. The laboratory is used to collect data from many control combinations, and roughly 90 hours of experiment data is used to train and calibrate the models, and 10 hours is used for validation and testing of the simulations. On this validation data the physical simulation model achieves a mean absolute error (MAE) of 1.45 °C on the cold aisle, and 1.35 °C on the hot aisle. The data-driven simulation model in turn achieves MAEs of 1.44 °C and 1.84 °C on the cold and hot aisles, respectively.

The accuracy of the simulations presented in this thesis are similar to what is often seen in the state-of-the-art physical models and surpass state-of-the-art neural network models. In addition, the simulation models in this thesis predict the thermal response of the entire cooling system, with further input/output temperature predictions for the other components between the server and the cooling tower. The simulations are can predict the system response for multiple hours into the future, and they can therefore be utilized for model predictive control and to find optimal DC cooling control strategies.

Keywords data center, cooling system, data-driven modeling, machine learning, Long Short-Term Memory, artificial neural network, physical modeling, thermal modeling

Aalto University
School of Electrical Engineering

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Diplomityön tiivistelmä

Tekijä Mikko Siltala

Työn nimi Palvelinkeskuksen jäähdytysjärjestelmän simulointi: dataohjatut ja matemaattiset mallinnustavat

Koulutusohjelma Automation and Electrical Engineering

Pääaine Säätötekniikka, robotiikka ja autonomiset järjestelmät **Pääaineen koodi** ELEC3025

Työn valvoja Prof. Quan Zhou

Työn ohjaajat Ph.D. Jonas Gustafsson, M.Sc. Rickard Brännvall

Päivämäärä 02.01.2020 **Sivumäärä** 76+6 **Kieli** Englanti

Tiivistelmä

Palvelinkeskuksien energiatehokkuutta on alettu tutkimaan niiden määrän sekä sähkönkulutuksen kasvun myötä. Tällä hetkellä palvelinkeskuksien energiatehokkuutta voidaan parantaa parhaiten kehittämällä niiden jäähdytysjärjestelmiä, sillä jäähdytyksen sähkönkulutus on käytännössä tuhlausta. Tämä on herättänyt kiinnostuksen mallipohjaisen säädön hyödyntämiseen palvelinkeskuksien lämmönhallinnassa. Tarkoitukseen sopivien simulaatiomallien, jotka ennustavat palvelinkeskuksen jäähdytysjärjestelmän termisen vasteen, luomiseen ei kuitenkaan ole lopullista ratkaisua.

Tässä diplomityössä käytetään perinteisiä matemaattisia mallinnusmenetelmiä, sekä uusia dataohjattuja mallinnusmenetelmiä kahden simulaatiomallin kehityksessä. Dataohjattu malli luodaan käyttäen LSTM-neuroverkkoarkkitehtuuria (engl. long short-term memory) ja TensorFlow- sekä Keras-koneoppimistyökaluja. Matemaattiset mallit, jotka perustuvat fysiikan lakeihin, luodaan käyttäen Matlab- sekä Simulink-ohjelmia.

Simulaatiomallit validoidaan käyttäen Edge-palvelinkeskuslaboratoriota, joka sijaitsee RISE ICE Datacenter -tutkimuslaitoksessa Ruotsissa. Järjestelmästä kerätään mittausdataa eri ohjaussignaalien yhdistelmillä, ja noin 90 tuntia dataa käytetään mallien opettamiseen sekä kalibroimiseen, ja noin 10 tuntia simulaatiomallien validointiin ja testaamiseen. Tällä validointijaksolla matemaattisten mallien keskipoikkeama on 1,45 °C palvelinkeskuksen kylmällä käytävällä, ja 1,35 °C kuumalla käytävällä. Neuroverkko-malli puolestaan saa keskipoikkeamat 1,44 °C ja 1,84 °C kylmällä ja kuumalla käytävällä.

Tässä diplomityössä esiteltyjen simulaatiomallien tarkkuudet vastaavat matemaattisten mallien viimeisintä tekniikan tasoa, sekä ylittävät tarkkuudessaan aiemmat neuroverkko-mallit. Tämän lisäksi diplomityön simulaatiomallit ennustavat jäähdytysjärjestelmän kaikkien komponenttien termisen vasteen palvelimista jäähdytystorniin. Järjestelmän vaste voidaan ennustaa lukuisten tuntien päähän, ja näitä simulaatiomalleja voidaan siten hyödyntää mallipohjaiseen säätöön ja löytämään optimaalisia palvelinkeskuksen jäähdytysjärjestelmän ohjausstrategioita.

Avainsanat palvelinkeskus, jäähdytysjärjestelmä, dataohjattu mallinnus, koneoppiminen, LSTM-verkko, neuroverkko, matemaattinen mallinnus, lämpömallinnus

Aalto University
School of Electrical Engineering

Aalto-universitetet, PB 11000, 00076 AALTO
www.aalto.fi
Sammandrag av diplomarbetet

Författare Mikko Siltala

Titel Simulering av kylningssystem i datacenter: datastyrda och matematiska modellerings metoder

Utbildningsprogram Automation and Electrical Engineering

Huvudämne Reglerteknik, robotik och autonomiska system **Huvudämnets kod** ELEC3025

Övervakare Prof. Quan Zhou

Handledare Ph.D. Jonas Gustafsson, M.Sc. Rickard Brännvall

Datum 02.01.2020 **Sidantal** 76+6 **Språk** Engelska

Sammandrag

Datacenters energieffektivitet har börjat granskas eftersom de ökat i energiförbrukning och i antal. Möjliga energibesparingar som kan åstadkommas i datacenter är i huvudsak kopplade till att förbättra kylningssystemen. Detta eftersom energiförbrukningen för nedkylning är slösad energi. Det här har väkt intresset att använda modellbaserade kontrollstrategier till datacentrens termiska styrning, men givna metoder att bygga sådana simuleringsmodeller som förutser den termiska responsen finns inte.

I detta examensarbete används både traditionella matematiska modelleringsmetoder samt nya datadrivna modelleringsmetoder för att utveckla två simuleringsmodeller. Datadrivna modeller skapas med hjälp av LSTM-neurala nätverksarkitekturen (eng. long short-term memory) samt maskininlärnings verktygen TensorFlow och Keras. De matematiska modellerna, som grundar sig i de fysikaliska lagarna, skapas med Matlab och Simulink.

Simuleringsmodellerna valideras med hjälp av Edge datacenter-laboratoriet vid RISE ICE Datacenters forskningsanläggningar i Sverige. Mätdata samlas in från datacentret. Av den experimentella datan används cirka 90 timmar för att träna ock kalibrera samt 10 timmar till att validera och testa simuleringsmodellerna. Med denna valideringsdata har den matematiska modellen en genomsnittlig avvikelse på 1,45 °C i den kalla gången, och 1,35 °C i den varma gången. Datadrivna modeller i sin tur har en genomsnittlig avvikelse på 1,44 °C och 1,84 °C i de kalla respektive varma gångerna.

Exaktheten av simuleringsmodellerna som presenteras i detta examensarbete motsvarar den senaste tekniska nivån i de matematiska modellerna och överträffar den i de neurala modellerna. Dessutom estimerar examensarbetets modeller den termiska responsen i hela kylningssystemet från servrar till kyltornen. Simuleringarna kan estimera responsen flera timmar framåt i tiden och simuleringsmodellerna kan därför användas för modellbaserad styrning samt för att hitta optimala styrningsstrategier för kylningssystemet.

Nyckelord datacenter, kylningssystem, datastyrd modellering, maskininlärning, LSTM-nätverk, artificiellt neuronnät, matematisk modellering, termisk modellering

Contents

Abbreviations

ANN	artificial neural network
API	application programming interface
CFD	computational fluid dynamics
COP	coefficient of performance
CPU	central processing unit
DC	data center
FC	fully connected
GPU	graphics processing unit
GRU	gated recurrent unit
HE	heat exchanger
ICE	Infrastructure and Cloud research & test Environment
ICT	information and communications technology
IT	information technology
LMTD	logarithmic mean temperature difference
LSTM	long short-term memory
MAE	mean absolute error
MPC	model predictive control
MSE	mean squared error
PUE	power usage effectiveness
RISE	Research Institutes of Sweden
RMSE	root mean squared error
RNN	recurrent neural network
SD	standard deviation
SMHI	Swedish Meteorological and Hydrological Institute

1 Introduction

In recent years, data center (DC) energy efficiency has received increasing attention due to the amount of DCs and their energy use rapidly increasing worldwide [1]. In a data center, it is common for approximately a third of the energy used to be spent on cooling [2], which is essentially wasted energy. In order to reduce the cooling energy usage, the cooling system control methods could be improved. DC cooling systems are often only static or reactive, attempting to hold a constant air temperature in the DC. Further optimization could be achieved using proactive control algorithms, such as the model predictive control (MPC) method. These methods use mathematical models to predict the response of the system, but it is challenging to create accurate models of the thermal dynamics of a DC cooling system.

The current state of the art methods in data center cooling system mathematical modeling are physical modeling, computational fluid dynamics (CFD) modeling, and recently also neural network modeling. However, these models are often lacking in some ways. The CFD and physical models are often used to simulate only parts of the cooling system, and other parts of the cooling system become simplified [3] [4]. Moreover, the neural network approach has been used to directly predict the effectiveness of DC cooling systems [5] [6], but rarely the system thermal responses [7]. Therefore, there are no common practices for creating the simulation models of an entire data center cooling system.

Therefore, the aim of this thesis is to develop simulation models for a data center cooling system, and to compare their accuracy against a real system, results from the state of the art, and against each other. This will be carried out by examining two models, a physical model based on known laws of physics, and a data-driven model trained on measurements from the Edge data center laboratory located at the RISE ICE Datacenter research facilities in Luleå, Sweden. An edge data center provides cloud computing and storage services and gets its name from literally being located at the edge of a network, close to the users. Both models will be created based on this data center. The physical models will attempt to improve the state-of-the-art models by accurately modeling the entire cooling system, and the data-driven models will be used in a new way to also estimate the internal states of the system. The hypothesis is that the performance of the state-of-the-art physical models can be improved on by using data-driven models, given that the data-driven model can learn the thermal dynamics. In addition, this thesis will be the first study to apply and compare the two modeling strategies on the same data center cooling system.

The scope of the models is limited to representing the cooling system of the Edge data center, which consists of the data center container comprising of the servers and an air-to-water heat exchanger, a coolant water storage unit, a chiller unit, and a cooling tower. The physical model will aim to only model the most important dynamics affecting the system by using textbook equations of thermodynamics as well as mass and heat transfer, and it will be created using the commonly available tools of Matlab and Simulink. The data driven model will be created using Python, TensorFlow, and Keras, as well as the Long Short-Term Memory (LSTM) neural network architecture, which is suitable for timeseries predictions.

This thesis is structured as follows. The thesis topic is introduced by stating the need for better simulation models, and the state of the art in data center cooling system modeling is introduced in Chapter 2. The background theory for the physical and data-driven modeling methods is described in the Chapters 3 and 4, respectively, and a description of the cooling system that will be modeled is given in Chapter 5. Next, the model construction methods are explained in detail for the physical model in Chapter 6 and data-driven model in Chapter 7. In Chapter 8 the training and validation data is introduced, after which the results of the validation and simulations are presented and analyzed in Chapter 9. Finally, conclusions of the simulations and this thesis are presented in Chapter 10.

2 State of the art

This chapter provides an overview into the current state of the art in data center cooling system modeling. It does this by first justifying the need for the simulations, and then proceeds to describe the modeling methods currently in use.

2.1 Data center cooling system optimization

The amount of data centers and their power consumption is on the rise. Since 2008 the power consumption of data centers has grown by 4.4 % annually, and this rate of increase has been expected to continue to 2020 [2], [8]. Data centers have been estimated to have the fastest growing energy consumption of the ICT sector, and the ICT sector power usage has been estimated to almost double by 2030 (from 2017) [2]. This growth is huge, considering that the amount of electricity used in data centers during 2011 was estimated to account for 1.1 % – 1.5 % of the world's energy consumption, which equals to 203 – 271 $\times 10^9$ kWh [1]. The energy consumption estimates have already made assumptions of increasingly energy efficient data centers, but the worst-case scenarios without efficiency improvements look more dire [8].

In addition to the high amount of CO^2 emissions the data center electricity use entails, power consumption is also the main source of a data centers running costs, and so the data center energy efficiency has become a hot topic [4], [9]. In a DC the bulk of the power is used by the IT equipment, but the second most power is consumed by the cooling system, which is needed to regulate the temperature on the IT equipment. In addition to the IT power consumption, an additional 55 % – 123 % of energy is used on cooling depending on the size of the data center (and available cooling solutions), but hyperscale data centers are estimated to only require an additional 16 % [8].

While the energy efficiency of IT equipment is better optimized, the cooling systems are often used inefficiently [4]. The efficiency is often calculated with the Power Usage Effectiveness (PUE), which is defined as the ratio of total power used to run the data center facility to the total power consumed by the IT equipment.

$$PUE = \frac{P_{total}}{P_{IT}}$$

This means that for an ideal data center where all power was used by the IT equipment the PUE score would be 1. In the United States Data Center Energy Usage Report from 2016 [8], the average PUE was found to be 1.8 – 1.9, however there are data centers with a wider range of PUE values. One major affecting factor was the size of the data center, where smaller DCs tend to have a PUE of over 2, while big hyperscale cloud data centers may have values even under 1.1. Another major factor is the location and climate the DC is in [10].

Since most data center cooling systems are suspected to be used inefficiently, searching for ways for energy efficient usage is an important task [4]. Reducing the power consumption might be achieved with hardware changes, or alternatively by searching for ways to control the existing system more efficiently [9].

Some of the common improvement attempts that have been studied are: optimizing, (often increasing) the server inlet temperature [4], [9], [11], [12], [13], [14], variable cooling control to match the IT load [4], fan speed optimization [4], [11], or maximizing temperature rise across the rack [13]. The optimization is almost as a rule done on a simulation of the system, which requires a model of the system to be prepared.

2.2 System modeling

To simulate a system its properties must first be understood. The properties can be found by experimentation, but there are cases when experimenting is not possible. For example, an experiment might tamper with a running process, or the system might not exist yet. To find the properties without experimenting on a real system a mathematical model may be created using known or assumed properties. In a mathematical model, mathematical equations are used to describe the relationships between quantities observable in the system, and a simulation may be carried out to find how a system would react in the case of some input [15]. Simulation therefore allows for experimentation without disturbing a real system, or without an existing real system. The accuracy of the model does however affect the quality of the simulation result.

The two approaches to model building are called physical modeling and system identification. Physical modeling is based on the knowledge of the laws of physics, and system identification is based on observations made on the system [15].

Using physical modeling for complex systems requires the system to be broken down into subsystems, whose behavior can be described with the laws of physics. Since the internal relations in these models are understood, they are also called white box models. System identification methods rely on observations of the system to derive relations between its inputs and outputs [15]. The internal relations are not necessarily understood, and these models are thus also called black box models. System identification is often used to supplement a physical model, and these combined models are called grey box models. The complexity of the model which can be derived from the data is related to the amount of data available, and when the model increases in complexity the system identification methods require more computing power.

2.3 Challenges specific to data center cooling system modeling

It is challenging to develop useful and detailed models of a complete data center cooling system, and thus many sources [3], [4] leave some aspects of the system unmodeled. Furthermore, the models are not available for others. This means that creating a model of a complete data center cooling system cannot be done by reproducing a previous solution. It instead requires in-depth knowledge of the system and its physics or the use of data-driven solutions. Since every system is unique, they also need unique solutions.

The challenge in modeling a cooling system comes from the complexity of the chain of components from the server to the cooling tower. Linking the components affects the performance of other components due to error propagation [11]. Furthermore, the complexity of a cooling system requires the mathematical equations to model complex interactions of laws of thermodynamics, joule heating, thermal dynamics of the building and heat transfers, and some of the properties such as the air flow speed and water pressure drops are challenging to model as a physical model [14].

2.4 Previous efforts in data center cooling system modeling

The problem of modeling a data center cooling system has been attempted in multiple ways: using computational fluid dynamics (CFD) models [7], physical models [11], [13] and data driven models [5], [7], [14], and these solutions vary from white to black box models.

In physical modeling the cooling chain is modeled as a white box model. The models are based on physical equations, such as fan laws, laws of heat balance and heat dissipation and heat exchanger effectiveness laws.

The cooling chain can be modeled block-by-block for each component from the servers to the cooling tower as multiple levels of white box models [13], but often the chiller and/or the cooling tower are not modeled [3], [4], [11]. It is possible to create the physical model using vectors and matrices of system parameters [9], [16], or by using MATLAB and Simulink [3], [4]. The models can achieve good accuracy, and their estimation errors can be as low as 1–2 °C or have errors of 1–5 % in their validation metrics.

In many papers [10], [11], [13], [16], [17], [18] the simulation temperatures are not validated against real temperatures, or the data center simulations do not provide a temperature value as an output. The simulations are instead validated using other metrics, such as the PUE, that are more relevant when predicting power usage effectiveness.

Criticism against white box solutions can be given due to them requiring extensive measurements of the system, datasheet values from component manufacturers and extensive assumptions, which mean that the model is difficult to replicate in practice. Some of these models are also static, and do not give insight to the behavior of the system dynamics [14].

Computational fluid dynamics are used to model the air flow and heat transfer in data centers using approximations of the laws of physics governing fluid dynamics. They are usually only used to study the dynamics in the server room air and can provide high resolution into the temperatures in the room. This however has high computational costs, and the simulations usually progress slower than real time [7], [17].

In comparison to physical or CFD modeling, there are not as many papers available with data driven models for data center cooling systems. There are even fewer that attempt to simulate the cooling system temperatures.

The justification for using data driven models is that they do not require extensive knowledge of the physics of the system to work [5], [14], and they can speed up the simulations [7]. The models created in this way are often global models, which characterize the entire DC room with one model using inputs such as the IT load, outside temperature, setpoints, cooling equipment usage and the cold aisle temperatures, and output the PUE, COP or DC air temperatures.

There are multiple different data-driven algorithms that can be used, and the choice is usually done based on the amount of data in the data set. Neural networks are used when there is a large amount of training data available [5], [7], and access to larger training data sets would enable users of the other techniques to also switch to using the more advanced machine learning technique [19].

Neural network models have been used to build models to estimate the PUE of the Google data centers [5]. The results of this model have good accuracy with an error of 0.4 % in the predicted PUE, and the results demonstrate that machine learning is an effective way to model DC performance and to improve energy efficiency. Artificial neural networks have also been used to predict server inlet temperatures, with average error of 0.6 °C for single time step predictions [7]. This simulation however performed poorly when attempting to predict multiple timesteps into the future, and within 5 minutes the estimate error was over 5 °C.

Others have also had success in simulating different parts of the DC with other types of machine learning algorithms, such as with the random decision forest technique [20], and linear regression [14], [19], [21].

3 Thermodynamic theory

This chapter aims to introduce the equations that are required for the creation of the physical models described in this thesis. These equations are some of the basic equations of mass and heat transfer, found in most textbooks on the subject. These equations are used to form energy balance equations for the different cooling system components in Chapter 6. The source for the equations described in this chapter is the Fundamentals of Heat and Mass Transfer, seventh edition [22].

3.1 First law of thermodynamics

The first law of thermodynamics is the law of conservation of thermodynamic energy. It describes how the change in total energy stored in a closed system is equal to the heat transferred to the system and the work done by the system to its surroundings. This equation is used multiple times to join the other equations together into energy balance equations.

$$\Delta E = Q - W \tag{1}$$

where

ΔE Change in the energy stored in the system.
Q Heat transferred to the system.
W Work done by the system.

3.2 Rate of change of a system's thermal energy

When heat is added to or removed from a system, as is the case for example for the hot and cold aisle air and the water storage tank, the rate of change of the internal energy must be estimated. When the internal temperature is assumed to be uniform, the rate of change in the internal thermal energy is expressed as

$$\Delta E = mc_p \frac{dT}{dt} \tag{2}$$

where

ΔE Change in the system's internal thermal energy.
m Mass of the system.
c_p Specific heat capacity of the system.
dT Change in the system temperature.
dt Change in time.

3.3 Thermal energy flow rate

When heat is transported in a flowing fluid, as is the case for any fluid flows in the cooling system, the thermal energy flow rate is described using equation (3).

$$q = \dot{m}c_pT \tag{3}$$

where

q Thermal energy flow rate.
$\dot{m}$ Mass flow rate.
c_p Specific heat capacity of the fluid.
T Temperature of the liquid.

3.4 Simplified steady-flow thermal energy equation

The simplified steady-flow thermal energy equation represents the net rate of heat outflow from a system. The equation (4) assumes one inflow and one outflow with equal flowrates, but the equation can be modified using the equation (3) to represent systems with more flows, like is done for the water storage tank, which has two inflows and two outflows.

$$q = \dot{m}c_p(T_{out} - T_{in}) \tag{4}$$

where

q Net heat transfer rate.
$\dot{m}$ Mass flow rate.
c_p Specific heat capacity of the fluid.
T_{out} Temperature at the outflow.
T_{in} Temperature at the inflow.

3.5 Convection heat transfer rate

The convection heat transfer rate equation is used in order to calculate the heat transfer rate between a solid and a fluid. For example, the data center air and the DC walls and IT equipment exchange heat through convection. The heat flow rate is proportional to the difference in the temperatures and the boundary surface area. The value of the heat transfer coefficient depends on the conditions present on the boundary layer, such as the surface geometry, fluid flow rate and turbulence, and other fluid thermodynamic and transport properties. This law is also known by the name of Newton's law of cooling.

$$q_{convection} = hA(T_{fluid} - T_{surface}) \tag{5}$$

where

$q_{convection}$ Convection heat transfer rate.
h Convection heat transfer coefficient.
A Effective surface area of the object that interacts with the fluid.
T_{fluid} Fluid temperature.
$T_{surface}$ Surface temperature of the object.

3.6 Lumped capacitance method

When the convected heat is assumed to change the internal temperature of the solid uniformly, the equations (2) and (5) can be used to create an energy balance equation, which is called the lumped capacitance method. The lumped capacitance method relates the rate of heat loss at the surface to the rate of change of the internal energy of the solid.

$$q = \Delta E = -hA(T_{solid} - T_{liquid}) = mc_p \frac{dT_{solid}}{dt} \tag{6}$$

where

q Convection heat loss rate.
ΔE Change in internal thermal energy.
h Convection heat transfer coefficient for the boundary between the solid and liquid.
A Surface area of the solid.
T_{solid} Temperature of the solid.
T_{liquid} Temperature of the liquid.
m Mass of the solid.
c_p Specific heat capacity of the solid.
dt Change in time.

3.7 Logarithmic mean temperature difference

The logarithmic mean temperature difference (LMTD) equation can be used to describe a heat exchanger. In the equation the total heat transfer rate from the hot fluid to the cold fluid along the heat exchanger is linked to the logarithmic mean temperature difference between the hot and cold fluids, and the overall heat transfer coefficient and effective surface area of the heat exchanger.

$$q = UA\Delta T_{LMTD} \tag{7}$$

where

$$\Delta T_{LMTD} = \frac{\Delta T_2 - \Delta T_1}{\ln \frac{\Delta T_2}{\Delta T_1}} \tag{8}$$

where

q Total heat transfer rate from the hot fluid to the cold fluid.
U Overall heat transfer coefficient of the heat exchanger.
A Surface area separating the hot and cold fluids in the heat exchanger.
ΔT_{LMTD} Logarithmic mean temperature difference between the hot and cold fluids.
ΔT_1 Local temperature difference between the hot and cold fluids at one end of the heat exchanger.
ΔT_2 Local temperature difference between the hot and cold fluids at the other end of the heat exchanger.

For a parallel flow heat exchanger the ΔT_1 and ΔT_2 are defined as

$$\begin{aligned} \Delta T_1 &= T_{h,i} - T_{c,i} \\ \Delta T_2 &= T_{h,o} - T_{c,o} \end{aligned} \tag{9}$$

And for a counter-flow heat exchanger they are as follows.

$$\begin{aligned} \Delta T_1 &= T_{h,i} - T_{c,o} \\ \Delta T_2 &= T_{h,o} - T_{c,i} \end{aligned} \tag{10}$$

where the indices h and c represent the hot and cold fluids, and the i and o the input and output of these flows, respectively.

3.8 Coefficient of performance

The cooling coefficient of performance (COP) of a system is the ratio of heat removed by a refrigerator system to the work input to the system. This equation is used to describe the efficiency of the chiller unit in Section 6.5.

$$COP_{cooling} = \frac{Q_{cooling}}{W} \tag{11}$$

where

$COP_{cooling}$	Cooling coefficient of performance.
Q_{in}	Cooling rate of the refrigerator.
W	Power input to the refrigerator.

Additionally, the heating coefficient of performance, or the ratio of the supplied heat to the work input can be defined.

$$COP_{heating} = \frac{Q_{out}}{W} = \frac{Q_{in} + W}{W} \tag{12}$$

where

$COP_{heating}$	Heating coefficient of performance.
Q_{out}	Heat rejection rate of the refrigerator.
W	Power input to the refrigerator.
Q_{in}	Cooling rate of the refrigerator.

The relationship from the cooling to the heating coefficient of performances is therefore

$$COP_{heating} = COP_{cooling} + 1 \tag{13}$$

4 Machine learning theory

Machine learning is the use of computational algorithms that learn from data as if being trained. The way a machine learning system learns is by gaining experience from performing a task and using the experience to improve its performance over time [23]. This chapter describes how the machine learning methods used in this thesis work, as well as introduces the machine learning tools used.

4.1 Choosing the machine learning algorithm

Many different machine learning algorithms are available, each suited for a different set of problems. The choice of the algorithm is therefore influenced by the requirements of the problem. This thesis attempts to estimate the internal states of a data center cooling system, and the outputs of the model should be temperatures at multiple locations in the cooling system. The input data is sequential, and the delays between the inputs and outputs have dynamic delays that depend on the operating modes of the system, and therefore the model must be able to learn distant relationships from the inputs to the outputs.

Different machine learning algorithms have different levels of complexity and performance, and since the complexity of the target system is high, simple linear regression algorithms might not work, and high-performance algorithms must be used. Artificial neural networks (ANN) work well for this type of task, and lately even users of other algorithms have switched to using neural networks [24].

Artificial neural networks are used for example by Google to manage their data centers [5], [23]. Their network estimates the efficiency of their data centers based on the IT load, setpoints, operation conditions and outside air temperatures, and this allows them to adjust the cooling equipment efficiently. The Google DC neural model takes 19 different inputs, has 5 hidden layers of 50 neurons each, and outputs the PUE, and achieves a mean absolute error of 0.4 %.

Since the artificial neural networks are powerful and are used by the industry leaders for data center management, they are going to be used in this thesis to create the data-driven model.

4.2 Artificial neural networks

Artificial neural networks are computing systems which have layers of interconnected neurons which activate based on their inputs and output the activation state to the next layer. The network approximates the complexity of a brain, and they are used to find complex relationships between input and output data [23].

The basic building block of an artificial neural network is the artificial neuron (Figure 1), which takes an input (or inputs), passes it through an equation and outputs the result. In addition to the primary inputs, there is a bias input, which always has a value of 1. The equation consists of calculating the weighted sum of the inputs and using an activation function to decide the activation (output value) of the

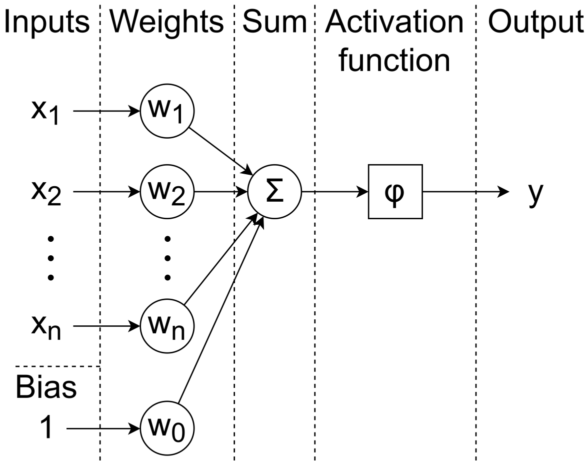

Figure 1: The artificial neuron.

neuron. The mathematical representation of a single artificial neuron is therefore

$$y = \varphi \sum_{i=0}^{n} w_i x_i \tag{14}$$

where

y = The output of the neuron.
φ = An activation function.
w = The input weights.
x = Inputs to the neuron.

There are multiple activation functions to choose from, and some of the most common ones are the step, tanh and sigmoid activation functions. The choice of activation function depends on the application of the network. For example, a binary classifier could be created by using a step activation function, which would divide the sum range into two different categories. Alternatively, a sigmoid activation function could be used to convey classification confidence, and multiple neurons could be used to reach multiple classification categories.

The input layer is the first layer of an ANN, and a single artificial neuron layer can already be used as the output layer. Using multiple neurons allows for solving for multiple outputs, and multiple layers of neurons must be utilized to solve nonlinear problems. When there is more than one layer of neurons, the middle layers (others than the input and output layer) are called the hidden layers. An ANN with a hidden layer is also called a deep neural network, although often networks with many more layers are meant when using the term. It has been proven that an ANN with one hidden layer is able to approximate any imaginable equation, and the addition of more layers does not increase the ability of the network, but they do help the networks to learn [25]. A simple multilayer neural network is presented in Figure 2.

The multilayer ANN calculates the final outputs by going through the network layer by layer calculating the neuron activations based on the values of the weights.

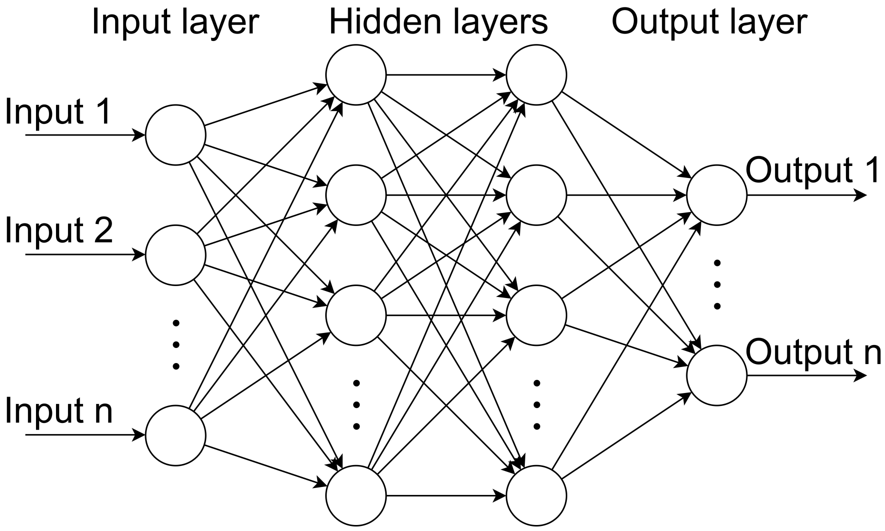

Figure 2: A simple neural network.

The values of these weights can be manually chosen only for networks approximating the most basic of equations. To find the weights for more complex networks, an optimization algorithm is used.

Gradient descent is an optimization algorithm that attempts to find an optimal solution to an equation by making gradual and iterative changes to the weight parameters [24]. It works by measuring the gradient of a loss function and following the descending gradient until it finds a minimum on the loss function. The loss function can also be called the cost function or the error function. The most common loss function used for ANNs is the mean squared error, that punishes more for larger errors. The value of the loss function is received by giving the network a training set of input-output data and computing the output errors. This step is called the feedforward step in the training process, since the data moves only forwards from layer to layer.

The next step is running the backpropagation algorithm on the network. It works backwards from the loss function and calculates the error gradient for each of the neurons in the network and alters the parameters with a step in the direction of descending gradient. The size of the steps the gradient descent algorithm takes affect the learning speed and performance of the network [24]. If the step size is too small, the algorithm will require a longer time to reach the minimum and might also get stuck in a local minimum. On the other hand, a high step size might make the algorithm jump over the minimum and never settle on a value.

4.3 Choosing the neural network architecture

The simulation models in this thesis estimate a sequence of future responses of the cooling system. Most of the system components have a delay between when the changes are made to the inputs and when the results are seen on the outputs. For a neural network to see the relationships over distant time spans from its inputs to the outputs, it must receive the relevant information from the past. When estimating the response of the next time step of some function, the network can receive the inputs either from multiple time steps or attempt to build a memory by storing the information internally between time step estimations. If long delays from inputs to outputs are expected, as is the case for the system modeled in this thesis, the amount of inputs required by the artificial neural network will grow large [24].

The neural network architectures which are specialized in predicting a time sequence where long timespans are expected are called the Recurrent Neural Networks (RNN). The RNN neurons work by not only using the network inputs to calculate the output, but by also using an input which they have sent back to themselves from the previous time step. This input is the internal state where a history of the important changes in the inputs is stored. The basic RNN has a limited memory, but the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures increase the length of memory [24].

4.4 Recurrent neural network architectures

Since the basic RNN has a limited long-term memory capability, the architectures which have a longer memory are considered. The structure of the RNN neuron is explained first, since it has similarities to the other architectures.

A recurrent neural network works mostly like a feedforward neural network, but these neurons have their output connections looped back to the inputs of the same neuron. This is called the recurrent input, and it is the hidden state of the neuron. The connections of a RNN neuron are shown in Figure 3. At each time step the recurrent input is taken from the output of the previous time step. The propagation of the output can be visually represented by unrolling the network through time, as is done for the recurrent neuron in Figure 3.

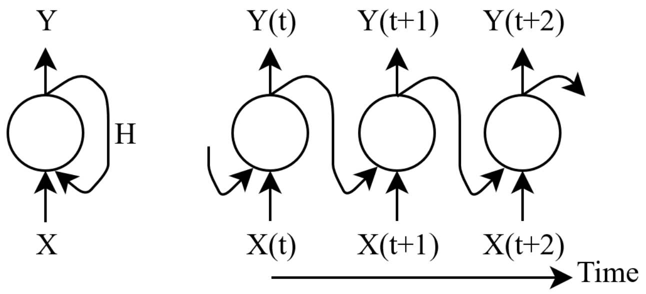

Figure 3: On the left is a single recurrent neuron, and on the right it is unrolled through time. X represents the inputs, Y the outputs, and H the hidden states.

The LSTM unit outputs an additional and separate internal state vector called the cell, which is only used as a recurrent output. This is in addition to the output vector of the unit that is also used as a recurrent output. These two recurrent inputs function as the long-term and short-term states, respectively. The cell learns when to input and output something to/from the long-term memory and what data in the long-term memory must be used in calculating the cell output.

The GRU unit is one version of a simplified LSTM unit. It does away with the short-term memory, and instead outputs always the long-term memory vector as its output. It also reduces the number of variables of the unit. GRU networks perform roughly as well as the LSTM networks [24].

There are many additional variants of the LSTM architecture which have been compared against each other in 2017 [26]. The comparison found out that the normal variant works very well amongst the tested versions, and no alternative had clear benefits in performance over the normal LSTM. Thus, the normal LSTM architecture is chosen to be used in this thesis.

4.5 The LSTM architecture

The LSTM unit uses four fully connected (FC) layers where the inputs ($x_{(t)}$ in the Figure 4) and the short-term state $h_{(t-1)}$ from the previous time step are used to calculate four activations. Three of them control gates that choose the locations in the memory vectors that will get altered, and one calculates candidate values to be stored to the memory. The LSTM unit first uses the activation $f_{(t)}$ of one of the fully connected layers to decide what the forget gate should remove from the long-term memory state $c_{(t-1)}$. Then the activation $i_{(t)}$ decides which candidate values of $g_{(t)}$ will get past the input gate to be added to the long-term memory vector. Lastly the activation $o_{(t)}$ chooses the new output $y_{(t)}$, which is also the new short-term memory vector $h_{(t)}$, using the output gate. The long- and short-term memory vectors are then looped back to the LSTM unit to be used in calculating the next time step.

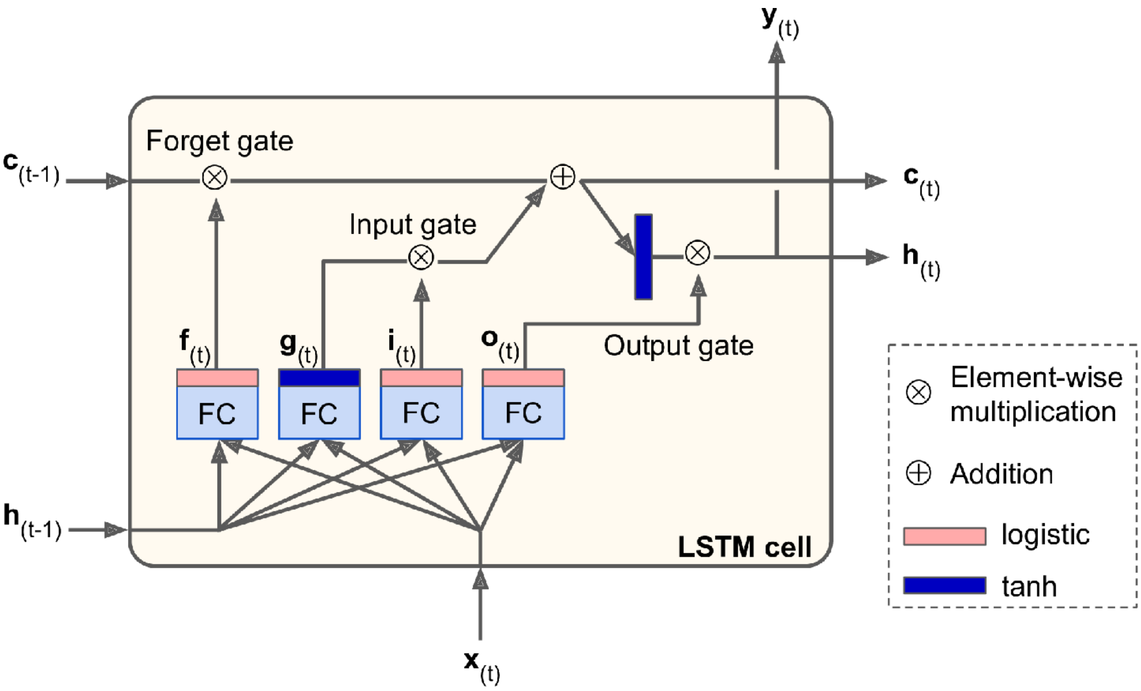

Figure 4: The LSTM unit. [24]

4.6 Machine learning tools

There are many tools available for machine learning and using them can simplify the process of machine learning greatly. The five most popular machine learning frameworks in order of decreasing popularity are TensorFlow, Keras, PyTorch, Caffe and Theano, according to a usage and popularity report from September 2018 [27].

The absolute winner in popularity is TensorFlow [27]. TensorFlow has the greatest number of users, and the most reference material available out of any tool available. It is also backed by the machine learning giant Google, and it is used and supported by many large companies [28]. Keras is an API that is used on top of other frameworks, including TensorFlow and Theano [29]. It is the easiest of the popular frameworks to begin using [27]. PyTorch is another new stand-alone framework that is quickly gaining users, while the Caffe and Theano are older frameworks, which are already on their way out of common use [27].

Due to TensorFlow being so widely used and having the best documentation, it is chosen as the tool to be used during this thesis. Keras is used on top of TensorFlow, as it promises to enable fast experimentation [29].

4.7 Cloud computing

TensorFlow can be used through the tool Jupyter Notebook [30], which allows for Python programming in a web-based environment. It can be used as the front-end interface to the machine learning programs which are run on a server. The data center where this research is conducted at has some in-house servers which already have Jupyter Notebook servers running on them. Therefore, this tool can be used to train the neural networks with more computation power and GPUs than what a typical laptop has, and this tool will speed up the training process of the neural network models.

5 System description

The Edge data center is a small data center laboratory built at the RISE ICE Datacenter research facilities. The purpose of this lab is to function as a test bed for an edge data center concept. An edge data center provides cloud computing and storage services and gets its name from literally being located at the edge of a network, close to the users. In the future edge data centers could be installed throughout cities, which would reduce the latencies for users using cloud computing and storage services in the area.

Additionally, the Edge data center laboratory tests the use of alternative energy sources in providing reliability and partial self-sufficiency for a data center. These methods include solar panels, batteries and a coolant storage tank. With these methods it is possible to provide service during power outages, and to store energy during off-peak hours to be used during more expensive peak hours.

A photo of the system is presented in Figure 5. Most of the components are visible in the photo, and the components are labeled in the caption. The cooling system consists of the components 1 to 7 in Figure 5. These components are further explained in the sections below.

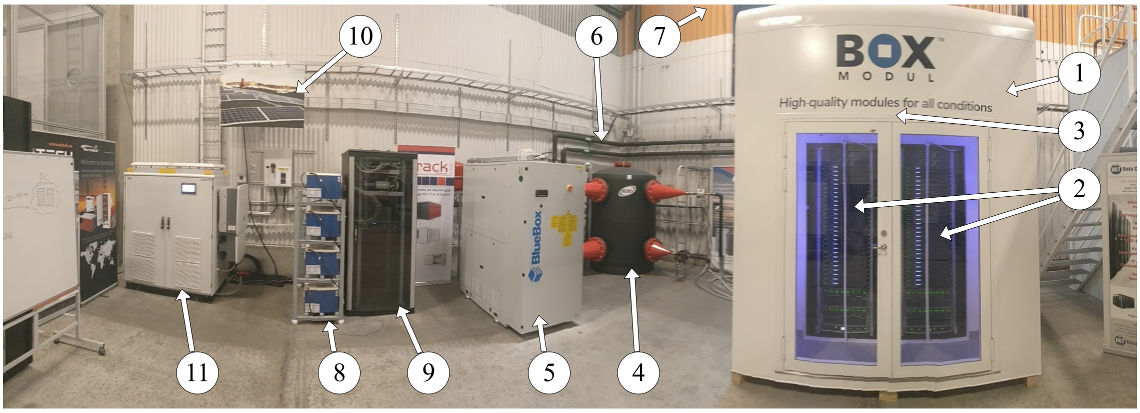

Figure 5: The Edge data center lab. (1) The module, a container for the data center. (2) Two server racks. (3) Air-to-water heat exchanger (in the module, above the racks). (4) The coolant water storage tank. (5) Chiller. (6) Coolant pipes going to the roof. (7) Cooling tower (on the roof). (8) Batteries. (9) The measurement and control system. (10) Solar panels (on the roof). (11) Microgrid inverter.

5.1 Module

The data center module is a sheet metal container, which is separated into a cold and hot aisle. The aisles are connected by a hole which fits two server racks, and an air-to-water heat exchanger above the racks.

The servers are installed on the two server racks. The racks can fit 42 units of server equipment each. During experimentation on the system, each rack had 28 Dell PowerEdge R430 servers as well as 2 HP C7600 blade enclosures, each with 16

(8x2) HP Proliant BL2x220 blades. The servers have a total idle load of about 9 kW, and a maximum load of roughly 11.5 kW. The HP servers were later replaced by the Dell servers, for a total of 38 per rack.

The power consumed by the servers is assumed to mostly turn into heat. The heat is removed by the flow of cool air originating from the cold aisle, that flows through the servers to the hot aisle. The server fans and the module heat exchanger fan contribute to this airflow.

The server exhaust air mixes in with the hot aisle air, which is in turn pulled into the heat exchanger by an air current created by the fan located in the heat exchanger. The air releases its heat to the cold water flowing through the heat exchanger. The cooled air exits the heat exchanger to the cold aisle, completing the coolant air loop inside the module.

5.2 Storage tank

The storage tank has a capacity of 2000 liters and is filled with water that is used in the cooling process. The waterflow through the heat exchanger is produced by a pump located at the base of the water storage tank. The water flows through the heat exchanger, heats up, and continues to the top of the coolant water storage tank. In addition to the input and output towards the heat exchanger, the tank also has an input and output towards the chiller, on the bottom and top of the tank respectively. The water output from the top gets cooled down by the chiller and is then returned to the base of the tank.

The inputs to the storage tank have flow diffusers to reduce the flow velocity once inside the tank, and likewise the outputs have nozzles that increase the flow velocity. This is to keep the flow velocity inside the tank low and to reduce the internal currents that would mix the water in the tank. This, and the choice of input/output locations based on the water temperatures so that cold water is input to the bottom and hot water to the top of the tank, creates a temperature gradient inside the tank where cold water is located at the bottom, and hot water at the top.

5.3 Chiller

The chiller unit has three modes of operation: the chiller mode, the free cooling mode, and a partial free cooling mode. The first two modes use different techniques to cool the incoming water, and the third uses both methods. A 3-way valve controls the flow of the glycol in these two modes to the correct heat exchangers inside the chiller unit.

5.3.1 Chiller mode

The chiller mode is based on the vapor-compression chiller technology, and consists of an evaporator heat exchanger, two compressors, a condenser heat exchanger and an expansion valve, as can be seen from Figure 6. The cooling effect is produced by some of the chiller coolant liquid vaporizing at the expansion valve which lowers the temperature of the coolant and allows it to absorb heat from the warm water in the evaporator heat exchanger. The liquid is recompressed at the compressor, which increases the temperature of the coolant, and the heat is removed in the condenser heat exchanger by the glycol returning from the cooling tower.

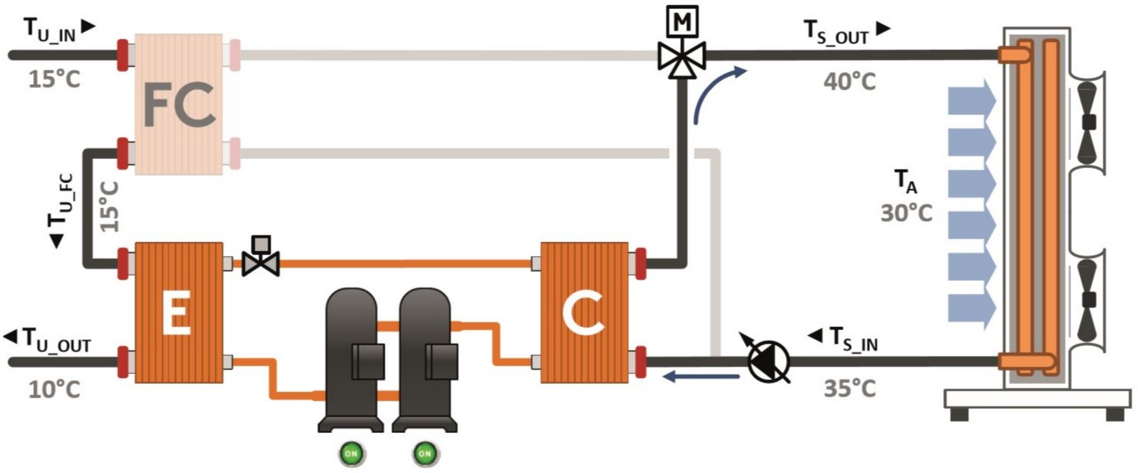

Figure 6: An example of how the chiller's vapor-compression cycle is used in chiller mode. [31]

5.3.2 Free cooling mode

The free cooling mode is used when the outside temperature is low. When this is the case, the vapor-compressor cycle is not required to increase the temperature difference between the water and glycol, and the free cooling heat exchanger is used to produce all of the cooling effect. The heat from the storage tank is exchanged to the glycol in the cooling tower pipes, as described by the Figure 7.

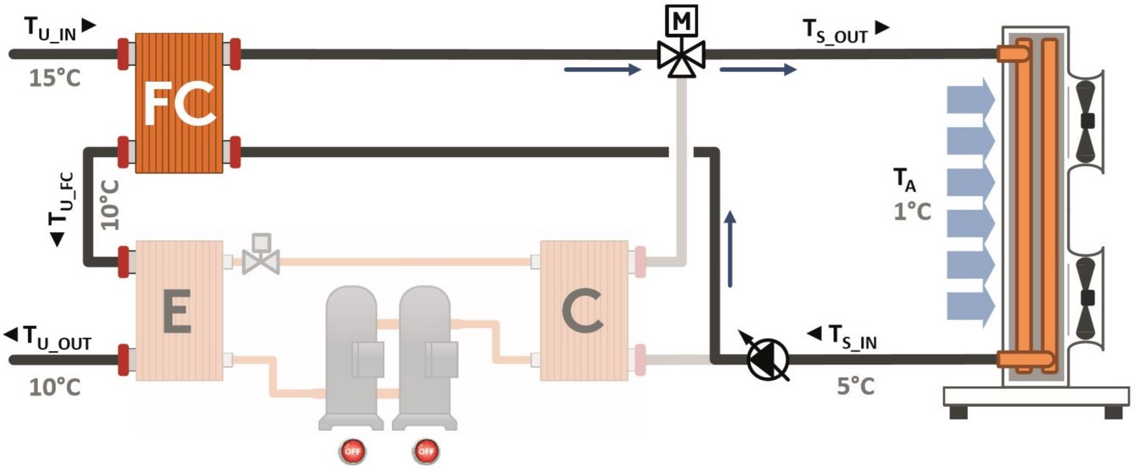

Figure 7: An example of how the chiller operates in free cooling mode. [31]

5.4 Cooling tower

The hot glycol output by the chiller is cooled at the cooling tower. The cooling tower is an air-liquid dry-cooler, where outside air is forced through the heat exchanger, and some of the heat in the glycol is removed.

5.5 Available inputs

The inputs that are used to control the Edge data center are:

- IT load setpoint.
- Chiller output setpoint, which is the temperature of the water the chiller outputs to the storage tank.
- Pump setpoint, which controls the pump between the storage tank and module heat exchanger.
- Fan setpoint, which controls the module heat exchanger fan.

Altering of the IT load is done using an interface, which gives a synthetic workload for the Dell servers to process. This interface allows for workload scheduling, where commands can be stored to be activated at specific times in the future. The HP servers instead were constantly on idle. The 100 % server load corresponds roughly to a power consumption of 11.5 kW, and the idle power draw of the servers is approximately 9 kW. To lower the load further than this requires the servers to be turned off.

The changes made to the other setpoints were done manually using another interface, which did not have the possibility for command scheduling. This limited the period in which these setpoints could be changed to the working hours.

The ranges in which the setpoints can be changed and their normal values when not running any experiments are described in Table 1.

Input	Unit	Min	Max	Normal conditions
IT load	%	0 (idle)	100	0
Chiller setpoint	°C	9	18	9
Pump setpoint	%	52	100	100
Fan setpoint	V	0	10	10

Table 1: Available inputs, their ranges and normal values.

6 Physical model

This chapter describes the creation of the physical model. The mathematical equations are derived from the equations described in Chapter 3, and screenshots of the final implementation in Simulink are provided in Appendix A. A block diagram representation of the cooling system (Figure 8) is drawn of the Edge datacenter cooling system. The physical model will represent each of the components in the block diagram as its own model, which are then joined as is done in the Figure 8 to create the complete system model. This method of dividing the model to sections is widely used in state-of-the-art solutions [3], [4], [11], [13], and many also have used Simulink to create the simulations [3], [4].

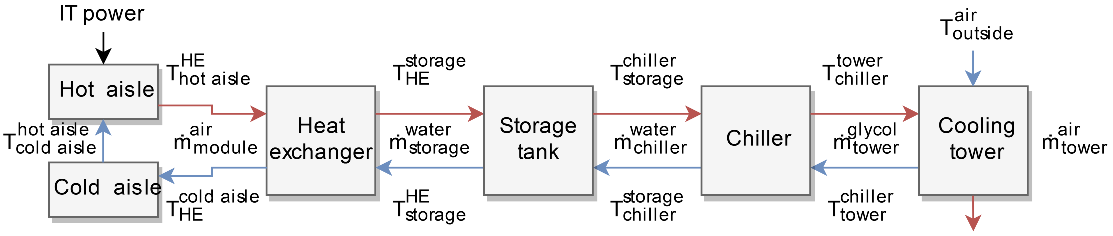

Figure 8: The Edge data center cooling system block diagram.

This method assumes ideal heat transfer, and uses nameplate values, calculated dimensions as well as natural constants when required to produce the model. These values are presented with their corresponding model, and the common ones in the Section 6.9. This chapter is divided into sections for each of the subcomponents of the system in the block diagram in Figure 8, and the mass flow rates and transport delays are also calculated in their own sections.

6.1 Cold aisle

The energy balance equation for the data center cold aisle is based on the simplified steady-flow equation (4), the convection heat transfer rate equation (5) and the internal thermal energy change rate equation (2). The net thermal energy outflow rate is equal to the convection losses through the walls to the ambient air, and the change in the internal thermal energy.

$$q_{flow} = q_{convection} + \Delta E_{internal} \tag{15}$$

$$\dot{m}^{a}_{module} c^{a}_{p} (T^{cold}_{he} - T_{cold}) = h_c A_c (T_{cold} - T_{ambient}) + \rho_a V_c c^{a}_{p} \frac{dT_{cold}}{dt} \tag{16}$$

where

$\dot{m}^{a}_{module}$ Mass flow rate of air flowing in the data center.
c^{a}_{p} Specific heat capacity of air.
T^{cold}_{he} Temperature of the air output from the heat exchanger to the cold aisle.

T_{cold} Temperature of the air on the cold aisle.
h_c Convection heat transfer coefficient between the cold aisle air and the ambient air outside of the module.
A_c Effective surface area between the cold aisle and the ambient air.
$T_{ambient}$ Temperature of the ambient air.
ρ_a Density of air.
V_c Volume of the air in the cold aisle.
t Time

The rate of change of the cold aisle air temperature is approximated using forward Euler discretization.

$$\frac{dT_{cold}}{dt} = \frac{T_{cold}^{(t+1)} - T_{cold}^{(t)}}{\Delta t} \tag{17}$$

where

Δt Length of the time step.

The values of the variables used in the equations are presented in Table 2, and how the cold aisle equations are realised in Simulink is presented in Figure A2.

Variable name	Value	Unit	Description
V_c	3.4	m^3	The volume of the cold aisle.
A_c	11.9	m^2	The surface area of the cold aisle walls.
h_c	1	$\frac{W}{m^2kg}$	The heat transfer coefficient between the cold aisle air and the ambient air. A typical value for a thin wall between a forced air flow and air in free convection is 3-10 [32].

Table 2: The values of the cold aisle variables.

6.2 Hot aisle

For the most part the hot aisle is described using the same equations as were used for the cold aisle in Section 6.1. The difference to the cold aisle is that the servers are physically located on the hot aisle, and the thermal input from the IT load has been placed to the hot aisle model. Therefore, the first law of thermodynamics (1) is used to input the thermal energy from the IT equipment to the air. Additional thermal mass to the IT equipment is also added using the lumped heat capacity equation (6). This in combination with the equations used in Section 6.1 gives the equation (19), where the IT load and net thermal energy outflow rate equal the convection losses to the environment, the heat conducted to the IT equipment and the change in the internal thermal energy of the air.

$$q_{IT} + q_{flow} = q_{convection} + q_{conduction} + \Delta E_{internal} \tag{18}$$

$$q_{IT}+\dot{m}^a_{module}c^a_p(T_{cold}-T_{hot}) = h_h A_h(T_{hot}-T_{ambient})+h_{IT}A_{IT}(T_{hot}-T_{IT})+\rho_a V_h c^a_p\frac{dT_{hot}}{dt} \tag{19}$$

where

q_{IT}	Total IT load in the data center.
T_{hot}	Temperature of the air on the hot aisle.
h_h	Convection heat transfer coefficient between the hot aisle air and the ambient air outside of the module.
A_h	Effective surface area between the hot aisle and the ambient air.
h_{IT}	Convection heat transfer coefficient between the hot aisle air and the IT equipment.
A_{IT}	Effective surface area between the hot aisle air and the IT equipment.
T_{IT}	Temperature of the IT equipment.
V_h	Volume of the air in the hot aisle.

Forward Euler discretization is used to approximate the rate of change of the hot aisle air temperature.

$$\frac{dT_{hot}}{dt} = \frac{T^{(t+1)}_{hot} - T^{(t)}_{hot}}{\Delta t} \tag{20}$$

The equation for the heat transfer between the equipment and the hot aisle air is defined using the lumped heat capacity equation (6) and Euler forward discretization.

$$h_{IT}A_{IT}(T_{hot} - T_{IT}) = m_{IT}c^{IT}_p\frac{dT_{IT}}{dt} = m_{IT}c^{IT}_p\frac{T^{(t+1)}_{IT} - T^{(t)}_{IT}}{\Delta t} \tag{21}$$

where

m_{IT}	Mass of the IT equipment.
c^{IT}_p	Specific heat capacity of the IT equipment.

As the IT load setpoint controls the synthetic load using a percentage value of the total load capacity as the input, the relationship from the IT load setpoint to the actual power used by the IT equipment is approximated using the results gained from the IT load setpoint experiment described in Section 8.1. A nonlinear load curve (Figure 9) was obtained by fitting a 2nd order polynomial to the measurements using regression analysis. This corresponds with the theory that every additional % increase of the setpoint increases the IT power use logarithmically due to the increasing CPU load involving other resources on the server initially inefficiently, and at higher loads more efficiently.

The values used for the hot aisle variables are described in the Table 3, and the realization of the hot aisle equations in Simulink is presented in Figure A3.

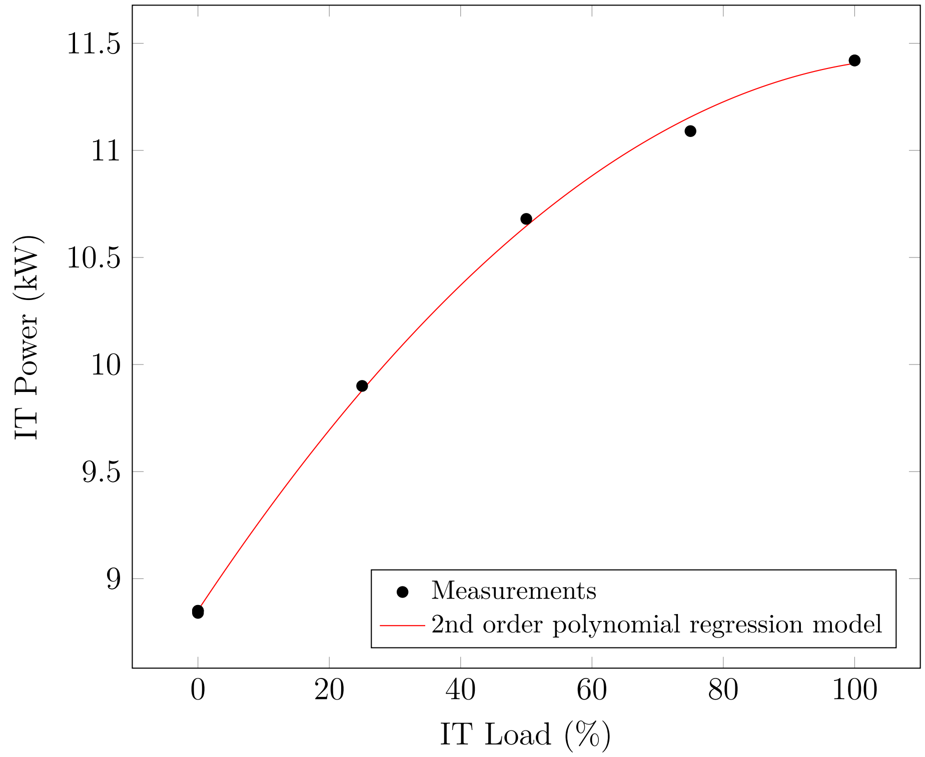

Figure 9: IT power consumption as a function of the IT load setpoint.

Variable name	Value	Unit	Description
V_h	6.7	m^3	The volume of the hot aisle.
A_h	22.1	m^2	The surface area of the hot aisle walls.
A_{IT}	24.7	m^2	The effective surface area of the IT equipment.
h_h	1	$\frac{W}{m^2kg}$	The heat transfer coefficient between the hot aisle air and the ambient air. A typical value for a thin wall between a forced air flow and air in free convection is 3-10 [32].
h_{IT}	50	$\frac{W}{m^2kg}$	The heat transfer coefficient between the hot aisle air and the servers.
m_{IT}	1671.6	kg	The mass of the IT equipment. Estimated by calculating the datasheet mass for a server times the unit capacity of the racks. [33]
c_p^{IT}	490	$\frac{J}{kgK}$	The specific heat capacity of the servers. Assumed to be that of steel.

Table 3: Values of the hot aisle variables.

6.3 Module heat exchanger

The heat exchanger completes the thermal loop inside the DC module. The heat exchanger energy balance equation from the Fundamentals of Heat and Mass Transfer assumes that there is negligible heat transfer between the heat exchanger and its surroundings, negligible potential and kinetic energy changes, there is no phase change and the specific heats are constant [22]. The equation is based on the simplified steady-flow thermal equation (4), used for both of the two mediums on the different sides of the heat exchanger, where the net rate of thermal energy removed from the air equals to the net rate of thermal energy output to the water.

$$q_{he} = q_{out}^{air} - q_{in}^{air} = q_{out}^{water} - q_{in}^{water} \tag{22}$$

$$\dot{m}_{module}^{a} c_p^a (T_{hot} - T_{he}^{cold}) = \dot{m}_{module}^{w} c_p^w (T_{he}^{storage} - T_{storage}^{he}) \tag{23}$$

where

$\dot{m}_{module}^{w}$ Mass flow rate of the water flow between the module and the storage tank.
c_p^w Specific heat capacity of water.
$T_{he}^{storage}$ Temperature of water output by the heat exchanger to the storage tank.
$T_{storage}^{he}$ Temperature of water coming from the storage tank to the heat exchanger.

Alternatively the logarithmic mean temperature difference (LMTD) equations (7) and (8) may be used to describe the heat exchanger.

$$q_{he} = U_{he} A_{he} \Delta T_{LMTD} = U_{he} A_{he} \frac{\Delta T_2 - \Delta T_1}{\ln \frac{\Delta T_2}{\Delta T_1}} \tag{24}$$

The ΔT_1 and ΔT_2 from equation (10) for a counterflow heat exchanger are used.

$$\begin{aligned} \Delta T_1 &= T_{hot} - T_{he}^{storage} \\ \Delta T_2 &= T_{he}^{cold} - T_{storage}^{he} \end{aligned} \tag{25}$$

where

q_{he} Heat transfer rate of the heat exchanger.
U_{he} Overall heat transfer coefficient of the heat exchanger.
A_{he} Total effective area of the heat exchanger.
ΔT_{LMTD} Logarithmic mean temperature difference.
ΔT_1 Temperature difference on the hot end of the heat exchanger.
ΔT_2 Temperature difference on the cold end of the heat exchanger.

The hot and cold fluids' local temperature difference differential along the heat exchanger $d(\Delta T)$ can be reorganized and used with (23) to create the equation (26).

$$\begin{aligned} d(\Delta T) &= \Delta T_2 - \Delta T_1 \\ &= (T_{he}^{cold} - T_{storage}^{he}) - (T_{hot} - T_{he}^{storage}) \\ &= -(T_{hot} - T_{he}^{cold}) + (T_{he}^{storage} - T_{storage}^{he}) \\ &= \frac{-q_{he}}{\dot{m}_{module}^{a} c_p^a} + \frac{q_{he}}{\dot{m}_{module}^{w} c_p^w} \\ &= -q_{he}\left(\frac{1}{\dot{m}_{module}^{a} c_p^a} - \frac{1}{\dot{m}_{module}^{w} c_p^w}\right) \end{aligned} \tag{26}$$

By substituting (24) into (26) and doing some reorganizing the equation (27) is received.

$$\frac{\Delta T_2 - \Delta T_1}{\Delta T_{LMTD}} = -U_{he}A_{he}(\frac{1}{\dot{m}^a_{module}c^a_p} - \frac{1}{\dot{m}^w_{module}c^w_p}) \tag{27}$$

By integrating over the length of the heat exchanger, from one end of the heat exchanger to the other, where dA_{he} is the effective surface area that heat is transferred across, the equations (28) are received.

$$\begin{aligned} \int_1^2 \frac{\Delta T_2 - \Delta T_1}{\Delta T_{LMTD}} &= -U_{he}(\frac{1}{\dot{m}^a_{module}c^a_p} - \frac{1}{\dot{m}^w_{module}c^w_p}) \int_1^2 dA_{he} \\ \ln\frac{\Delta T_2}{\Delta T_1} &= -U_{he}A_{he}(\frac{1}{\dot{m}^a_{module}c^a_p} - \frac{1}{\dot{m}^w_{module}c^w_p}) \\ \ln\frac{\Delta T_1}{\Delta T_2} &= U_{he}A_{he}(\frac{1}{\dot{m}^a_{module}c^a_p} - \frac{1}{\dot{m}^w_{module}c^w_p}) \end{aligned} \tag{28}$$

By removing the logarithm from the equation an equation is formulated from which the temperatures can be more easily solved.

$$\frac{\Delta T_1}{\Delta T_2} = \frac{T_{hot} - T^{storage}_{he}}{T^{cold}_{he} - T^{he}_{storage}} = \exp\left(U_{he}A_{he}(\frac{1}{\dot{m}^a_{module}c^a_p} - \frac{1}{\dot{m}^w_{module}c^w_p})\right) \tag{29}$$

Now the equations (23) and (29) may be used to find the two outputs T^{cold}_{he} and $T^{storage}_{he}$. The equations are combined and solved for T^{cold}_{he}.

$$\begin{aligned} &T^{cold}_{he} = \\ &\frac{T_{hot}(1 - \frac{\dot{m}^w_{module}c^w_p}{\dot{m}^a_{module}c^a_p}) + T^{he}_{storage}\frac{\dot{m}^w_{module}c^w_p}{\dot{m}^a_{module}c^a_p}(1 - \exp\left(U_{he}A_{he}(\frac{1}{\dot{m}^a_{module}c^a_p} - \frac{1}{\dot{m}^w_{module}c^w_p})\right))}{1 - \frac{\dot{m}^w_{module}c^w_p}{\dot{m}^a_{module}c^a_p}\exp\left(U_{he}A_{he}(\frac{1}{\dot{m}^a_{module}c^a_p} - \frac{1}{\dot{m}^w_{module}c^w_p})\right)} \end{aligned} \tag{30}$$

which can be, after being solved, easily used to find out the missing $T^{storage}_{he}$ when T^{cold}_{he} is substituted back into (23).

The value of the thermal conductance $k = U_{he}A_{he}$ is dependent on the temperature and flow rate of the two fluids, as well as the surface area, geometry and condition of the heat exchanger surfaces [22]. The effective surface area can be estimated based on measurements of the heat exchanger, but the other effects are more difficult to model. Therefore, the heat exchanger condition is assumed to be constant, and the effects of the flow rates will be investigated using regression analysis. The effect of the temperatures is neglected and assumed to not affect the thermal conductance.

As the other factors are static the remaining factor is the flow rate. The effect of changing each of the flow speeds is investigated by solving for the thermal conductance in equation (29) by calculating a regression representation using data from measurements of the real heat exchanger. The experiments which were used are described in Sections 8.1 through 8.7. The polynomial equation (31) which is also presented in the plot in Figure 10 is then used as the input for the thermal

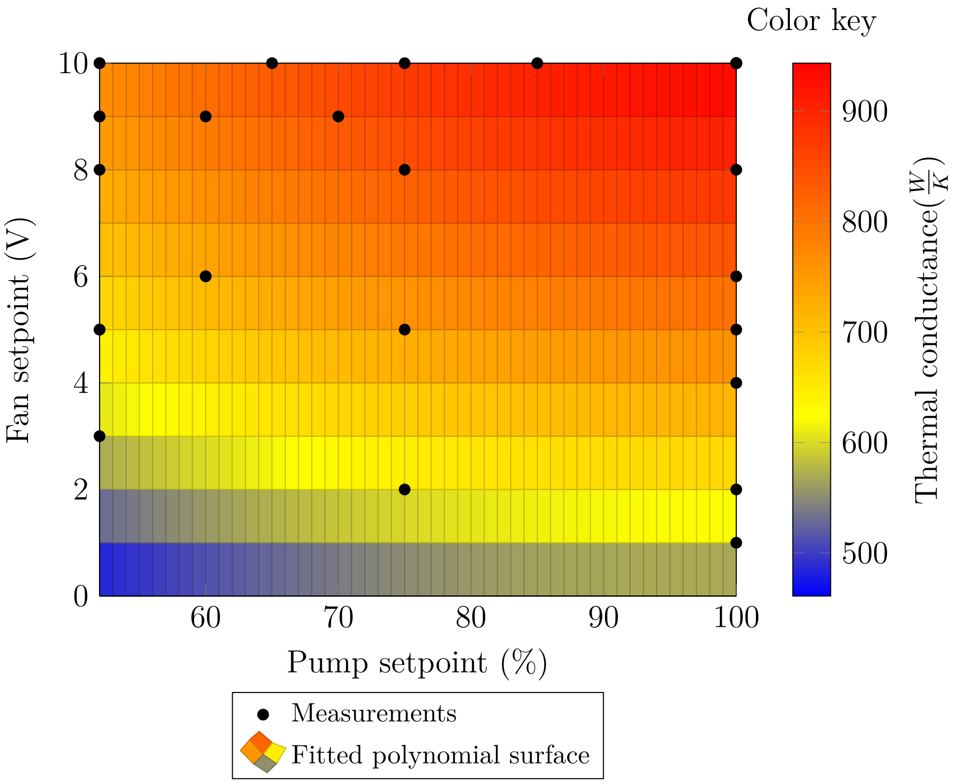

Figure 10: Thermal conductance of the data center heat exchanger as a function of the fan and pump setpoints.

conductance in the simulation. The fitted polynomial surface is of degree 2 in the pump setpoint and fan setpoint axis. The realization of the heat exchanger equations in Simulink is presented in Figure A4.

$$k = 183.8 + 7.435 s_{pump} + 33.46 s_{fan} - 0.03818 s_{pump}^2 + 0.2025 s_{pump} s_{fan} - 1.386 s_{fan}^2 \tag{31}$$

where

k	Thermal conductance
s_{pump}	Pump setpoint
s_{fan}	Fan setpoint

6.4 Storage

The inputs and outputs to the storage tank can be expressed in one energy balance equation as is done in (33), if the temperature is assumed to be uniform in the entire storage tank. The heat flow rate in the inputs to the storage tank equal to the heat flow rate in the outputs plus the change in the internal energy.

$$q_{he} + q_{chiller} = q_{out}^{he} + q_{out}^{chiller} + \Delta E_{internal} \tag{32}$$

$$\dot{m}^{w}_{module}c^{w}_{p}(T^{storage}_{he} - T_{storage}) = \dot{m}^{w}_{chiller}c^{w}_{p}(T_{storage} - T^{storage}_{chiller}) + m_{storage}c^{w}_{p}\frac{dT_{storage}}{dt} \tag{33}$$

where

$\dot{m}^{w}_{chiller}$ Mass flow rate of water between the chiller and storage tank.
$T^{storage}_{chiller}$ Temperature of water coming from the chiller to the storage tank.
$T_{storage}$ Temperature of water in the storage tank.
$m_{storage}$ Total mass of the water in the storage tank.

But since the storage tank has a temperature difference between the top and bottom, the tank model must also attempt to grasp this dynamic. Therefore, the tank is horizontally divided into two imaginary sections along the middle, and the heat flows are calculated for both. The following energy balance equations assume that the temperature is uniform in these two sections. Upon further inspection of the tank dynamics it was evident that this was not enough to fully model the outputs, and internal currents from the inflows to the outflows were found, where some of the water flows through the surrounding water without mixing with it, and in a way "bypassing" the tank.

In addition, when the flow rates to the top or bottom of the storage tank are not equal, an internal flow from one section to the other is present. The direction of the internal flow is assumed to be positive upwards in the tank. This is since the net flow rate in normal conditions is usually up. The internal flow rate is represented by the following difference of mass flow rates between the two flows interacting with the storage tank (34).

$$\dot{m}^{w}_{internal} = \dot{m}^{w}_{chiller} - \dot{m}^{w}_{module} \tag{34}$$

where

$\dot{m}^{w}_{internal}$ Internal water flow rate in the storage tank.

Depending on if the internal flow is positive or negative, the following heat flow equation must be correctly selected. The flow is assumed to be positive in all following equations for the sake of simplicity.

$$q = \begin{cases} \dot{m}^{w}_{internal}c^{w}_{p}T_{bottom} \text{ if } \dot{m}^{w}_{internal} \geq 0 \\ \dot{m}^{w}_{internal}c^{w}_{p}T_{top} \text{ if } \dot{m}^{w}_{internal} < 0 \end{cases} \tag{35}$$

where

T_{bottom} Temperature of the storage tank bottom section.
T_{top} Temperature of the storage tank top section.

With time the water temperature between the two sections will even out due to free convection. This mixing is represented with the convection equation (5).

$$q = h_{mixing}A_{tank}(T_{top} - T_{bottom}) \tag{36}$$

where

h_{mixing} Convection heat transfer coefficient across the imaginary top/bottom boundary.

A_{tank} Horizontal cross-sectional area of the storage tank that separates the two sections.

6.4.1 Storage tank bottom section

When all the equations introduced for the storage tank are combined for the bottom section of the tank, the energy balance equation (38) is received. The heat flow from the chiller and the energy gained through convection from the top section of the tank are equal to the heat outflows to the heat exchanger and to the top section of the tank as well as the change in internal energy.

$$q_{chiller}^{storage} + q_{convection} = q_{storage}^{he} + q_{bottom}^{top} + \Delta E_{internal} \tag{37}$$

$$\begin{aligned} &\dot{m}_{chiller}^{w} c_p^w T_{chiller}^{storage} + h_{mixing} A_{tank}(T_{top} - T_{bottom}) \\ &= \dot{m}_{module}^{w} c_p^w T_{bottom} + \dot{m}_{internal}^{w} c_p^w T_{bottom} + m_{bottom} c_p^w \frac{dT_{bottom}}{dt} \end{aligned} \tag{38}$$

However, this equation does not yet take into consideration the "bypassing flow". If the flow bypass in the bottom is considered as a ratio γ_{bottom} of the outflow $\dot{m}_{module}^{w}$, the updated equation (39) is received.

$$\begin{aligned} &(\dot{m}_{chiller}^{w} - \gamma_{bottom} \dot{m}_{module}^{w}) c_p^w T_{chiller}^{storage} + h_{mixing} A_{tank}(T_{top} - T_{bottom}) \\ &= \dot{m}_{module}^{w}(1 - \gamma_{bottom}) c_p^w T_{bottom} + \dot{m}_{internal}^{w} c_p^w T_{bottom} + m_{bottom} c_p^w \frac{dT_{bottom}}{dt} \end{aligned} \tag{39}$$

where the output temperature from the bottom is

$$T_{storage}^{he} = \gamma_{bottom} T_{chiller}^{storage} + (1 - \gamma_{bottom}) T_{bottom} \tag{40}$$

where

$T_{storage}^{he}$ Output water temperature towards the heat exchanger.

The rate of change of the storage tank bottom water temperature is solved using forward Euler.

$$\frac{dT_{bottom}}{dt} = \frac{T_{bottom}^{(t+1)} - T_{bottom}^{(t)}}{\Delta t} \tag{41}$$

The Simulink realization of the storage tank bottom section equations, together with the top section equations, is presented in Figure A5.

6.4.2 Storage tank top section

Similarly to the tank bottom section described in Section 6.4.1, the top section consists of the same equations, and can be considered dual to the bottom. The heat flow from the module heat exchanger and the bottom of the tank equal to the heat outflows to the chiller, energy lost to the bottom through convection and the change in internal energy.

$$q_{he}^{storage} + q_{bottom}^{top} = q_{storage}^{chiller} + q_{convection} + \Delta E_{internal} \tag{42}$$

$$\begin{aligned}&\dot{m}_{module}^{w} c_p^w T_{he}^{storage} + \dot{m}_{internal}^{w} c_p^w T_{bottom} \\ &= \dot{m}_{chiller}^{w} c_p^w T_{top} + h_{mixing} A_{tank} (T_{top} - T_{bottom}) + m_{top} c_p^w \frac{dT_{top}}{dt}\end{aligned} \tag{43}$$

The equation (43) must be modified to take into consideration the "bypassing flow". If the flow bypass in the top is considered as a ratio γ_{top} of the outflow $\dot{m}_{chiller}^{w}$, the updated equation (44) is received.

$$\begin{aligned}&(\dot{m}_{module}^{w} - \gamma_{top} \dot{m}_{chiller}^{w}) c_p^w T_{he}^{storage} + \dot{m}_{internal}^{w} c_p^w T_{bottom} \\ &= \dot{m}_{chiller}^{w} (1 - \gamma_{top}) c_p^w T_{top} + h_{mixing} A_{tank} (T_{top} - T_{bottom}) + m_{top} c_p^w \frac{dT_{top}}{dt}\end{aligned} \tag{44}$$

where the output from the top is

$$T_{storage}^{chiller} = \gamma_{top} T_{he}^{storage} + (1 - \gamma_{top}) T_{top} \tag{45}$$

where

$T_{storage}^{chiller}$ Output water temperature towards the chiller.

The rate of change of the storage tank top section water temperature is again solved using forward Euler.

$$\frac{dT_{top}^{w}}{dt} = \frac{T_{top}^{w(t+1)} - T_{top}^{w(t)}}{\Delta t} \tag{46}$$

The values used for the storage tank variables are described in the Table 4, and the realization of the storage tank equations in Simulink are presented in Figure A5.

6.5 Chiller

The chiller has two operating modes. These modes are the chiller mode and the free cooling mode, that are used during different operating conditions. Since switching between the modes diverts the flow to different heat exchangers inside the chiller, two sets of equations are required to represent the dynamics of the chiller. Furthermore, the modes are not static, but have internal controls as well, in the form of compressor on/off control in chiller mode, and 3-way valve regulation in free cooling mode. The operation logic in both modes is described in Section 6.5.1, and the thermodynamic equations for each mode follow in Sections 6.5.2 to 6.5.4.

Variable name	Value	Unit	Description
$m_{storage}$	1.996.4	kg	The mass of the water in the storage tank. Calculated from the nameplate volume of 2000 liters of water.
ρ_{top}	25	%	The share of the flow going through the top section of the storage tank that "bypasses" the storage tank.
ρ_{bottom}	25	%	The share of the flow going through the bottom section of the storage tank that "bypasses" the storage tank.
A_{tank}	1.14	m^2	The cross-sectional area of the storage tank.
h_{mixing}	50	$\frac{W}{m^2K}$	The heat transfer coefficient for the convective flow between the imaginary sections in the tank.

Table 4: Values of the storage tank variables.

6.5.1 Operation mode selection

The selection of which mode the chiller operates in is described in the manufacturers manuals [31], [34]. The same logic was programmed into the model, and the logic is presented in simple terms below.

The system can be in chiller mode or free cooling mode. The free cooling mode is activated when the input temperature to the chiller minus the outside air temperature is above the free cooling activation temperature, and is turned off when the input temperature to the chiller minus the outside air temperature is below the free cooling activation temperature plus a free cooling hysteresis value. This is visually described in the Figure 11.

In chiller mode the compressors are activated and deactivated according to the delays explained in Figure 12. The compressors are turned on after a variable delay when the output water temperature (to the storage tank) is more than the setpoint plus a neutral zone, and when the output temperature is under the setpoint, the compressors are turned off according to another variable delay. If the output water temperature is below the operating limit, the compressors are shut down immediately.

The 3-way valve is opened when the reference water temperature is higher than the setpoint minus the activation differential, and is closed proportionally when the reference temperature is between the setpoint minus the activation differential and the setpoint minus the closing differential. This is visually explained in Figure 13. The reference water temperature can be chosen between the chiller input or output water temperatures, but the output temperature is used for this unit.

The variables used for the control logic are presented in the Table 5, and the Simulink realization of the operation mode selection is presented in Figure A6.

6.5.2 Free cooling mode

In free cooling mode the exchange of heat happens at one water-to-glycol heat exchanger, as is presented in Figure 7 and explained in Section 5.3.2. The equations

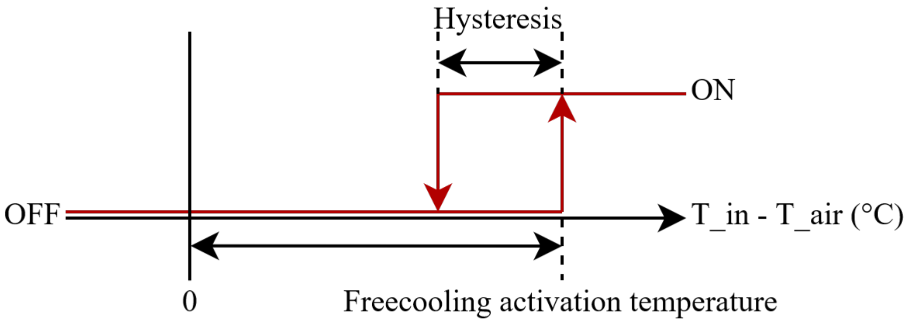

Figure 11: The activation logic of the free cooling mode.

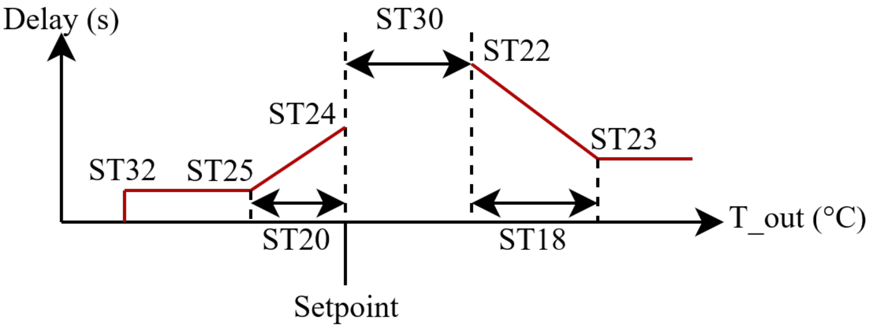

Figure 12: The compressor activation/deactivation delays as a function of the reference temperature. The variables are explained in Table 5.

in this mode are therefore similar to the module heat exchanger equations seen in Section 6.3. The net heat flow rate into the heat exchanger equals to the net heat flow rate in the outflows.

$$q_{storage}^{chiller} + q_{tower}^{chiller} = q_{chiller}^{storage} + q_{chiller}^{tower} \tag{47}$$

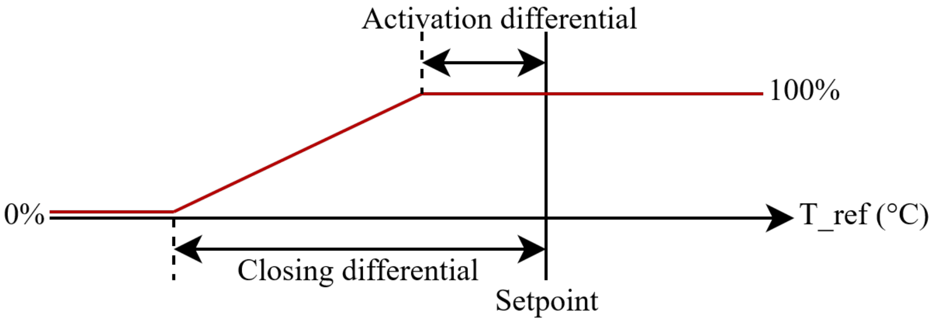

Figure 13: The control logic of the 3-way valve.

Variable	Description	Value	Unit
ST2	Minimum setpoint	9	$°C$
ST3	Maximum setpoint	18	$°C$
ST18	Activation offset	5.0	$°C$
ST20	Deactivation offset	2.0	$°C$
ST22	Max activation delay	120	s
ST23	Min activation delay	90	s
ST24	Max deactivation delay	30	s
ST25	Min deactivation delay	10	s
ST30	Chiller neutral zone	3.0	$°C$
ST32	Operating limit	3.5	$°C$
FC7	Free cooling activation temperature	5.0	$°C$
FC8	Free cooling hysteresis	2.0	$°C$
FC2	3-way valve closing differential	1.0	$°C$
FC25	3-way valve activation differential	0.5	$°C$

Table 5: The variables used in the chiller control process.

$$\dot{m}^{w}_{chiller}c^{w}_{p}(T^{chiller}_{storage} - T^{storage}_{chiller}) = \dot{m}^{g}_{tower}c^{g}_{p}(T^{tower}_{chiller} - T^{chiller}_{tower}) \tag{48}$$

where

$\dot{m}^{g}_{tower}$ Mass flow rate of glycol between the chiller and cooling tower.
c^{g}_{p} Specific heat capacity of glycol.
$T^{tower}_{chiller}$ Temperature of glycol output to the cooling tower.
$T^{chiller}_{tower}$ Temperature of glycol coming from the cooling tower.

Additionally the LMTD equations can be solved for the free cooling heat exchanger similarly to how they were solved for the data center heat exchanger in (29).

$$\frac{T^{chiller}_{storage} - T^{tower}_{chiller}}{T^{storage}_{chiller} - T^{chller}_{tower}} = \exp\left(U_{fc}A_{fc}(\frac{1}{\dot{m}^{w}_{chiller}c^{w}_{p}} - \frac{1}{\dot{m}^{g}_{tower}c^{g}_{p}})\right) \tag{49}$$

where

U_{fc} Overall heat transfer coefficient for the free cooling heat exchanger.
A_{fc} Effective area of the free cooling heat exchanger.

When (48) and (49) are combined and solved for $T^{storage}_{chiller}$, the equation (50) is received.

$$T^{storage}_{chiller} = \frac{T^{chiller}_{storage}(1 - \frac{\dot{m}^{g}_{tower}c^{g}_{p}}{\dot{m}^{w}_{chiller}c^{w}_{p}}) + T^{chiller}_{tower}\frac{\dot{m}^{g}_{tower}c^{g}_{p}}{\dot{m}^{w}_{chiller}c^{w}_{p}}(1 - \exp(U_{fc}A_{fc}(\frac{1}{\dot{m}^{w}_{chiller}c^{w}_{p}} - \frac{1}{\dot{m}^{g}_{tower}c^{g}_{p}})))}{1 - \frac{\dot{m}^{g}_{tower}c^{g}_{p}}{\dot{m}^{w}_{chiller}c^{w}_{p}}\exp(U_{fc}A_{fc}(\frac{1}{\dot{m}^{w}_{chiller}c^{w}_{p}} - \frac{1}{\dot{m}^{g}_{tower}c^{g}_{p}}))} \tag{50}$$

The other output temperature $T^{tower}_{chiller}$ can be solved from the equation (48) with a solved $T^{storage}_{chiller}$.

6.5.3 Chiller mode

In chiller mode the heat is transmitted from the storage tank to the cooling tower through a refrigerant loop as is explained in Section 5.3.1 and visually explained in Figure 6. The efficiency of a refrigeration loop is defined by the coefficient of performance (COP), for which an equation is presented in (11). When this equation is combined with the thermal energy flow equation (4) and the lumped heat capacity equation (6) for the evaporator heat exchanger thermal mass, we receive the following equation. The heat flow rate from the storage to the chiller minus the cooling rate of the chiller equals the heat outflow rate to the storage tank and the energy lost through convection to the system.

$$q_{storage}^{chiller} - q_{COP} = q_{chiller}^{storage} + q_{convection} \tag{51}$$

$$\dot{m}_{chiller}^{w} c_p^w T_{storage}^{chiller} - COP_C W_{net} = \dot{m}_{chiller}^{w} c_p^w T_{chiller}^{storage} + h_E A_E \left(\frac{T_{storage}^{chiller} + T_{chiller}^{storage}}{2} - T_E\right) \tag{52}$$

where

COP_C Coefficient of performance of the chiller unit cooling.
W_{net} Net work input to the chiller unit.
h_E Heat transfer coefficient from the water to the evaporator heat exchanger.
A_E Effective surface area between the evaporator heat exchanger and the water.
T_E Temperature of the evaporator heat exchanger.

Which can be solved to find one of the outputs of the chiller.

$$T_{chiller}^{storage} = T_{storage}^{chiller} - \frac{COP_C W_{net} + h_E A_E \left(\frac{T_{storage}^{chiller} + T_{chiller}^{storage}}{2} - T_E\right)}{\dot{m}_{chiller}^{w} c_p^w} \tag{53}$$

The evaporator heat exchanger thermal mass dynamics are calculated using the lumped heat capacity equation (6).

$$h_E A_E \left(\frac{T_{storage}^{chiller} + T_{chiller}^{storage}}{2} - T_E\right) = m_E c_p^s \frac{dT_E}{dt} \tag{54}$$

where

m_E Mass of the evaporator heat exchanger.
c_p^s Specific heat capacity of steel.

The other output on the other side of the chiller can then be solved from the heating COP equation (13), and the lumped heat capacity equation for the condenser heat exchanger thermal mass. The heat flow rate in the inflow from the cooling tower and the heat removed from the storage water line and the work input to the system equals to the heat outflow rate to the cooling tower and the thermal energy lost to the condenser heat exchanger through convection.

$$q_{tower}^{chiller} + q_{COP} + W = q_{chiller}^{tower} + q_{convection} \tag{55}$$

$$\dot{m}_{tower}^{g} c_p^g T_{tower}^{chiller} + (COP_C + 1)W_{net} = \dot{m}_{tower}^{g} c_p^g T_{chiller}^{tower} + h_C A_C \left(\frac{T_{tower}^{chiller} + T_{chiller}^{tower}}{2} - T_C\right) \tag{56}$$

$$T_{chiller}^{tower} = T_{tower}^{chiller} + \frac{(COP_C + 1)W_{net} - h_C A_C \left(\frac{T_{tower}^{chiller} + T_{chiller}^{tower}}{2} - T_C\right)}{\dot{m}_{tower}^{g} c_p^g} \tag{57}$$

where

h_C Condenser heat exchanger heat transfer coefficient.
A_C Effective area of the condenser heat exchanger.
T_C Temperature of the condenser heat exchanger.

The condensor heat exchanger thermal mass dynamics are calculated using the lumped heat capacity equation (6).

$$h_C A_C \left(\frac{T_{tower}^{chiller} + T_{chiller}^{tower}}{2} - T_C\right) = m_C c_p^s \frac{dT_C}{dt} \tag{58}$$

where

m_C Mass of the evaporator heat exchanger.

When the chiller compressors are off, the chiller coolant and the cooling tower glycol are not circulated, and any cooling effect happening to the water from the storage tank happens by the evaporator heat exchanger acting as a heat sink with thermal mass. Therefore, the energy balance equation for the storage water loop is almost the same as when the compressors are on. The modified equation is presented in (59), from which the $T_{chiller}^{storage}$ can be solved from.

$$\dot{m}_{chiller}^{w} c_p^w T_{storage}^{chiller} = \dot{m}_{tower}^{g} c_p^g T_{chiller}^{storage} + h_E A_E \left(\frac{T_{storage}^{chiller} + T_{chiller}^{storage}}{2} - T_E\right) \tag{59}$$

When the cooling tower glycol does not flow, there are no major dynamics affecting the temperature of the glycol outflow, as the piping is well insulated. As this effect is minimal, the glycol temperature is considered to not change when there is no flow.

The values of the variables used in the chiller equations are presented in Table 6, and the realization of the chiller thermodynamics into Simulink are presented in Figures A7 and A8, where each figure represents the equations used for solving one of the outputs.

Variable name	Value	Unit	Description
$COP_{cooling}$	4.2		The coefficient of performance of the chiller operating in chiller mode.
c_p^s	490	$\frac{J}{kgK}$	The specific heat capacity of the chiller heat exchangers, which are made of steel.
U_{fc}	3000	$\frac{W}{m^2K}$	The overall heat transfer coefficient of the free cooling heat exchanger. A typical value for a water-water heat exchanger is 900-2500 [32].
A_{fc}	2.4	m^2	The effective surface area of the free cooling heat exchanger.
h_E	2500	$\frac{W}{m^2K}$	The heat transfer coefficient of the evaporator heat exchanger. A typical value for a water-water heat exchanger is 900-2500 [32].
A_E	3.2	m^2	The effective surface area of the evaporator heat exchanger.
m_E	32	kg	The mass of the evaporator heat exchanger.
h_C	900	$\frac{W}{m^2K}$	The heat transfer coefficient of the condenser heat exchanger. A typical value for a water-water heat exchanger is 900-2500 [32].
A_C	4	m^2	The effective surface area of the condenser heat exchanger.
m_C	40	kg	The mass of the condenser heat exchanger.

Table 6: Values of the chiller variables.

6.5.4 Partial free cooling mode

The partial free cooling mode is chosen to not be modeled, and there are two reasons that contribute to this decision. First, the system can be very rarely seen to operate in this mode. It occurs only during the transition between the chiller and free cooling modes, and the proportional amount of time spent in this mode is small compared to the other two modes. Secondly, the partial mode would require the use of both chiller and free cooling mode equations in series to provide the cooling effect, and this would be impossible to validate as the temperature in the different branches in the chiller are not measured. Not modeling this partial mode also reduces the complexity of the chiller model.

6.6 Cooling tower

The last component of the system is the cooling tower, where the heat is expelled from the system. This component is again a heat exchanger, and the dynamics are solved using the same equations as the previous heat exchangers, where the heat radiated to the outside air equal to the heat removed from the glycol.

$$q_{chiller}^{tower} + q_{outside}^{air} = q_{tower}^{chiller} + q_{exhaust}^{air} \tag{60}$$

$$\dot{m}^{g}_{tower}c^{g}_{p}(T^{tower}_{chiller} - T^{chiller}_{tower}) = \dot{m}^{a}_{tower}c^{a}_{p}(T_{exhaust} - T_{outside}) \tag{61}$$

where

$\dot{m}^{a}_{tower}$ Mass flow rate of air through the cooling tower.
$T_{exhaust}$ Temperature of the exhaust air from the cooling tower.
$T_{outside}$ Temperature of the outside air.

The LMTD equation is also used here to describe the temperature differences of the different flows. The flow direction in this heat exchanger is assumed to be counterflow.

$$\frac{T^{tower}_{chiller} - T_{exhaust}}{T^{chiller}_{tower} - T_{outside}} = \exp\left(U_t A_t\left(\frac{1}{\dot{m}^{g}_{tower}c^{g}_{p}} - \frac{1}{\dot{m}^{a}_{tower}c^{a}_{p}}\right)\right) \tag{62}$$

where

U_t Overall heat transfer coefficient of the cooling tower.
A_t Total cooling tower radiator area.

The equations (61) (62) can be combined similarly to what was done in (30) to find out $T^{chiller}_{tower}$.

$$T^{chiller}_{tower} = \frac{T^{tower}_{chiller}(1 - \frac{\dot{m}^{a}_{tower}c^{a}_{p}}{\dot{m}^{g}_{tower}c^{g}_{p}}) + T_{outside}\frac{\dot{m}^{a}_{tower}c^{a}_{p}}{\dot{m}^{g}_{tower}c^{g}_{p}}(1 - \exp\left(U_t A_t(\frac{1}{\dot{m}^{g}_{tower}c^{g}_{p}} - \frac{1}{\dot{m}^{a}_{tower}c^{a}_{p}})\right))}{1 - \frac{\dot{m}^{a}_{tower}c^{a}_{p}}{\dot{m}^{g}_{tower}c^{g}_{p}}\exp\left(U_t A_t(\frac{1}{\dot{m}^{g}_{tower}c^{g}_{p}} - \frac{1}{\dot{m}^{a}_{tower}c^{a}_{p}})\right)} \tag{63}$$

The $T_{exhaust}$ is not necessary to be solved, as it does not affect the system by looping back as an input.

The values of the variables used in the cooling tower equations are presented in Table 7, and the Simulink realization of the cooling tower dynamics is presented in Figure A9.

Variable name	Value	Unit	Description
h_t	10	$\frac{W}{m^2K}$	The heat transfer coefficient between the cooling tower heat exchanger and the outside air. A typical value for a water-air heat exchanger is 10-50 [32].
A_t	800	m^2	An estimated effective surface area of the cooling tower heat exchanger.

Table 7: Values of the cooling tower variables.

6.7 Mass flow rates

There are five mass flow rates that are required to be solved for the system to function. The locations of these mass flows in the system are visually presented in the Figure 8. The values of the mass flows which have been observed to be constant are presented in Table 8. The equations to calculate these mass flow rates are presented in the following sections, and the Simulink realization of them is presented in Figure A10.

Variable name	Value	Unit	Description
$\dot{m}^{chiller}_{storage}$	0.64	$\frac{kg}{s}$	The mass flow rate of water flowing between the storage tank and the chiller.
$\dot{m}^{tower}_{chiller}$	0.89	$\frac{kg}{s}$	The mass flow rate of glycol flowing between the chiller and cooling tower that is observed during the chiller mode.
$\dot{m}^{tower}_{chiller,fc}$	1.8	$\frac{kg}{s}$	The mass flow rate of glycol flowing between the chiller and cooling tower during free cooling mode.
$\dot{m}^{air}_{tower}$	4.0	$\frac{kg}{s}$	The mass flow rate of air flowing through the cooling tower. Method of estimation explained in Section 6.7.5.

Table 8: Averaged values of the constant mass flow rates.

6.7.1 Data center air mass flow rate

The fan setpoint experiment (described in Section 8.4) was used to calculate the relation between the fan setpoint and the air mass flow in the data center. The heat exchanger energy balance equation (23) was used to calculate the air mass flow rate by solving for the flow rate and inserting all the other variables from measurements at each setpoint. The resulting linear regression fit is presented in Figure 14. Each data point is the average of one hour of measurements. The reason why the rate estimate is nonzero when the fan is turned off, is due to the servers having their own fans which provide a base level of air flow.

6.7.2 Storage–heat exchanger water mass flow rate

The pump setpoint experiment described in Section 8.3 was used to calculate the relationship from the pump setpoint to the water mass flow. The resulting polynomial regression representation of the flow rate dependency is presented in Figure 15.

6.7.3 Chiller–storage tank water mass flow rate

The mass flow of the water between the chiller and storage tank is constant. The pump is controlled by the chiller, and there is no operation mode where the mass flow would be adjusted. This means that a mean mass flow rate received from

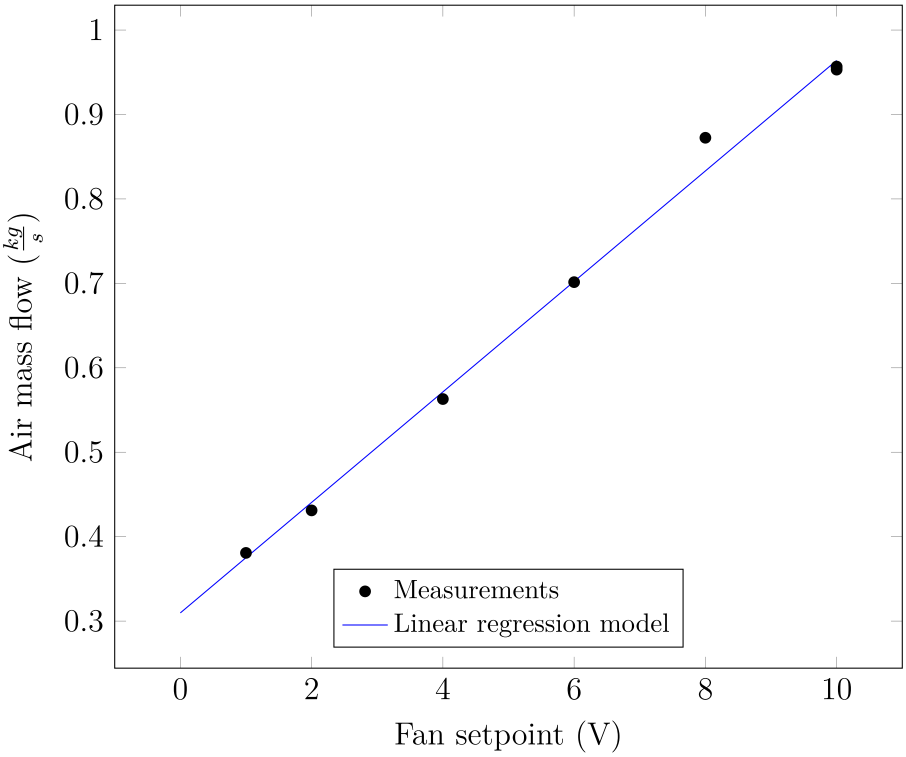

Figure 14: The data center air mass flow rate as a function of the fan setpoint.

measurements from a flow meter located on the waterline is used as the input for the simulation.

6.7.4 Cooling tower–chiller glycol mass flow rate

The mass flow rates in the two chiller modes are different. In the chiller mode the flow is constant when the compressors and source side pump are on, and zero when the compressors are off. In free cooling mode the flowrate is increased, and the direction or proportion of the flow to the different chiller heat exchangers is adjusted using the 3-way valve. The rate is thus estimated as a function of the chiller control signals.

These dynamics have been observed to be true during most of the collected data, but some variations to this behaviour have also been found. For example in the data used for model validation, while the system is in free cooling mode, the flow rate is measured to decrease to the rate it usually has during chiller mode without changing the mode. The reason why this happens is unknown as the chiller manuals do not contain much on pump control.

6.7.5 Cooling tower air mass flow rate

The air mass flow through the cooling tower is not measured by any sensor, and must be estimated instead. The fan located at the cooling tower is controlled by the chiller, and based on the power usage measurements, which are relatively constant,

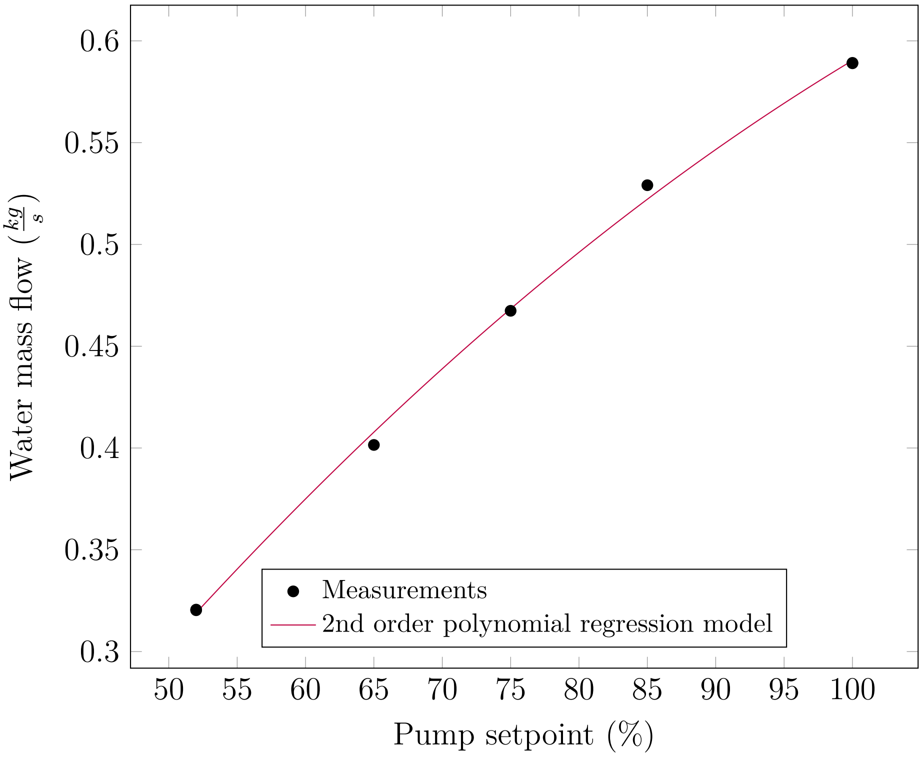

Figure 15: Water mass flow rate between the storage tank and module heat exchanger as a function of the pump setpoint.

the fan speed should be constant. Without enough measurements to be able to solve the air flow rate as was done for the module in Section 6.7.1, the mass flow rate is assumed to be linearly dependent on the fan power use like in the data center heat exchanger fan, and by using the known values for that fan, the tower air mass flow is solved from (64).

$$\dot{m}^a_{tower} = \frac{W^{fan}_{tower}}{W^{fan}_{module}} \dot{m}^a_{module} \tag{64}$$

where

W^{fan}_{tower} Power used by the cooling tower fan.
W^{fan}_{module} Power used by the module heat exchanger fan.

6.8 Transport delays

Estimating the transport delays in the system can be calculated for the sections where fluid is pumped through pipes by comparing the volume of the pipe and the flow rate. The equation used for this is presented below (65). The transport delays in the airflow in the module are estimated similarly by comparing the flow rate to the volume it must flow through.

$$Delay = \frac{V\rho}{\dot{m}} \tag{65}$$

where

Delay	Transport delay in seconds.
V	Volume of the pipe/aisle.
ρ	Density of the fluid in the pipe/aisle.
$\dot{m}$	Mass flow rate of the fluid.

During simulation the transport delay is dependent on the dynamic mass flow rates, and thus the delays are calculated dynamically by dividing the $V\rho$ with the mass flow rate. The values of $(D = V\rho)$ that are used in the delay calculations are presented in the Table 9.

Variable	Value	Transport delay location
D_1	1.6	Cold aisle – hot aisle
D_2	8.1	Hot aisle – heat exchanger
D_3	4.0	Heat exchanger – cold aisle
D_4	7.2	Heat exchanger – storage tank
D_5	4.7	Storage tank – heat exchanger
D_6	6.4	Storage tank – chiller
D_7	4.8	Chiller – storage tank
D_8	81.9	Chiller – cooling tower
D_9	83.2	Cooling tower – chiller

Table 9: Values of $D = V\rho$ used to calculate the transport delays in the system.

6.9 Common variables

The common variables that are used by all components in the simulation are the physical constants, which are presented in Table 10. The variables were assumed to be at 20 °C and 1 atm where appropriate, and the same everywhere in the simulation.

Variable name	Value	Unit	Description
c_p^{air}	1006	$\frac{J}{kgK}$	The specific heat capacity of air at 20 °C and 1 bar [32].
c_p^{water}	4184.4	$\frac{J}{kgK}$	The specific heat capacity of water at 20 °C [32].
c_p^{steel}	490	$\frac{J}{kgK}$	The specific heat capacity of steel at 20 °C [32].
c_p^{glycol}	3132	$\frac{J}{kgK}$	The specific heat capacity of glycol at 4.4 °C [32].
ρ_{air}	1.204	$\frac{kg}{m^3}$	The density of air at 20 °C and 1 atm [32].
ρ_{water}	998.21	$\frac{kg}{m^3}$	The density of water at 20 °C and 1 atm [32].
ρ_{steel}	8000	$\frac{kg}{m^3}$	The density of steel [32].
ρ_{glycol}	1095	$\frac{kg}{m^3}$	The density of 60 % glycol solution at 20 °C [32].

Table 10: The physical constants used by the simulation.

7 Data-driven model

To create a data-driven model might only require a few lines of code depending on the chosen tools, but the biggest task in creating a good model is understanding the problem and which tools will work best for the task. The LSTM neural networks were chosen to be used for constructing the data-driven models. The decision is justified and more theory on why this approach will work is described in Chapter 4. When a suitable tool is chosen it will still require the user to input the data in a correct format and to configure the algorithm to produce the results that are wished for. Therefore, this chapter will describe and justify the methods used in creating the data-driven models described in this thesis. The data-driven models are validated in Section 9.3, and the simulation results are described in Section 9.4.

7.1 Model requirements

The objective of the data-driven models is to predict sequences of the same temperatures that are also predicted by the physical model. A small simplification is made where the cold aisle temperature is not estimated, as it can be approximated by the heat exchanger output air temperature. The data-driven models should attempt to model the individual components of the cooling system, as well as the entire system. The system is divided into five sections, and there are 8 internal states which connect the different sections. The sections, the output temperatures and their connections are presented in Figure 16.

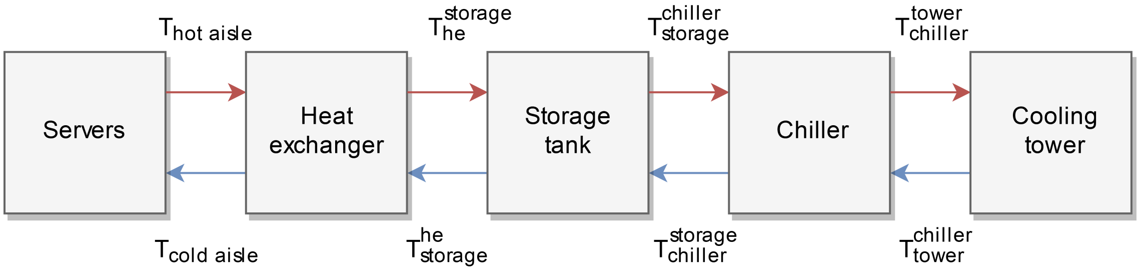
$T_{\text{hot aisle}}$
$T_{\text{he}}^{\text{storage}}$
$T_{\text{storage}}^{\text{chiller}}$
$T_{\text{chiller}}^{\text{tower}}$
Servers
Heat exchanger
Storage tank
Chiller
Cooling tower
$T_{\text{cold aisle}}$
$T_{\text{storage}}^{\text{he}}$
$T_{\text{chiller}}^{\text{storage}}$
$T_{\text{tower}}^{\text{chiller}}$

Figure 16: The division of the cooling system into data-driven models and their outputs.

The models must also be initializable. This means that the model architecture must allow the initial values of the internal states to be given before the simulation start. The internal state variables of the LSTM neuron are stored in the state vector, which is unintelligible to an observer, so the values cannot be directly written into it [24]. The other method of initialization is to give the model an initial input from measured data. In an LSTM network the output is dependent on the internal state to be accurate, and just one time step will not be enough to fully initialize the models. Therefore the initialization is provided by running the simulation with an initialization sequence before predictions can be made.

With these requirements in place, the structure of the model components should become clear. The outputs of one model are therefore used as inputs for other models, creating a dependency between the models. The way the models are connected is further explained in detail in Figure 17. For simplicity there are just two models shown, marked A and B. The models have two types of inputs, marked with X and Y. The X inputs contain those inputs which are not dependent on the other models, such as the system setpoints, and Y contains those inputs which are the output of some other model. Training of the models is therefore possible separately, and the models must also be initialized separately, but when the simulation period begins the model components can be connected.

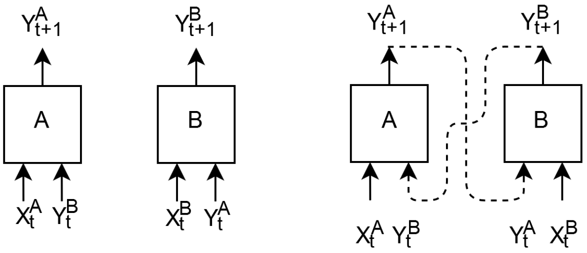

Figure 17: The data driven model component input/output structure showing the connections between the models. The models are trained and initialized without connecting them with each other (on the left), but the components are connected during simulation (on the right).

7.2 Input data

The starting point to creating a data-driven model is the data [24]. In this case the data measurement devices were already in place in the system, and the first task was to pick suitable measurements that could allow the model to gain insightful information into the inner states of the system. Furthermore, as the goal was to model the system components as well as the entire system as a whole, data from multiple locations in the system was gathered, but not all of the data was used as inputs for any given model.

The first data that was collected were the output values that are estimated by each model. These measurements were taken from the system so that they most closely would represent the output temperatures from each of the components, and also would correspond to the output temperatures modeled by the physical model described in Figure 8.

The other part of the required data are the inputs, which were initially chosen to be the same measurements that the physical models require as their inputs, such as the setpoints to control the fans, pumps, chiller, and the IT-load, and the input temperatures and mass flows. In addition to these values other suitable measurements were added, such as the differential pressure between the cold and hot aisles, that might provide additional information for the models to ease the training process. At the beginning of the modeling process many of the inputs that were considered got removed, as comparisons of different models revealed some measurements to not enhance the model accuracy above a marginal amount.

In order to avoid the models from becoming persistence models (a model which has the same values as an input and as an output, and assumes no change to take place between time steps, and propagates the input to the output without changes), the measurements the model attempted to estimate were avoided from being used as its inputs. This could not be done for the chiller model, as the only models which could correctly learn the rapid changes of the compressor going on/off used the output temperatures from the previous timestep as their inputs. The final input and output data are described in the Table 11.

Model	Inputs	Outputs
Server	IT setpoint, Fan setpoint, $T_{ambient}$, T_{HE}^{server}	T_{server}^{HE}
Heat Exchanger	IT setpoint, Fan setpoint, Pump setpoint, $T_{ambient}$, T_{server}^{HE}, $T_{storage}^{HE}$	T_{HE}^{server}, $T_{HE}^{storage}$
Storage tank	Pump setpoint, $T_{ambient}$, $T_{HE}^{storage}$, $T_{chiller}^{storage}$	$T_{storage}^{HE}$, $T_{storage}^{chiller}$
Chiller	Chiller setpoint, $T_{ambient}$, $T_{outside}$, $T_{storage}^{chiller}$, $T_{tower}^{chiller}$, $T_{chiller}^{storage}$, $T_{chiller}^{tower}$	$T_{chiller}^{storage}$, $T_{chiller}^{tower}$
Cooling tower	Chiller setpoint, $T_{ambient}$, $T_{outside}$, $T_{chiller}^{tower}$, $T_{chiller}^{storage}$, $T_{storage}^{chiller}$	$T_{tower}^{chiller}$

Table 11: The inputs and outputs of the data-driven models.

The training data was chosen so that it contains all but one of the experiments that were conducted with the system. The remaining experiment was used as the validation data. All experiments are described in the Section 8. Additional data of the system idling, where no experimentation on the system was done, was not used as training or validation data as it did not provide new information of the system dynamics, and would prioritize training for the idle state instead of generalizing to multiple states.

7.3 Data-driven model architecture

After the available input data was chosen a suitable architecture could be chosen. As the problem at hand is of time sequence prediction, architectures specifically designed for this task were considered. The main ones taken into consideration were the Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures. As is also described in more details in Chapter 4, the LSTM architecture is considered to be the best option for this kind of problem [24], and was chosen to be used.

There is one more choice that must be made with the use of the LSTM network architecture, and that is the choice between using a stateless or stateful architecture. The difference in these methods is in the way the input data is presented to the model, and that affects the way it will train. In the stateless method the input data is presented from multiple time steps at the same time, which gives more inputs for the network to use, but the internal state of the neurons gets reset between each batch of input data [24]. The stateful model on the other hand receives the input data for the current time step only, but the internal states do not need to be reset until the user wishes to [24].

Since the system goes through different operating modes, and there are dynamic delays, the stateless method would drop this information when the internal states would be reset. The stateful method, if it is able to learn these dynamics, will be able to store the data in the internal states which do not need to ever be reset. Therefore, the stateful LSTM network architecture will be used.

7.4 Data preprocessing

A dataset used for training the models is created from the raw data collected from the system. The raw data is processed uniformly, which then gets split into separate training and testing datasets to be used directly by the training algorithm.

The preprocessing begins by resampling all of the measurements uniformly and regularly, since the measurements are not necessarily taken from the same points in time. The resampling period is 30 seconds, which is the longest sampling period of some of the measurements. Sometimes there is no measurement for multiple minutes, for example due to the measurement network being overloaded. If there is no measurement during a sampling period something must be done about the empty slot in the dataset.

In order to keep the time sequence intact, the measurement cannot be removed as then the entire time step would be removed. Missing values can be replaced for example with some constant value like zero, the last known value or with a linear estimation between the last known value and the next known measurement in the series [24]. The linearization method was chosen to be used, as it approximates the change without adding sudden changes to the values (compared to filling with the last known values) or input false values (compared filling with some constant value).

The data is then normalized to between 0 and 1. This is done because the data of different inputs have different ranges that vary by a lot. For example, the outside temperature can be in the range of -30 °C to +30 °C, while the fan setpoint ranges from 0 V to 10 V. This difference does not mean that that the variable with the greater variation should affect the results more. In order to give each of the inputs a similar weight, all of them should be normalized to the same range [24].

As the model outputs predictions of the next time steps values, columns of the output data, shifted by one time step, are added to the dataset. This enables us to eventually overwrite the input column values with the predictions, while saving a copy of the real outputs for calculating the error later. This results in an unknown value at the last time step in the dataset, as the correct value for the last prediction is not recorded. To correct for this the last row in the dataset is removed.

When all of the data is modified uniformly, it is split into training and testing datasets, which are further divided into the inputs and outputs. The input matrices must also be reshaped into 3D tensors with the correct shape in preparation to being given to the neural network as inputs [24].

7.5 Neural network configuration

The neural network structure is what decides the networks performance [24]. This section explains the configuration steps of the neural network structure.

7.5.1 Layer types

As has been described in earlier sections, the model components have multiple inputs, and one or two outputs, and the network will be based on the LSTM neurons. Therefore, the first layer of the network is the input layer, and the hidden layers will be all LSTM layers, but the amount of these layers can vary. The last layer of the network is the output layer. Here a Dense-layer is used to connect all LSTM neurons from the second to last layer into the required number of output neurons.

7.5.2 Network size

In simplicity the number of neurons in a layer increases the number of trainable parameters, which potentially increases the performance of the network, but also increases the training time per epoch. Increasing the number of hidden layers does not necessarily increase the performance, given that the first layer is large enough, but does instead help the training process [24]. Increasing the network size too much might give diminishing results in regard to time spent training as each epoch takes longer with an increasing number of parameters. There is therefore an optimal size of a network, which produces the best result in a reasonable training time.

To optimize the computing resource use an optimal network size was identified. The tested networks had 1, 2 or 3 LSTM layers, and $N = 2^n$, where n = [1..10] neurons per LSTM layer. The models were given one hour to train on the GPU server, and the losses of the best models from each network size were compared. In general, the 2-layered models had the best results, and a layer size of 32 provided a high speed per epoch joined with good performance. Further models were trained with 2 layers of 32 neurons each. Such a model has about 13000 trainable parameters, depending slightly on the amount of model inputs and outputs.

7.5.3 Loss functions

The model error is computed using a loss function. The result of the loss function is used to start the backpropagation algorithm that makes the improvements to the network, so the choice of the loss function will decide what the algorithm thinks are the optimal improvements. The most common loss function used for neural networks is the Mean Squared Error (MSE), which emphasizes and reduces large errors [35].

The training algorithm attempts to optimize the MSE of the training data towards zero through gradient descent. This does not necessarily mean that the network will find generalizations of the dynamics, but it might also overfit to the training data, which means that the model will not function well with other input data, such as the validation data. Therefore the actual performance of the network is followed by measuring the MSE of the validation data.

7.5.4 Optimizer

The optimizer is the algorithm responsible for the training backpropagation process, sometimes called the gradient descent algorithm. There exist multiple versions of the algorithm, and the best ones have adaptive learning rates, which means that they can adjust the gradient descent step size for each parameter, and thus usually converge faster than nonadaptive algorithms. The Adam-optimizer is the usual go-to adaptive optimizer which is easy to use and has among the best convergence speed and quality [24]. Therefore, the Adam-optimizer is used for the models.

7.5.5 Training a model

Initial models were trained for a hundred epochs, which took about an hour or two, but the final models were trained for a thousand epochs, which took roughly 15 hours of CPU time. This was done in order to find further performance increases compared to the models that could be trained in a shorter time.

Since there is no guarantee that the validation loss decreases with every epoch, the best model is not necessarily from the last epoch [24]. Therefore, the models with the best validation MSEs were saved to be analyzed further.

7.6 Model validation process

After the training process had halted, the best performing models were loaded from file, and they were used to create a prediction of the validation data. As the output data is normalized in the preprocessing stage, the prediction outputs are also normalized. After using a model for predicting a sequence, the output normalization is reversed so that the output has the correct units and can be analyzed.

Visual analysis was done in order to understand what the network could and could not predict, and this information was then used to make improvements to the input selection, data preprocessing and model configuration phases. The results of the final model validations are presented in the Section 9.3.

7.7 Complete model

Once suitable models have been trained for all five components, they are used simultaneously to produce the complete simulation which can produce the time series of predictions of each of the outputs. The model components are loaded from their save files, and the models are initialized with known data. One hour of data is used for this, after which the simulation begins. The model components are used one after another to produce their predictions of the next time step. These predicted outputs are written to file, so that they can be used as the inputs of the next time step. This loop is continued for as many time steps as the other input measurements last, and in the end of the simulation the file then contains the predictions of the joined simulation. The results of the joint model are described in the Section 9.4.

8 Experiments

This Chapter presents the experiments which were conducted on the system to create the training and validation data for the models. The experiments were designed so that one experiment was conducted for each of the inputs, during which only that input was altered while the other inputs were kept constant. More complex experiments were also conducted, where all of the inputs are altered throughout a day. The experiment in Section 8.6 was used as the validation data, while the others were used as training data.

8.1 Experiment 1: IT load

The IT load experiment was conducted to examine the effects that changing the IT load setpoint had on the system. The inputs described in Table 12 were tested. The experiment was conducted on 11.09.2019 09:00 - 13.09.2019 01:00 CEST, changing profile to the next every 5 hours. The other setpoints were kept at their normal values described in Table 1.

Time and date	IT load setpoint	IT power consumption
09:00 11.09	0 %	8.84 kW
14:00 11.09	100 %	11.42 kW
19:00 11.09	50 %	10.68 kW
24:00 11.09	0 %	8.85 kW
05:00 12.09	100 %	11.42 kW
10:00 12.09	25 %	9.90 kW
15:00 12.09	75 %	11.09 kW
20:00 12.09	0 %	8.85 kW

Table 12: IT load setpoint experiment inputs.

8.2 Experiment 2: Chiller setpoint

The chiller setpoint experiment tested the effects the chiller to storage tank water temperature setpoint had on the system. The inputs described in Table 13 were tested. The experiment was conducted on 13.09.2019 06:00 - 13.09.2019 14:00 CEST, changing profile to the next every 2 hours. In each of the profiles the other setpoints were kept at their normal values described in Table 1.

Time	Chiller setpoint
06:00	9 °C
08:05	15 °C
10:00	18 °C
12:00	10 °C

Table 13: Chiller setpoint experiment inputs.

8.3 Experiment 3: Pump setpoint

In the pump setpoint experiment the effect of the water flow rate between the module heat exchanger and the storage tank is examined. Due to this setpoint directly affecting the ability of the heat exchanger to remove heat from the module, the experiment was monitored to make sure that the servers did not overheat. The inputs described in Table 14 were tested. The experiment was conducted on 16.09.2019 07:30 - 16.09.2019 14:30 CEST, changing profile to the next every hour. The other inputs were at their normal values during this experiment, except for the chiller setpoint which was left at 10 °C from the chiller setpoint experiment.

Time	Pump setpoint
07:30	100 %
08:30	52 %
09:30	75 %
10:30	85 %
11:30	65 %
12:30	52 %
13:30	100 %

Table 14: Pump setpoint experiment inputs.

8.4 Experiment 4: Fan setpoint

In the fan setpoint experiment the effect of the air flow rate in the data center is investigated. Due to this setpoint directly affecting the air temperature at the servers, the system was monitored to be sure the servers did not overheat. The lower limit of 1 V (from $0 - 10\ V$) was chosen to guarantee that the airflow would be sufficient to cool the servers. The inputs described in Table 15 were tested. The experiment was conducted on 17.09.2019 08:00 - 17.09.2019 16:00 CEST, changing profile to the next every hour. The other inputs were at their normal values during this experiment, except for the chiller setpoint which was left at 10 $°C$ from the chiller setpoint experiment.

Time	Fan setpoint
08:00	10 V
09:00	6 V
10:02	4 V
11:00	8 V
12:00	2 V
13:00	10 V
14:00	1 V
15:00	10 V

Table 15: Fan setpoint experiment inputs.

8.5 Experiment 5: Complex operations 1

This complex experiment was done more arbitrarily than the other experiments. The experiment was done during one workday, changing inputs at roughly regular intervals with arbitrary order and magnitude. Changes to the setpoints are documented in Table 16. The experiment was conducted on 18.09.2019 07:00 - 18.09.2019 15:00 CEST.

Time	IT load	Chiller setpoint	Pump setpoint	Fan setpoint
07:00	0 %	10 °C	100 %	10 V
08:20	100 %	10 °C	100 %	10 V
08:50	100 %	10 °C	52 %	10 V
09:20	100 %	10 °C	52 %	5 V
09:50	100 %	10 °C	75 %	5 V
10:20	100 %	10 °C	75 %	2 V
10:50	100 %	18 °C	75 %	2 V
11:20	100 %	18 °C	75 %	8 V
11:50	100 %	18 °C	52 %	8 V
12:20	100 %	18 °C	100 %	10 V
12:50	100 %	18 °C	52 %	5 V
13:20	100 %	9 °C	100 %	5 V
13:50	100 %	9 °C	100 %	10 V

Table 16: Inputs of the Complex operations 1 experiment.

8.6 Experiment 6: Complex operations 2

This experiment was done similarly to the first complex experiment, but one hour was always given in between the changes to the setpoints to give better chance for the system to adjust to the new setpoints. The Table 17 documents the changes made to the system inputs during the experiment. The experiment was conducted on 19.09.2019 07:15 - 19.09.2019 17:00 CEST.

This experiment was also used as the validation data for the simulation models. To aid in understanding these changes and the effects they have on the real and simulated systems, these inputs as well as the ambient and outside temperatures are presented in a graph in the Figure B1.

8.7 Experiment 7: Complex operations 3

This complex experiment is similar to the Complex 2 experiment, where one hour was given in between changes to the setpoints, with additional changes to the IT load setpoint afterwards. The Table 18 documents the changes made to the system inputs during the experiment. The experiment was conducted on 20.09.2019 07:30 - 21.09.2019 06:00 CEST.

Time	IT load	Chiller setpoint	Pump setpoint	Fan setpoint
07:15	0 %	9 °C	100 %	10 V
08:15	60 %	9 °C	100 %	10 V
09:15	0 %	9 °C	100 %	7 V
10:15	80 %	9 °C	100 %	7 V
11:15	80 %	9 °C	80 %	7 V
12:15	40 %	9 °C	80 %	7 V
13:15	0 %	15 °C	80 %	7 V
15:00	90 %	15 °C	80 %	5 V
16:00	90 %	15 °C	70 %	9 V

Table 17: Inputs of the Complex operations 2 experiment.

Time	IT load	Chiller setpoint	Pump setpoint	Fan setpoint
07:30	0 %	15 °C	70 %	9 V
08:30	40 %	15 °C	100 %	10 V
09:30	0 %	15 °C	60 %	6 V
10:30	70 %	15 °C	60 %	6 V
11:30	70 %	15 °C	60 %	9 V
12:30	70 %	17 °C	60 %	9 V
13:30	70 %	17 °C	52 %	9 V
14:30	40 %	17 °C	52 %	9 V
15:30	40 %	17 °C	52 %	3 V
16:30	40 %	11 °C	100 %	10 V
20:00	90 %	11 °C	100 %	10 V
22:00	70 %	11 °C	100 %	10 V
00:00	30 %	11 °C	100 %	10 V
02:00	60 %	11 °C	100 %	10 V
04:00	0 %	11 °C	100 %	10 V

Table 18: Inputs of the Complex operations 3 experiment.

9 Results

This chapter describes how the models were validated and analyzes the results of the simulations.

9.1 Validation of the physical model components

Each of the components in the Simulink model are validated in this section. The model components are used to run partial simulations of the cooling system using measured data from the experiment: Complex operations 2 that was presented in Section 8.6. This experiment was chosen over the other experiments as it includes sections of chiller and free cooling mode, and changes into all input setpoints with enough settling time between changes. This means that by validating the models using this data, responses to all changes can be analyzed, although not in their entire operational ranges. The inputs can also be seen in Figure B1.

Each of the models are validated individually, by using real measured data as inputs, and the output of each component is also compared to respective measurements of the system. This way the errors in the outputs can be assumed to be purely a consequence of modeling insufficiencies, and not of input errors. A joint simulation in which all components are joined together is also presented later in Section 9.2.

The root mean squared error (RMSE) and standard deviation (SD) of the errors are used to numerically analyze the accuracy and precision of the models. These performance metrics for each of the outputs for the simulation period are presented below in the Table 19. The outputs are also compared with the real values for the entire simulation period in each respective section. In addition the error and error distribution of the models are shown in the Figures C1 and C2. The results in the table and figures are analyzed in detail in their separate sections.

Output	RMSE (°C)	SD (°C)
Hot aisle	0.43	0.43
Cold aisle	0.19	0.19
HE to storage	0.14	0.11
Storage to HE	0.66	0.59
Storage to chiller	0.39	0.22
Chiller to storage	1.93	1.92
Chiller to tower	2.13	2.12
Tower to chiller (locally measured)	3.55	2.86
Tower to chiller (SMHI)	0.85	0.79

Table 19: Individual physical models performance values.

9.1.1 Hot aisle validation

As can be seen from the Table 19, the RMSE of the hot aisle temperature is low, which means that the air temperature on the hot aisle is estimated very accurately.

In addition, the standard deviation is low, indicating that the estimate is also precise. This can also be seen from the Figure 18, where the estimate and measured values are presented. The model does a very good job for most of the simulation period.

As can be seen the largest error happens at t = 31500 s, when the fan setpoint is changed. The estimate settles to the steady state far faster than the measured value, which indicates problems in the thermal lag of heat being transmitting to the ambient and to the equipment. Those heat transfer coefficients are assumed static, but in reality they are dependent on the flow rate of air and the temperature difference, but these dynamics have not been added to the model.

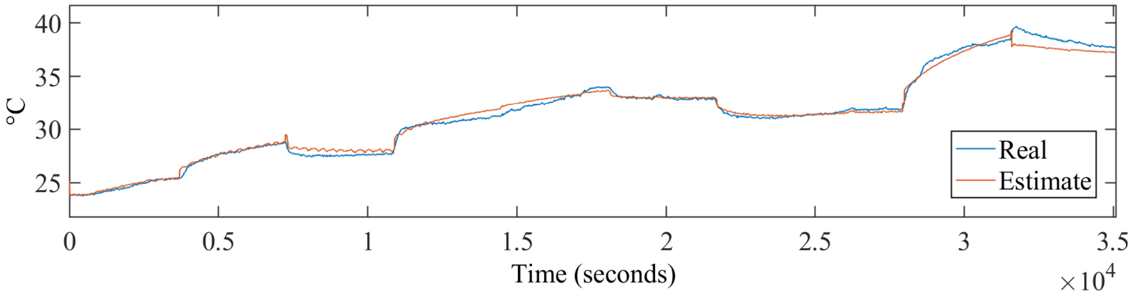

Figure 18: Hot aisle output temperature estimated by the physical model.

9.1.2 Heat exchanger validation

The heat exchanger has two outputs, one of which is the return air temperature to the cold aisle and the other is the water temperature of the return flow to the storage tank. The heat exchanger RMSEs and SDs are low, and as the Figure 19 shows, the estimate is very accurate. The changes in the flow speeds of the air or water flowing through this components do not result in large errors, as the dependency of the thermal conductance of the heat exchanger on them was carefully studied and modeled in Section 6.3. The effect that temperatures have to the thermal conductance could still be added to further improve the performance, as well as the addition of some thermal mass.

9.1.3 Storage tank validation

The RMSE and SD of the storage tank outputs are also low, indicating an accurate and precise model. The estimate is visualized in Figure 20. The estimate for the output towards the heat exchanger can be seen to react too much to the temperature changes of the inflow from the chiller. This happens during the chiller mode when the compressors are turned on and off repeatedly between t = 2500 s to t = 12500 s, and when the 3-way valve is adjusted repeatedly at t = 22000 s to t = 26000 s. Otherwise this output works quite well.

The output towards the chiller has a systematic error, which is also visible from the Figure C2. Additionally, the change in setpoints at t = 31500 s is seen to result

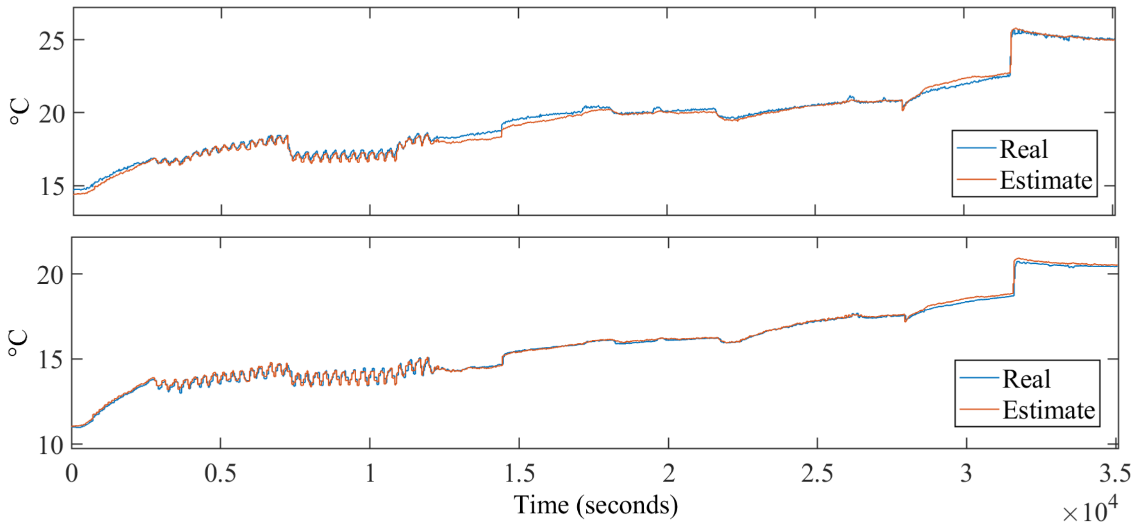

Figure 19: Heat exchanger output temperatures to the cold aisle (on top) and storage tank (on bottom).

in an instantaneous change in the output estimate, even though the real system shows a similar increase in temperature roughly 1000 s later.

Therefore, the method of dividing the tank into two sections and adding a bypassing flow can be seen to be at its limits. Any further improvements should aim to increase the accuracy of the dynamics inside the tank and attempt to add more internal states. Some losses to the ambient air could also be added, even though the effect should be noticeable only on very long time periods.

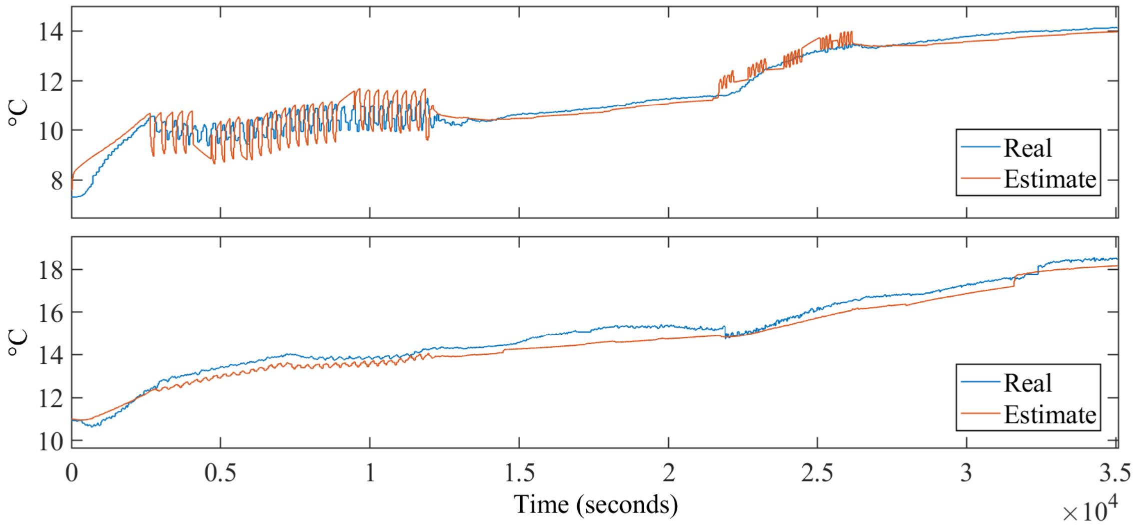

Figure 20: Storage tank output temperatures towards the heat exchanger (on top) and chiller (on bottom).

9.1.4 Chiller validation

The chiller component includes the control logic for all the different modes in addition to the output estimations, and in order to accurately analyze the performance of these subcomponents, first the control logic is analyzed, and then the outputs. Each of the components are tested by using measured data as the inputs, and when the entire chiller model is tested, all dependencies between the components are reconnected.

Free cooling control validation

For the free cooling control signal there is no measurement from the system to compare against. Therefore, its correctness must be analyzed in other ways. The free cooling activation estimate is calculated using the values given in the manufacturers manuals and they are also verified from the chiller control panel menu. The system activates free cooling once during the validation period, at t = 11350 s, as can be seen from the Figure 21. This behaviour is as expected, as based on other measurements the system begins opening the 3-way valve at the same time, indicating that the system is in partial free cooling mode until t = 11850 s, when the chiller quits controlling the compressor on and off.

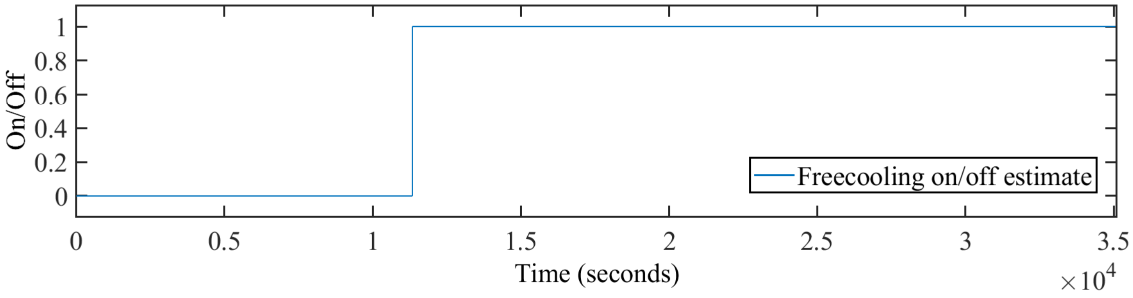

Figure 21: The free cooling activation estimate.

Compressor control validation

The period when the compressor is controlled coincides with the period when it is measured to happen, as is seen in Figure 22. By analyzing the closeup, it is visible how there are small errors in when the compressor is active.

The compressor gets activated most often during the same time step as the measurement indicates, but also sometimes some time steps before. There are even larger differences in the deactivation times. Since the estimate is calculated using the logic provided in the operation manuals and using the values readable from the chiller control panel, the difference should be due to other reasons. The measurement of the compressor activation is an estimate derived from the chiller power use measurement. This value is directly related to the state of the compressors, and a shift in the sampling times can only account for a maximum of 30 seconds of the discrepancy to either direction. The remainder of the discrepancies cannot be accounted for.

3-way valve control validation

The estimated behavior of the 3-way valve is presented in Figure 23. The estimate is accurate when the system is in chiller mode, and the valve is closed. When the mode changes to free cooling mode, the estimate begins to open and close the valve

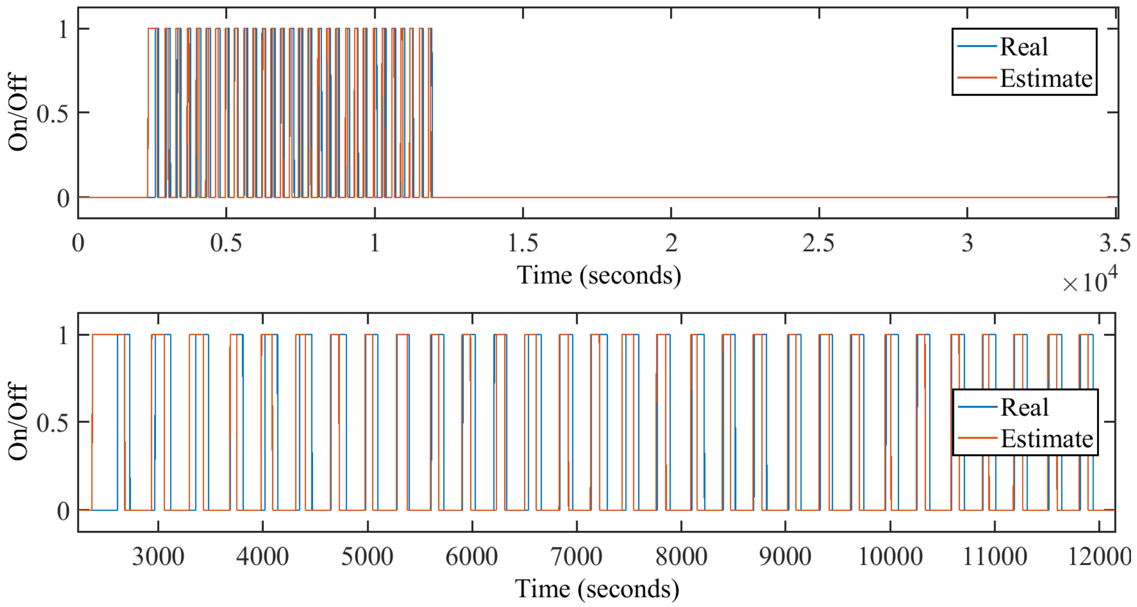

Figure 22: The compressor on/off signal estimate, as well as a closeup.

earlier than the measurements indicate. The valve is then correctly kept open until t = 21550 s, when the modulation of the valve resumes. What happens during this period is best seen from the closeup in Figure 23.

The valve is modulated open and closed quickly, about once every two minutes according to the measurements and estimates. Therefore, precise validation is difficult with the sampling time of the measurements being 30 s. Nevertheless, it can be seen that the modulation estimate occurs during the same time period and at the same frequency as the measurements indicate. After the modulation phase the estimate closes the valve and begins to modulate the opening towards the end of the simulation, but the measured behavior keeps the valve open continuously. The reason for these discrepancies is unknown, as the simulation follows the logic described in the manufacturer's manuals, and the values are verified to correspond to the ones in the chiller control panel menu.

Chiller output validation

To validate the chiller outputs the measured control signals were used as inputs, except for the free cooling activation, which is not measured. The reason for doing this is to be able to accurately compare the temperature calculations, without any possible errors in the operation mode estimates affecting that result. The outputs are presented in Figure 24.

As can be seen from the Figure 24, the estimated output to the storage tank is fairly accurate throughout the simulation period. During the partial free cooling mode (from t = 11350 s to t = 11850 s) the output can be seen to not make the characteristic dip in temperature from the compressors going on, as the partial free cooling mode is not implemented in the chiller model. The reasoning behind this is explained in Section 6.5. A small systematic error can also be seen during the end of the simulation, beginning at t = 26200 s. The chiller is in free cooling mode

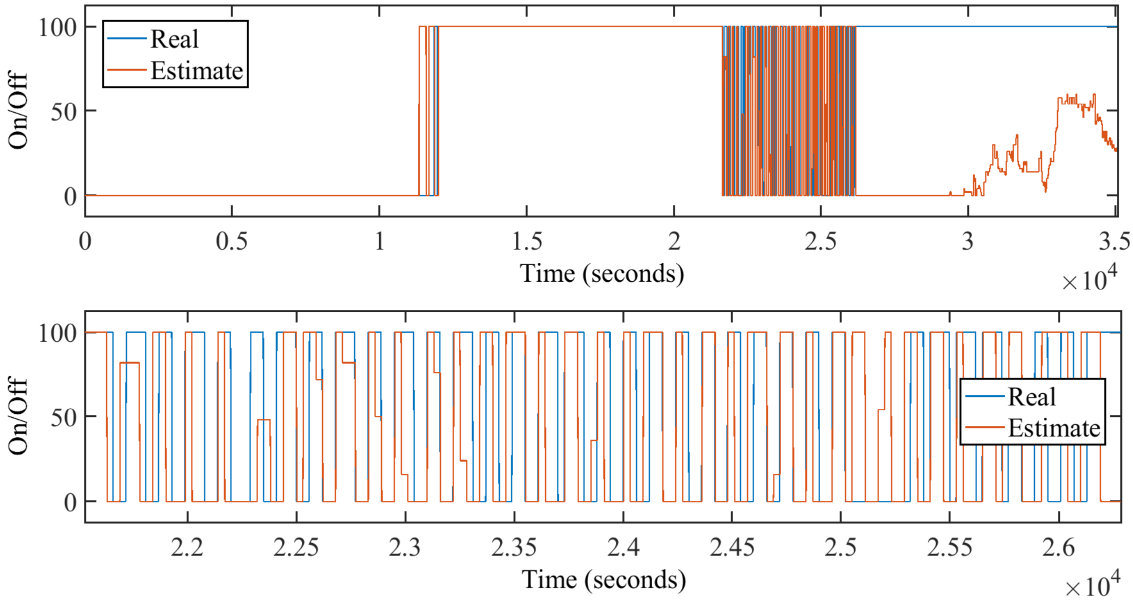

Figure 23: The 3-way valve control signal and its estimate, as well as a closeup.

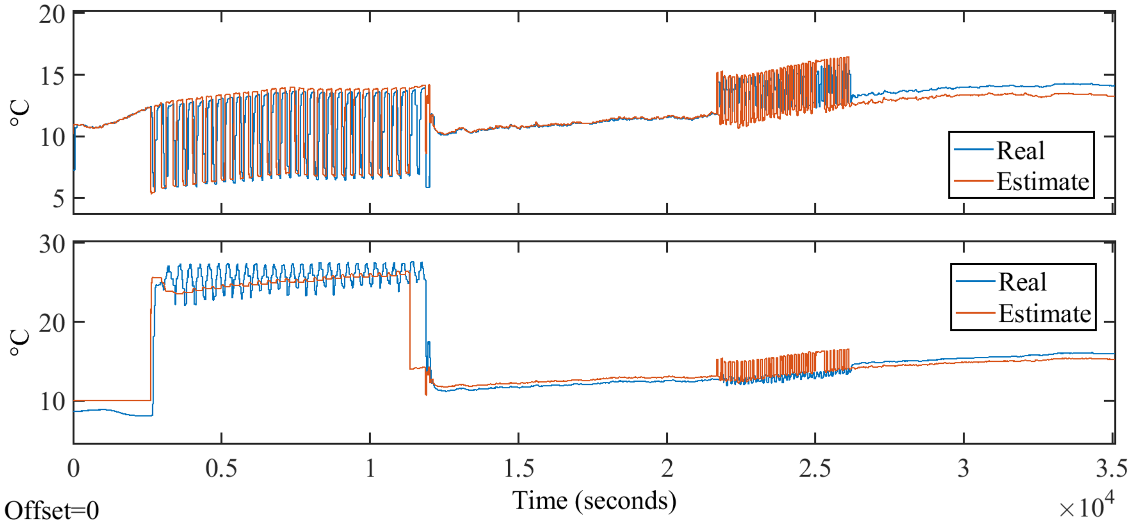

Figure 24: Chiller output temperatures towards the storage tank (on top) and cooling tower (on bottom).

from t = 11350 s to the end of the simulation, but the free cooling glycol mass flow decreases at t = 26200 s to the level it usually is during chiller mode. As no changes in the flow were expected without a change in mode, the thermal coefficient of the chiller heat exchangers are modeled to be static, and therefore the temperature cannot correctly be estimated with the new flowrate. Further observations of the flowrates are explained in Section 6.7.

The estimate of the output towards the cooling tower is presented in the lower graph of Figure 24. Initially the output temperature is unchanging since the glycol does not flow when the pump is off. The temperature in the sensor can be seen to change even with no flow, possibly due to heat being transmitted through the insulation from the surrounding air to the pipe. At t = 2640 s the compressors and the pump are started, and the output temperature estimate increases roughly to the correct level, but does not make the slight decrease in temperature every time the flow stops. When the partial free cooling mode is activated, the model instead uses total free cooling dynamics for its estimate, which results in the premature decrease in the temperature. The estimate can also be seen to be more aggressive in reacting to the 3-way valve being modulated between t = 21690 s and t = 26200 s than the real system.

Joint chiller validation

All of the chiller components are tested together with their internal dependencies to see the true outputs of the chiller model. In Figure 25 the control signals of the chiller are shown, and in Figure 26 the output temperatures.

As can be seen from the Figure 25, the compressor activation has a higher frequency, indicating that the changes to the estimated output temperature happen faster than in the real system. The lack of a partial free cooling mode is still visible from the results at t = 11350 s. The free cooling valve does not close during the rest of the simulation when the estimated temperatures are used as the inputs for the control logic. This indicates that the output temperature towards the storage tank does not increase enough during the modulation phase.

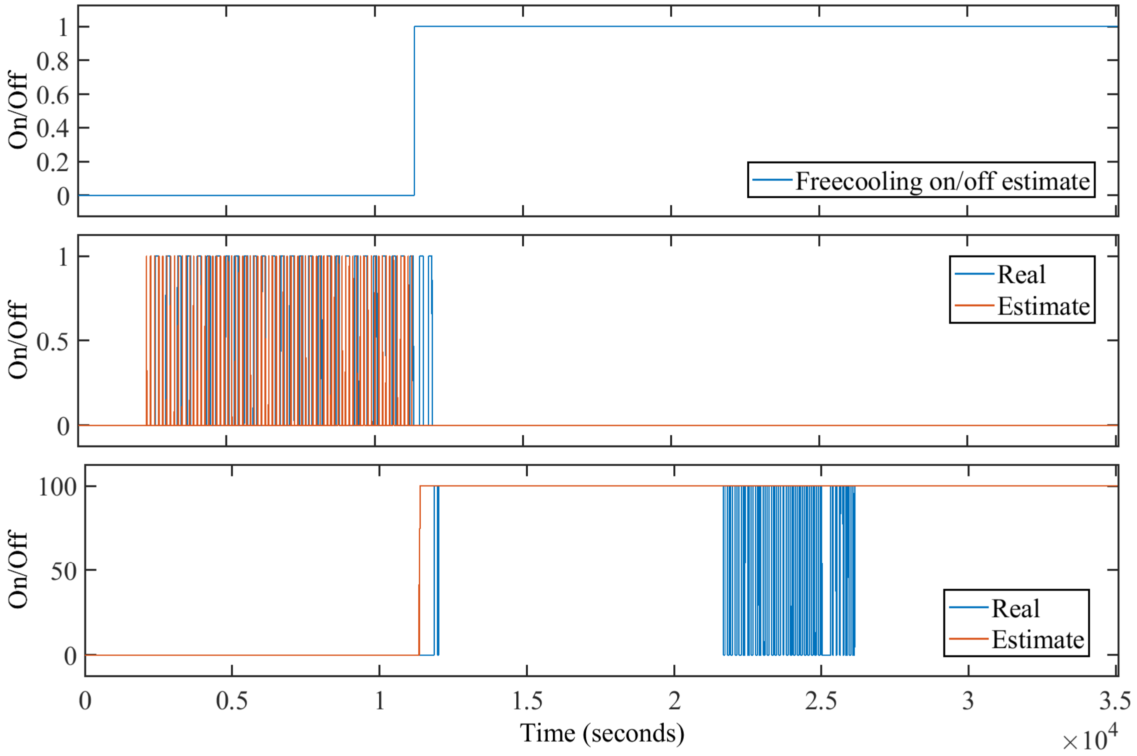

Figure 25: Chiller operation mode control signals for the joint chiller validation. Free cooling (on top), compressor (in middle) and 3-way valve (on bottom) activations.

The output temperature estimates in Figure 26 are not affected much from the simulation where the correct control logic was provided for the system. The 3-way valve open/closed modulation was removed and instead it is held open from t = 21690 s to t = 26200 s, so the rapid changes of the output temperatures during this time also are reduced.

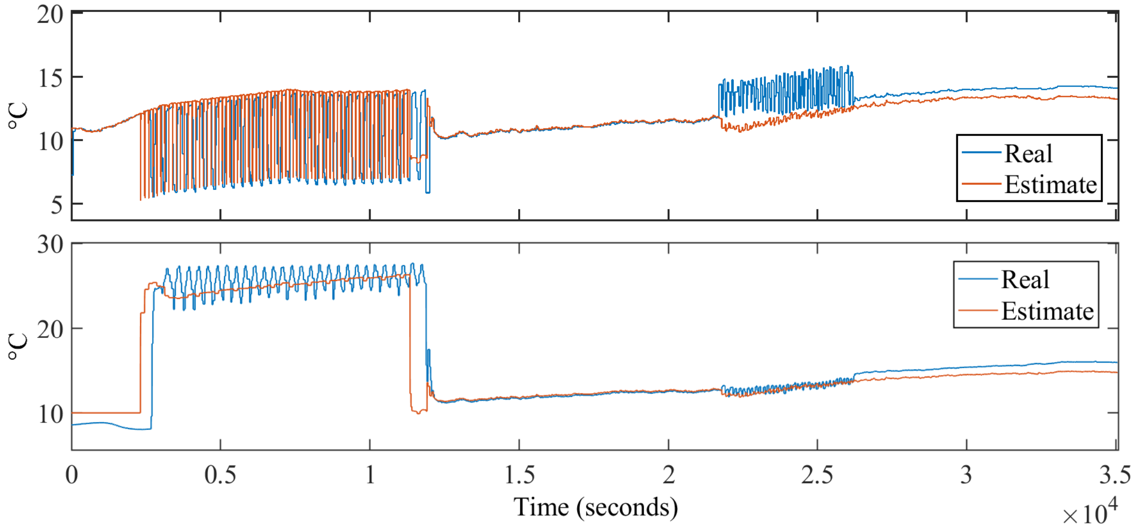

Figure 26: Chiller outputs towards the storage tank (on top) and cooling tower (on bottom) for the joint chiller validation.

The chiller outputs may be improved by adding dynamics for the partial free cooling mode, by adding the missing dynamics for when there is no flow, and the thermal coefficients of the heat exchanger could be made dependent on the flowrates, as was done to the module heat exchanger. In addition, the thermal lag in the heat exchangers during the rapid changes that the compressor or 3-way valve affect should be calibrated further to match the lag of the real system.

9.1.5 Cooling tower validation

The cooling tower glycol output temperature estimates are presented in Figure 27. There are two graphs presented, the first one uses the local measurements for the outside temperature and the second the Swedish Meteorological and Hydrological Institute (SMHI) measurements from the local measurement station at the Luleå Airport [36], which is located 7.66 km away. There are two reasons for doing this. The first reason to this is that the temperature sensor located on the roof of the building that is used by the measurement network often receives too high temperature measurements during the morning, when the sun shines on the side of the cooling tower where the sensor is located. This happens during the validation data from t = 0 s to t ≈ 10800 s. To counter this known error in the measurement, the simulation is tested with other temperature measurements as a control. The other reason is that the simulation should be able to be used to predict to the future, using temperature predictions which might not be provided for the exact location of the data center.

By analyzing the figure, it is clear that the local measurement provides the better estimate accuracy when there is no error in the measurement from direct sunlight, but the other measurement functions better during the start of the simulation. This indicates that the estimator functions correctly when it receives the correct input data. As the SMHI weather station is located some kilometers away, local temperature differences give a slight error to the SMHI estimate most notably during the end of the simulation.

The remaining errors occur when the pump is regulated on/off by the chiller and the glycol mass flow goes quickly to and from zero. This change in flow changes the overall thermal conductivity of the heat exchanger, but it is modeled to be a constant. The addition of this dynamic would increase the accuracy of the cooling tower model. When there is no flow the estimated output temperature is of the heat exchanger temperature, but the sensor which the output is compared against is inside the building, almost at the chiller, and therefore the dynamics affecting it are different when no flow is present.

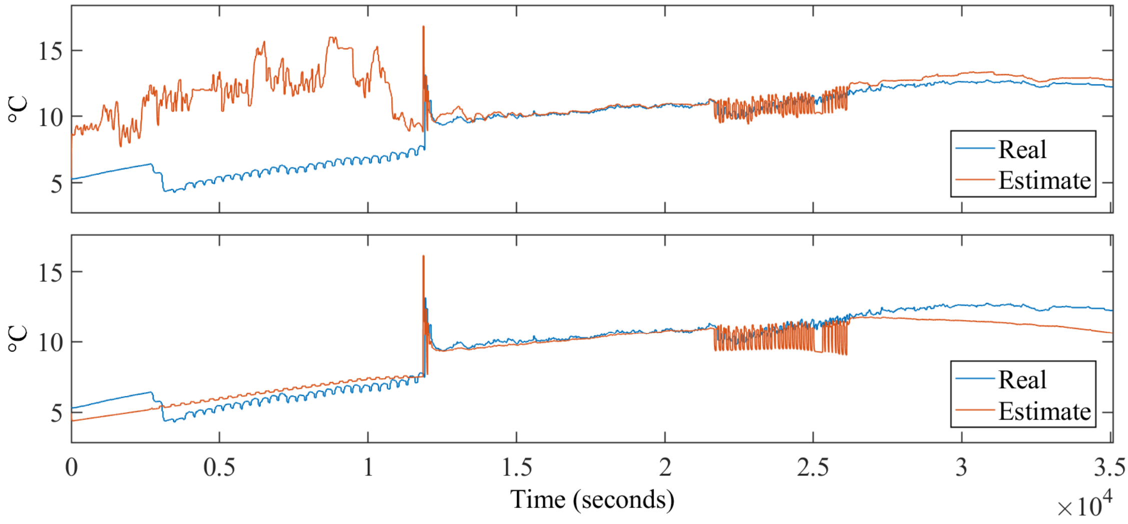

Figure 27: Cooling tower glycol output temperature estimate. First estimate is calculated using the outside temperature measurements from the local sensor, and the second using the SMHI measurements.

9.2 Physical model simulation

The physical models are connected with each other to create the joined model that can simulate the whole system. The outputs are connected to their respective inputs using the delays calculated in Section 6.8. Therefore, no measured temperatures must be input during the simulation, except for the initial values of all the various components which can be initialized. The simulation then takes as inputs only the IT load, fan, pump and chiller setpoints, as well as the ambient and outside temperatures. Everything else is calculated in the simulation.

The performance of the outputs of this simulation are presented in Table 20, and the visual representation of the outputs is presented in Figure 28. If the performance values are compared with the individual model performance in Table 19 all of them indicate poorer performance than when validating the individual components. The reason is that the errors of each component get propagated back to the system, resulting in the errors growing with time. The outputs stay stable even though the errors are occasionally large during the simulation, and after the nearly 10-hour simulation the largest error in the outputs is only 2.1 °C. The physical model dynamics therefore seem to be good at limiting the error from increasing through various means such as the convection losses to the environment.

As can be seen from the RMSE in Table 20 and from Figure 28, the first five outputs obey roughly the same form, and their errors in Figure C3 are also of similar sizes. This leads to the conclusion that the part of the model responsible for estimating these outputs behaves appropriately, and can function stable even though the other components have larger errors. The large errors are visible in the chiller outputs, where the chiller to tower output does not represent reality during the chiller mode, and the tower model, where the incorrect outside temperature input causes errors during the simulation. This also is the reason for the high error SD seen for the chiller and cooling tower outputs.

Output	RMSE ($°C$)	SD ($°C$)
Hot aisle	1.68	1.04
Cold aisle	1.70	0.93
HE to storage	1.76	0.90
Storage to HE	1.70	0.90
Storage to chiller	1.27	0.72
Chiller to storage	2.64	2.62
Chiller to tower	3.25	3.06
Tower to chiller	3.89	2.99

Table 20: The joined physical model performance values.

9.3 Data-driven model validation

Already during training the data-driven models compared their performance against the training and validation data by calculating the MSE loss function. The MSE of the training data was used by the algorithm to improve the network weights, and the validation MSE was used to keep track of the generalization of the solution, and the best models with the lowest validation MSE were saved. The validation MSE is of interest also after the training is complete, as it was used to calculate the RMSE to analyze the performance of alternative model configurations, together with the plots of the prediction over the validation period. The validation data is from the same experiment as is used for validating the physical simulation models, the experiment

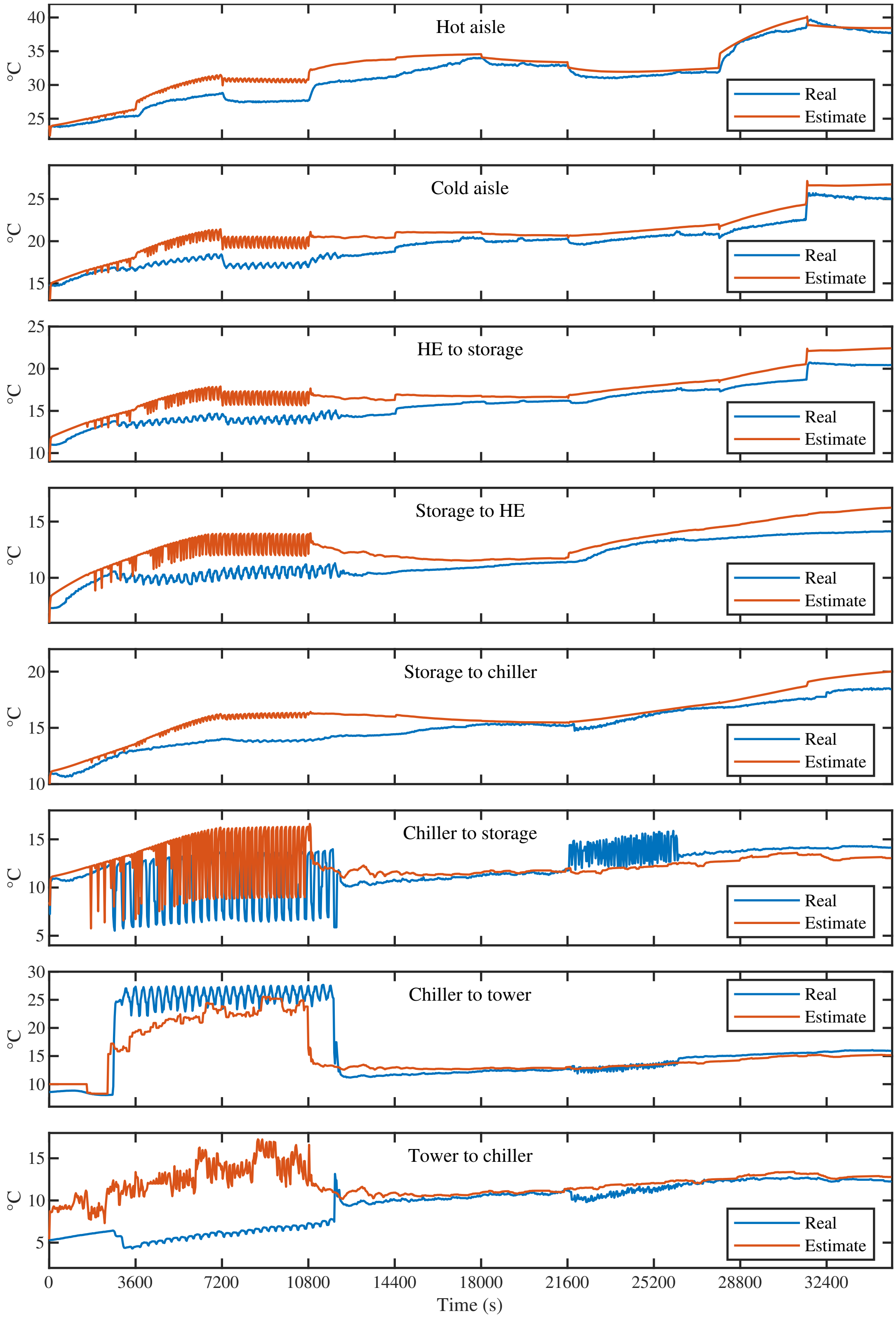

Figure 28: Joined physical model simulation outputs.

Complex operations 2, which is presented in the Section 8.6. Improvements on each model could be made by taking more measurements of the system to be used to train the models. Currently the models are trained with a time series that represents only 93.5 hours of operation and validated using 9.75 hours of data.

The initialization period of one hour was taken into consideration when calculating the RMSE and SD, and those errors have been excluded when the values were calculated. The values are presented in the Table 21, and the predictions in the Figure 29. In addition, the errors are presented in the Figure C1 and the distribution of the errors in Figure C2.

Output	RMSE (°C)	SD (°C)
Server to HE	1.72	1.43
HE to server	0.48	0.40
HE to storage	0.39	0.35
Storage to HE	0.63	0.50
Storage to chiller	0.35	0.31
Chiller to storage	1.20	1.20
Chiller to tower	0.97	0.76
Tower to chiller	1.46	1.45

Table 21: The performance values of the data-driven model components during validation.

Server model validation

The server to heat exchanger output can be seen to take roughly 1.5 hours to fully initialize, after which the estimate is fairly accurate until at t = 21600 s, when the IT load is reduced, and the estimate begins to gain an error. The changes made to the IT load and fan setpoint at t = 27900 s also result in the error increasing. This is also the reason why the RMSE of this output is quite high, as for the first half the estimate is more accurate. The reason for why the estimate does not stay accurate until the end of the validation can only be hypothesized, but more training might find a better model that also can successfully estimate these changes. Other alternatives are to add more inputs to the model or to give it more data to train with.

Heat exchanger model validation

The heat exchanger estimates can be seen to function well. The estimates take almost two hours and under one hour respectively to successfully initialize, and from there on they have only slight errors during the simulation. No specific change in the inputs can be seen to negatively affect the estimates, which indicates that the model works at the least in all the input states in the validation data.

Storage tank model validation

The storage tank outputs take under an hour to initialize, and the model estimates the outputs fairly well throughout the validation period. The output towards the heat exchanger should have larger oscillation of the temperature when the input temperature from the chiller also oscillates, but this is not seen in the estimate. Additionally, the changes that result from the IT load and chiller setpoint being

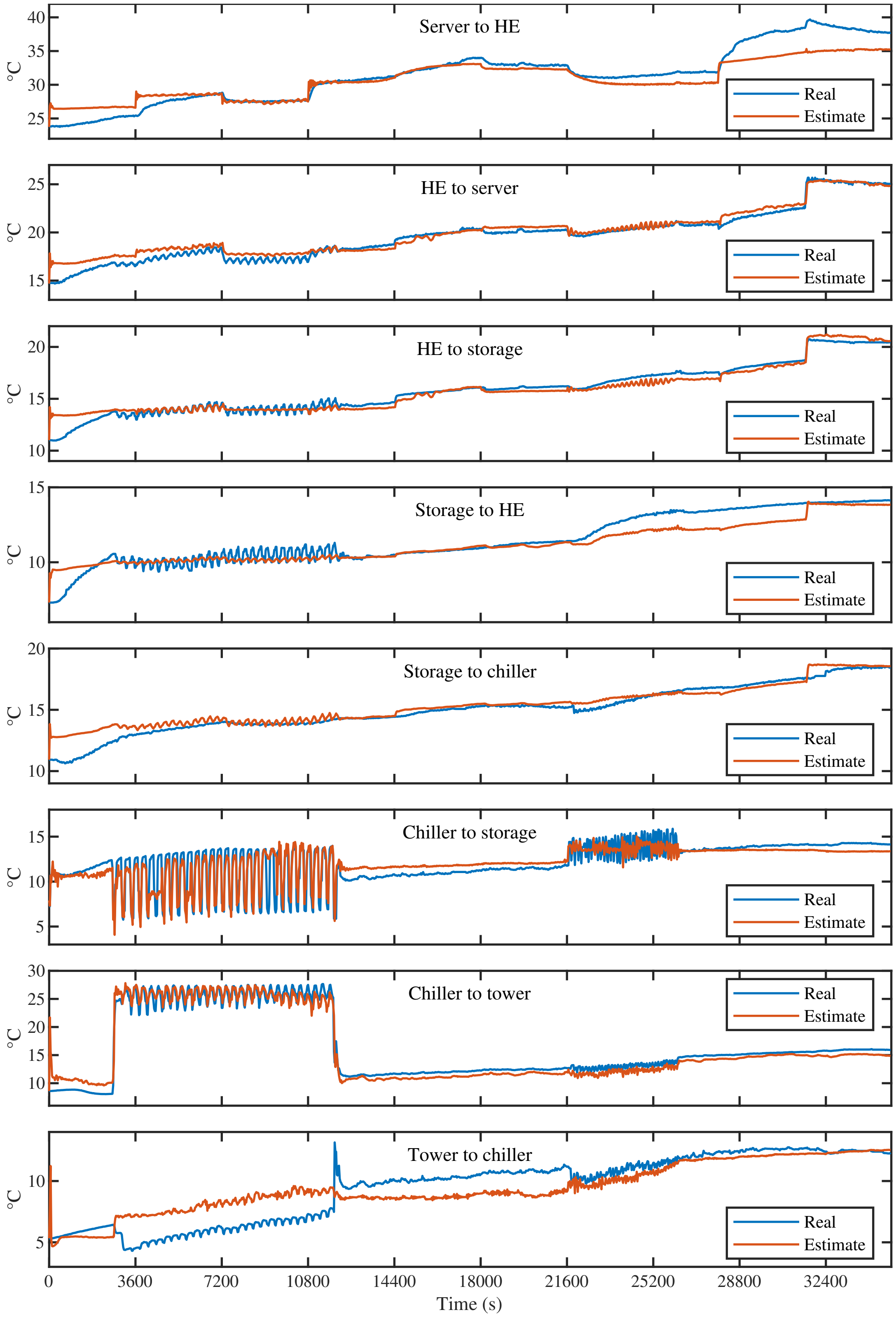

Figure 29: Verification results of the data-driven model components.

changed at t = 21600 s are not entirely conveyed to the estimate, but the error is later corrected at t = 31500 s, when the pump and fan setpoints are changed. The reason why the model does this is probably due to it not understanding the dynamics in all ranges of inputs. The addition of more training data or retraining of the model could increase performance in also this specific combination of inputs.

Chiller model validation

The chiller outputs can be seen to follow the real values rather well, and the initialization takes about 45 minutes until the first change happens to the system in the form of the compressors activating, and the model learns of the internal state. There are small gaps in the input data where the values in the gap are estimated linearly, which results in some of the estimated oscillations to be of smaller amplitude. In addition, the output has minor biases during the free cooling mode.

This chiller model required the use of the chiller outputs also as the model inputs, which risks the model becoming a persistence model, where the input temperature would be directly sent as the output without any change. Models which were trained without these inputs were not able to successfully model the oscillating behaviors, and instead predicted some constant value for the duration of the oscillation. How this chiller model functions, and whether the decision to include these inputs was sensible can be seen when the inputs are looped back in the joint simulation.

Cooling tower validation

The cooling tower output can be seen to have large errors during most of the verification period. The estimate suffers from the errors which are present in the input data. The outside air temperature gives higher than real measurements during mornings (from the validation data start to roughly t = 10800 s), when the sun can warm up the sensor to temperatures higher than the outside air really is. Nevertheless, the model can be seen to have learned some of the dynamics as the estimation converges with the real temperature at t = 21600 s. Fixing the sensor and using new data to retrain the model could improve the performance, or the local measurement could be changed to some other outside temperature measurement, such as to the SMHI measurements.

9.4 Data-driven model simulation

The results of the joined data-driven model simulation, which is comprised of the five data-driven model components, is presented in this section.

The models are used together as is described in the Chapter 7, and then simulated against the verification data. The first hour of the data is used as the initialization period, and during the rest of the simulation the estimates are fed back to the models in a feedback loop. The performance of this simulation is summarized in Table 22, and shown in detail in Figure 30.

As can be seen from the performance values in Figure 22, the errors grow considerably from the validation errors when the model components are joined in this way. The outputs of the simulation in the Figure 30 show how the first hour of data is the initialization period, where the feedback is not done, and the rest of the simulation is the actual simulation. It is clear that the simulation slowly loses

Output	RMSE (°C)	SD (°C)
Server to HE	2.30	1.79
HE to server	1.72	1.35
HE to storage	1.82	1.19
Storage to HE	1.55	1.06
Storage to chiller	1.09	0.85
Chiller to storage	2.55	2.55
Chiller to tower	4.31	3.04
Tower to chiller	4.08	3.06

Table 22: The joined data-driven simulation performance values.

accuracy with time after the initialization period ends at t = 3600 s.

After the initialization period the outputs have a good accuracy during the next two hours, except for the chiller outputs which mostly stop their oscillating behavior, and the cooling tower output estimate which has an error throughout the simulation due to the errors in its inputs (this is explained more in Section 9.3). This means that the chiller model could not sustain the compressor on/off behavior without receiving the real inputs as it did during validation. Therefore, a chiller model which did not have the output temperatures as its inputs could have performed largely the same. The chiller outputs begin showing more errors when the compressor on/off phase stops, as the chiller outputs continue oscillating, and the other outputs also gain the bias from the incorrect cooling tower output. The chiller model did not show the oscillation while the model was validated, and it must be due to the feedback loop of the chiller outputs being fed back to the chiller. Therefore, the chiller model should be improved by finding a way to remove the feedback loop, while trying to keep the oscillation accuracy.

The simulation begins slowly diverging due to the extra chiller output oscillation and bias from the cooling tower output at t = 11850 s, until the changes made to the inputs at t = 21600 s give the chiller to storage output an even larger error, which accelerates the divergence of the estimates. The simulation ends with the estimates having gained considerable errors, and the chiller outputs still oscillating.

The joined data-driven model becomes inaccurate also partly due to the setpoints being changed during the simulation, since with each passing change the errors become slightly worse. This indicates that the models have better accuracy for steady state simulation than dynamic control. In addition, the use of the feedback loop in modeling the chiller model can be seen to be a bad idea. Furthermore, as long as the cooling tower model inputs have errors, the errors will propagate into the simulation and slowly increase the errors of the entire simulation.

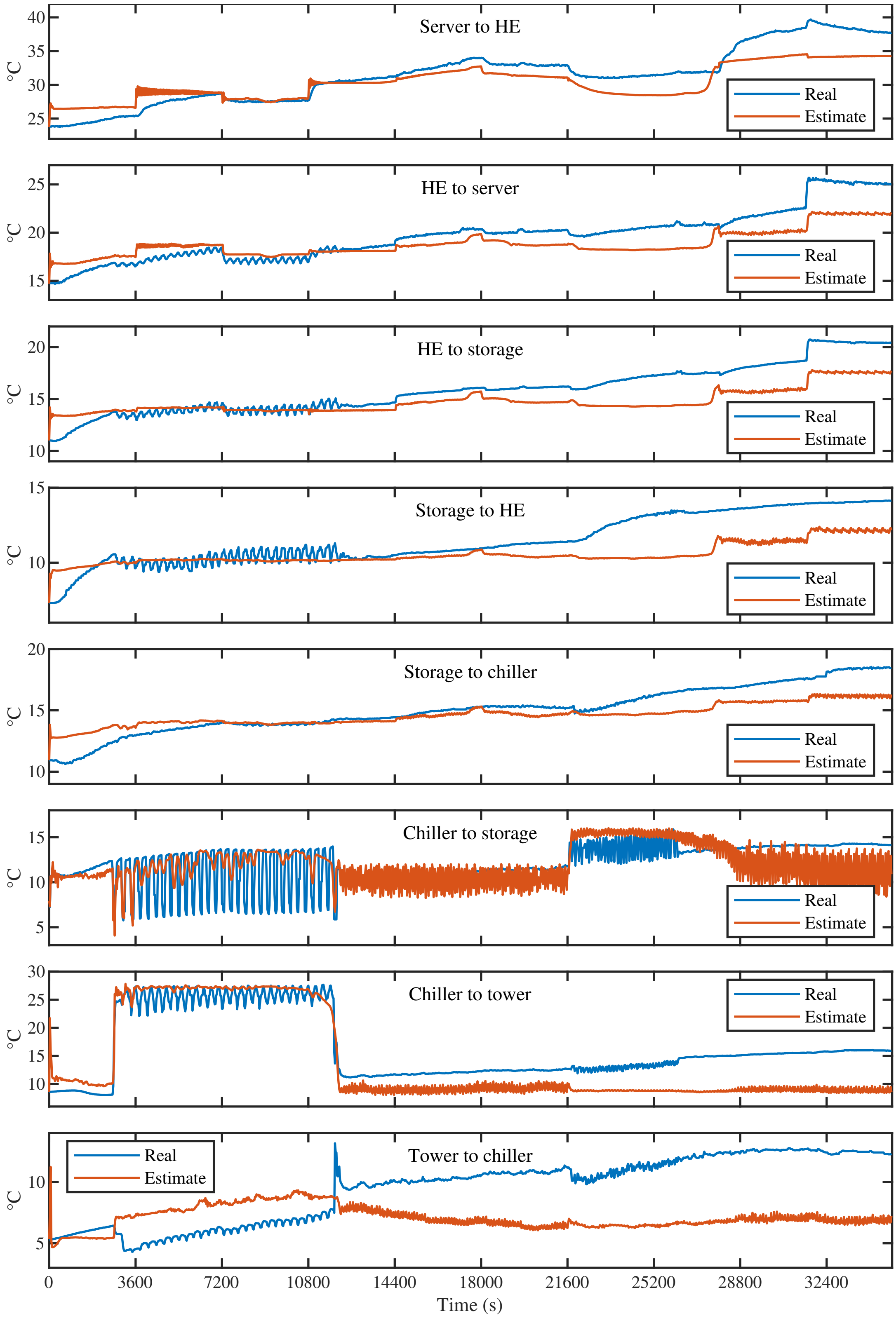

Figure 30: The joined data-driven simulation outputs.

9.5 Comparison of the two modeling methods

9.5.1 Comparison of the model components

The validation root mean squared errors and error standard deviations of the physical and data-driven model outputs are gathered into Table 23 for comparison.

Output	Physical		Data-driven	
	RMSE (°C)	SD (°C)	RMSE (°C)	SD (°C)
Hot aisle (Server to HE)	0.43	0.43	1.72	1.43
Cold aisle (HE to server)	0.19	0.19	0.48	0.40
HE to storage	0.14	0.11	0.39	0.35
Storage to HE	0.66	0.59	0.63	0.50
Storage to chiller	0.39	0.22	0.35	0.31
Chiller to storage	1.93	1.92	1.20	1.20
Chiller to tower	2.13	2.12	0.97	0.76
Tower to chiller	3.55	2.86	1.46	1.45
Mean	1.18	1.05	0.90	0.80

Table 23: Validation accuracies of the physical and data-driven models.

When the two modeling methods are compared with each other by looking at the validation performance through the RMSEs, no clear winner can be directly proclaimed even if the means are lower for the data-driven models. The differences in the metrics show that some components function better using the physical modeling method, and some using the data-driven method.

The standard deviations of the errors are usually quite close to the RMSE values, which indicates that the RMSE is mostly due to the errors having a spread out distribution, and not due to systematic errors. The error distribution is further shown in Figure C2, which shows how the models which perform best have a tight distribution and low mean of error. Using this graph, it can be seen just how the distribution is formed. An error plot is also prepared in Figure C1. The two methods are also compared visually (using all figures in Sections 9.1 and 9.4).

The simplest models, the server and heat exchanger models, were the two physical models which win in performance against their data-driven counterparts. Comparing the errors and error distributions also indicate that the physical models perform more accurately. The performance of these physical models indicates that a system which can be represented in simple thermodynamics equations can perform well, and even better than a data-driven model.

The storage tank accuracies are very close to each other, but the data-driven approach is slightly more accurate according to the RMSEs, while both methods each have one output with the lower SD. By comparing the errors, it is seen that the physical model performs roughly the same during all sections, while the data-driven has problems with certain input combinations.

The most complex model out of the ones modeled is the chiller model, where the data-driven model has the clear advantage in all metrics. The chiller outputs have quickly oscillating values with large amplitudes, which are difficult to model in a physical model and can increase the RMSE a lot. The data-driven model has smaller errors during these periods, which indicates that it can model these dynamics more accurately, whereas the physical model has a smaller error during the more stable periods.

The cooling tower output is better estimated by the data-driven model when the accuracy metrics are compared, but the error and error distribution graphs show that the physical model works better when the input temperatures are accurate. This is not entirely fair for the data-driven model, as it has been trained with the erroneous data, and therefore cannot perform as it could if the error was not present.

The comparison of the validation results provides some insight into how the two methods compare to each other, and also show how the two methods can both function well. Since both methods have their own inadequacies that can be improved on, no definite conclusion can be made based on the validation results.

9.5.2 Comparison of the simulations

The root mean squared errors and error standard deviations for the joint physical and data driven model simulations are gathered into Table 24 for comparison. In addition the mean absolute errors (MAE) are calculated, as it is a common metric used to rate other similar simulation models [3], [4].

Output	Physical			Data-driven		
	RMSE	SD	MAE	RMSE	SD	MAE
Hot aisle (Server to HE)	1.68	1.04	1.35	2.30	1.79	1.84
Cold aisle (HE to server)	1.70	0.93	1.45	1.72	1.35	1.44
HE to storage	1.76	0.90	1.53	1.82	1.19	1.48
Storage to HE	1.70	0.90	1.45	1.55	1.06	1.20
Storage to chiller	1.27	0.72	1.05	1.09	0.85	0.83
Chiller to storage	2.64	2.62	1.71	2.55	2.55	1.90
Chiller to tower	3.25	3.06	1.86	4.31	3.04	3.76
Tower to chiller	3.89	2.99	2.50	4.08	3.06	3.81
Mean	2.24	1.65	1.61	2.43	1.85	2.03

Table 24: Simulation accuracies for the joined physical and data-driven models.

Judging by the performance metrics, the physical models are more often the more accurate model, with lower scores than those of the data-driven model. The means are also lower for the physical model. But using the performance metrics alone to compare the models is not enough, and the outputs in Figures 28 and 30 and the errors in Figure C3 are also compared.

When the figures are compared, it is clear that the data-driven model produces the more accurate estimates during the first half of the simulation, but as it continues to diverge, the physical model estimates instead reconverge with the real values for a while, and then again diverge at the end of the simulation. This behavior tells a lot about the two models.

The physical model errors increase quickly at the start of the simulation, when the system is in chiller mode, and when the free cooling mode is activated the errors begin decreasing. This indicates that the errors in the physical model during the chiller mode are the reason for why the errors increase, and that the model performs better in free cooling mode. The data driven model works roughly the same in all modes, as the divergence of the estimates increases gradually. The data driven model is therefore more consistent in estimating the outputs of the system. The physical model however might be more accurate during free cooling mode.

9.5.3 Comparison with the state of the art

The accuracies of other DC cooling system simulation models are most often reported using the mean absolute error or a percentage error for a roughly 12-hour steady state simulation, where the system modes and controls are constant, but the load might change [3], [4]. However, these models do not simulate the entire cooling system or might assume some ideal behavior for some of the components further from the DC servers.

The MAE that has been reported from state-of-the-art physical models is 1.1 °C [3] or 1.7 – 0.5 °C [4] for the cold aisle temperature, and 1.15 °C [3] for the hot aisle temperature. Therefore, the accuracies of the simulation models of this thesis are in the same range as what has been reached before, but achieve only slightly worse results during the 9.75-hour long simulation. During this simulation the internal controls were however adjusted to test multiple operating regimes, and the simulations in this thesis simulate the complete cooling system, instead of the partial system as is done in the other models. Therefore, this simulation trades some accuracy for more insight to the internal states of the entire cooling system, when compared with the state-of-the-art physical models.

An ANN solution which was designed to estimate the cold aisle temperature for individual disconnected points in time has also reached a MAE of 0.6 °C on the cold aisle temperature [7]. However, this model performed extremely poorly when a predictive time-series like the ones in this thesis was attempted, and the estimate error grew to over 5 °C in under 300 seconds. As such, the true MAE of that model when performing a predictive time-series on the time scale of hours would have extreme errors and would be unusable for any long-term predictions. Therefore, the models presented in this thesis both perform much better than a state-of-the-art neural network model.

10 Conclusion

The goal of this thesis was to develop simulation models for a data center cooling system using data-driven and physical modeling methods, to validate them using a real system, and to compare their accuracies with each other as well as other recently published models. For this purpose, two simulation models for the data center cooling system were created. The models were made using state-of-the-art methods: The physical model was created with Matlab and Simulink, and the data-driven model using LSTM neural networks with TensorFlow and Keras. The simulations were validated against the Edge data center laboratory located at RISE ICE research facilities in Luleå, Sweden. The cooling system consists of the data center, an air-to-water heat exchanger, a water storage tank, a chiller and a cooling tower.

Most of the individual components of the simulation models perform very well during validation, and most of the outputs have root mean squared errors (RMSE) and standard deviations (SD) of under 1 °C over a simulation period of 9.75 hours. There were however challenges in correctly simulating the chiller using both methods, and the accuracies of those models were poorer than the rest. In addition, it was noticed that an incorrectly installed temperature sensor decreased the accuracy of the cooling tower models.

When joining the model components to create the joint simulations, an error in any of the components will be propagated to the other components and lower the accuracy of all outputs. Therefore, the MAE for the physical and data-driven model simulations in this thesis are 1.45 and 1.44 °C for the cold aisle temperature, and 1.35 and 1.84 °C for the hot aisle temperature, respectively. The other six model output temperature MAEs as well as the RMSE and SD accuracy metrics are presented in Table 24. By comparing all of the accuracy metrics, the physical model achieves a higher accuracy than the data-driven model. In the introduction of this thesis the data-driven model was hypothesized to achieve the better accuracy, and therefore that hypothesis is shown to be invalid for now. However, both models contain some inadequacies, and the hypothesis could still be proved valid if the inadequacies with the models are solved.

The simulation models perform accurately enough to be able to predict the system responses hours into the future. The data-driven predictions were mostly accurate for 5 hours, whereas the physical predictions lost accuracy already after one hour, but regain some accuracy later. However, it should be noted that the accuracy and time it takes for the predictions to diverge depend on how much the simulation conditions change during the simulation, and more stable conditions would provide even better results. During the validation period the setpoints were adjusted much more than during any normal operation.

The simulation accuracy of the models prepared for this thesis can be compared with the state of the art by comparing the hot and cold aisle temperature MAEs. The state of the art calculates the MAE from a 12-hour simulation of normal stable cooling system operation, while the simulations of this thesis are 9.75 hours, throughout which all of the system inputs are actively altered. The state of the art MAE is 1.1 °C [3] or 1.7-0.5 °C [4] for the cold aisle, and 1.15 °C [3] for the hot aisle temperature.

According to these values the accuracy of the models presented in this thesis fall slightly behind the models in the state of the art they are compared against. However, the models in this thesis also provide temperature estimations for the other cooling system components, which lower the accuracy of the compared metrics by adding to the model complexity. The other estimates this thesis presents are not usually predicted by the state of the art, and the chiller and cooling tower are instead assumed to have a constant cooling effect, and no other system has contained cooling storage tanks. The accuracy of these other predictions cannot therefore be compared with previous work.

State-of-the-art artificial neural network models that estimate the thermal response of a DC have terrible performance and are outright unable to predict an accurate time-series on the time scale the models in this thesis do [7]. Therefore, the models in this thesis, and especially the LSTM-model has been shown to surpass the earlier state of the art in using neural network models to predict a thermal response in a data center.

Future work that would attempt to improve the accuracy of these models should first fix the outside temperature thermometer so that sunshine cannot distort the results by warming the outdoor temperature sensor. After the temperature sensor is fixed new data must be gathered for the data-driven model, and it should be retrained. Additionally, the data-driven model accuracy could be improved by including the chiller control signals as model inputs, and training new networks or programming the control logic to estimate these controls. The addition of other submodels, such as to estimate flow rates, should also be investigated. The physical model mass flow rate calculations should be improved, and the partial free cooling mode should be added to the chiller model. In addition, the thermodynamics in no-flow states should be improved. Furthermore, mixing both physical and data-driven components in a joint grey-box model was outside the scope of this thesis. Using data-driven methods to train the variables in the physical model equations could also be investigated for probable increases in calibration accuracy.

The results of this thesis can be used by engineers and researchers to help choose which modeling method to pursue when creating new data center cooling system simulations, and to see what kind of accuracy the methods can achieve. Additionally, the models created for this thesis can be used to find optimal values for the Edge data center cooling system setpoints, or alternatively they can be used to predict the system response faster than real time, which allows for dynamic model predictive control of the setpoints.

References

[1] J. G. Koomey, "Growth in data center electricity use 2005 to 2010," tech. rep., Analytics Press, 2011. [Cited 6.12.2019] Available at: https://www.koomey.com/research.html.

[2] M. Avgerinou, P. Bertoldi, and L. Castellazzi, "Trends in data centre energy consumption under the European code of conduct for data centre energy efficiency," *Energies*, vol. 10, no. 10, 2017. ISSN 1996-1073.

[3] G. Zhabelova, M. Vesterlund, S. Eschmann, Y. Berezovskaya, V. Vyatkin, and D. Flieller, "A comprehensive model of data center: From CPU to cooling tower," *IEEE Access*, vol. 6, pp. 61254–61266, 2018. ISSN 2169-3536.

[4] Y. Berezovskaya, A. Mousavi, V. Vyatkin, and X. Zhang, "Smart distribution of IT load in energy efficient data centers with focus on cooling systems," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4907–4912, IEEE, 2018. ISSN 2577-1647.

[5] J. Gao, "Machine learning applications for data center optimization." Google AI Research Publication, 2014. [Cited 28.11.2019]. Available at: https://ai.google/research/pubs/pub42542.

[6] H. Shoukourian, T. Wilde, D. Labrenz, and A. Bode, "Using machine learning for data center cooling infrastructure efficiency prediction," in *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 954–963, IEEE, 2017. ISBN 978-1-5386-3408-0.

[7] J. Athavale, Y. Joshi, and M. Yoda, "Artificial neural network based prediction of temperature and flow profile in data centers," in *2018 17th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, pp. 871–880, IEEE, 2018. ISSN 2577-0799.

[8] A. Shehabi, S. J. Smith, D. A. Sartor, R. E. Brown, M. Herrlin, J. G. Koomey, E. R. Masanet, N. Horner, I. L. Azevedo, and W. Lintner, "United States data center energy usage report," tech. rep., Lawrence Berkeley National Laboratory, 2016. [Cited 6.12.2019]. Available at: https://eta.lbl.gov/publications/united-states-data-center-energy.

[9] M. Nakamura, "Learning and optimization models for energy efficient cooling control in data center," in *2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 395–400, IEEE, 2016. ISBN 978-4-907764-50-0.

[10] K. Nemati, A. Zabalegui, M. Bana, and M. J. Seymour, "Quantifying data center performance," in *2018 34th Thermal Measurement, Modeling Management Symposium (SEMI-THERM)*, pp. 141–147, IEEE, 2018. ISSN 2577-1000.

[11] E. J. Walsh, T. J. Breen, J. Punch, A. J. Shah, and C. E. Bash, "From chip to cooling tower data center modeling: Part II Influence of chip temperature control philosophy," in *2010 12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, IEEE, June 2010. ISSN 1087-9870.

[12] M. K. Patterson, "The effect of data center temperature on energy efficiency," in *2008 11th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pp. 1167–1174, IEEE, 2008. ISSN 1087-9870.

[13] T. J. Breen, E. J. Walsh, J. Punch, A. J. Shah, and C. E. Bash, "From chip to cooling tower data center modeling: Part I Influence of server inlet temperature and temperature rise across cabinet," in *2010 12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, IEEE, 2010. ISSN 1087-9870.

[14] B. Hadid, S. Lecoeuche, D. Gille, and C. Labarre, "Energy efficiency of data centers: A data-driven model-based approach," in *2016 IEEE International Energy Conference (ENERGYCON)*, IEEE, 2016. ISBN 978-1-4673-8463-6.

[15] L. Ljung and T. Glad, *Modeling of Dynamic Systems.* Prentice-Hall information and system sciences series, PTR Prentice Hall, 1994. 361 p. ISBN 9780135970973.

[16] L. Parolini, B. Sinopoli, B. H. Krogh, and Z. Wang, "A cyber–physical systems approach to data center modeling and control for energy efficiency," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 254–268, 2012. ISSN 0018-9219.

[17] C. Patel, C. Bash, R. Sharma, M. Beitelmal, and R. Friedrich, "Smart cooling of data centers," in *2003 International Electronic Packaging Technical Conference and Exhibition*, vol. 2 of *International Electronic Packaging Technical Conference and Exhibition*, pp. 129–137, ASME, 2003. ISBN 0-7918-3690-8.

[18] A. Mousavi, V. Vyatkin, Y. Berezovskaya, and X. Zhang, "Towards energy smart data centers: Simulation of server room cooling system," in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, IEEE, 2015. ISSN 1946-0740.

[19] G. Smpokos, M. A. Elshatshat, A. Lioumpas, and I. Iliopoulos, "On the energy consumption forecasting of data centers based on weather conditions: Remote sensing and machine learning approach," in *2018 11th International Symposium on Communication Systems, Networks Digital Signal Processing (CSNDSP)*, IEEE, 2018. ISBN 978-1-5386-1335-1.

[20] M. Marwah, C. Bash, R. Zhou, C. Felix, R. Shih, and T. Christian, "Estimating data center thermal correlation indices from historical data," in *13th InterSociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pp. 344–352, IEEE, 2012. ISSN 1087-9870.

[21] M. Baxendale, J. Athavale, S. Robertson, and Y. Joshi, “Data center temperature control using PI System and MATLAB,” in *2019 18th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, pp. 897–904, IEEE, 2019. ISSN 2577-0799.

[22] F. P. Incropera, D. P. DeWitt, T. L. Bergman, and A. S. Lavine, *Principles of Heat and Mass Transfer; International Student Version.* Wiley, 7th ed., 2013. 1048 p. ISBN 9780470646151.

[23] J. Bell, *Machine learning : hands-on for developers and technical professionals.* Wiley, 2015. 407 p. ISBN 9781118889398.

[24] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.* O’Reilly Media, Inc., 2nd ed., 2019. 856 p. ISBN 9781492032649.

[25] S. Sugiyama, *Human Behavior and Another Kind in Consciousness: Emerging Research and Opportunities: Emerging Research and Opportunities.* Advances in Human and Social Aspects of Technology (2328-1316), IGI Global, 2019. 102 p. ISBN 9781522582182.

[26] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, “LSTM: A search space odyssey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, p. 2222–2232, 2017. ISSN 2162-2388.

[27] J. Hale, “Deep learning framework power scores 2018,” 2018. [Cited 12.11.2019]. Available at: https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a.

[28] “TensorFlow: An end-to-end open source machine learning platform,” 2019. [Cited 12.11.2019]. Available at: https://www.tensorflow.org/.

[29] “Keras: The Python deep learning library,” 2019. [Cited 12.11.2019]. Available at: https://keras.io/.

[30] “Project Jupyter,” 2019. [Cited 28.11.2019]. Available at: https://jupyter.org/.

[31] Blue Box Group S.r.l., *Technical catalogue - Tetris W Rev FC/NG*, 2017.

[32] “The Engineering Toolbox,” 2004. [Cited 09.07.2019]. Available at: https://www.engineeringtoolbox.com/.

[33] Dell Inc., *Dell PowerEdge R430 Owner’s Manual*, 2018. [Cited 12.11.2019]. Available at: https://topics-cdn.dell.com/pdf/poweredge-r430_owners-manual_en-us.pdf.

[34] Blue Box Group S.r.l., *Controller manual - Service, Series: Zeta Rev Series, Beta Rev Series, Tetris 2 Series, Tetris Rev W LC, Tetris Rev W LC/HP*, 2018.

[35] L. Massaron and A. Boschetti, *Regression Analysis with Python*. Packt Publishing, 1st ed., 2016. 312 p. ISBN 9781785286315.

[36] Swedish Meteorological and Hydrological Institute (SMHI), “Ladda ner meteorologiska observationer - Luleå Flygplats (Swedish) [Download meteorological observations - Luleå Airport].” [Cited 12.11.2019]. Available at: https://www.smhi.se/data/meteorologi/ladda-ner-meteorologiska-observationer/#param=airtemperatureInstant,stations=all,stationid=162860.

Appendices

A Simulink simulation models

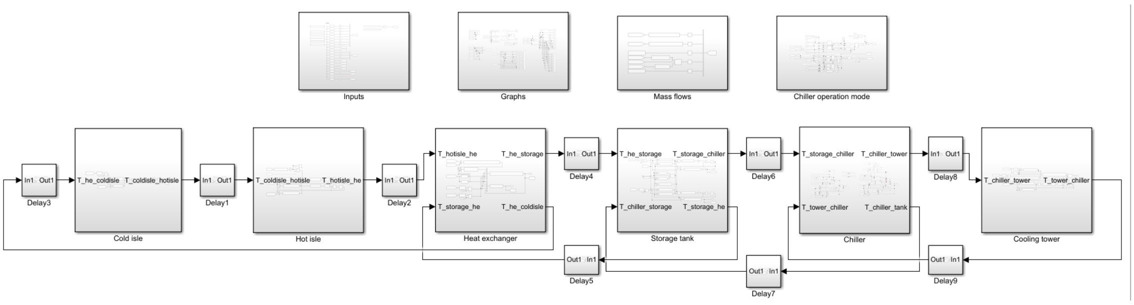
Inputs
Graphs
Mass flows
Chiller operation mode
In1 Out1
Delay3
T_he_coldisle T_coldisle_hotisle
Cold isle
In1 Out1
Delay1
T_coldisle_hotisle T_hotisle_he
Hot isle
In1 Out1
Delay2
T_hotisle_he T_he_storage
T_storage_he T_he_coldisle
Heat exchanger
In1 Out1
Delay4
T_he_storage T_storage_chiller
T_chiller_storage T_storage_he
Storage tank
In1 Out1
Delay6
T_storage_chiller T_chiller_tower
T_tower_chiller T_chiller_tank
Chiller
In1 Out1
Delay8
T_chiller_tower T_tower_chiller
Cooling tower
Out1 In1
Delay5
Out1 In1
Delay7
Out1 In1
Delay9

Figure A1: The Simulink simulation model.

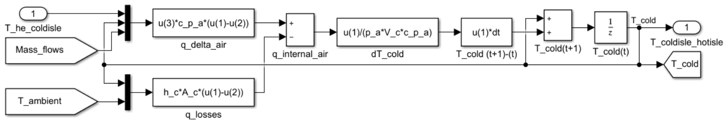

Figure A2: The cold aisle equations realized in Simulink.

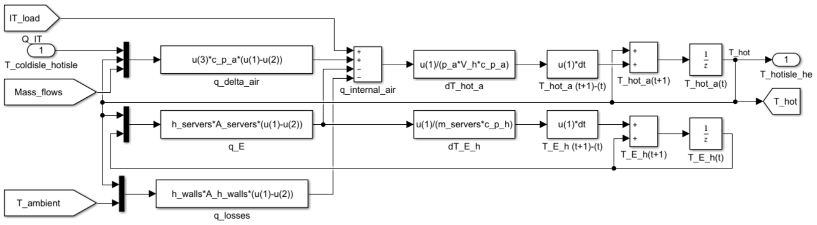

Figure A3: The hot aisle equations realized in Simulink.

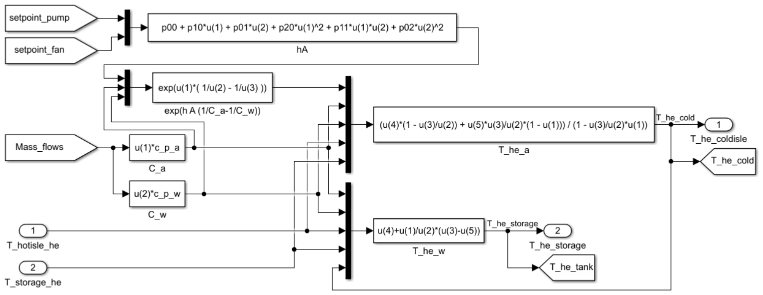
setpoint_pump
setpoint_fan
p00 + p10*u(1) + p01*u(2) + p20*u(1)^2 + p11*u(1)*u(2) + p02*u(2)^2
hA
exp(u(1)*(1/u(2) - 1/u(3)))
exp(h A (1/C_a-1/C_w))
Mass_flows
u(1)*c_p_a
C_a
u(2)*c_p_w
C_w
(u(4)*(1 - u(3)/u(2)) + u(5)*u(3)/u(2)*(1 - u(1))) / (1 - u(3)/u(2)*u(1))
T_he_a
T_he_cold
1
T_he_coldisle
T_he_cold
1
T_hotisle_he
2
T_storage_he
u(4)+u(1)/u(2)*(u(3)-u(5))
T_he_w
T_he_storage
2
T_he_storage
T_he_tank

Figure A4: The data denter heat exchanger equations realized in Simulink.

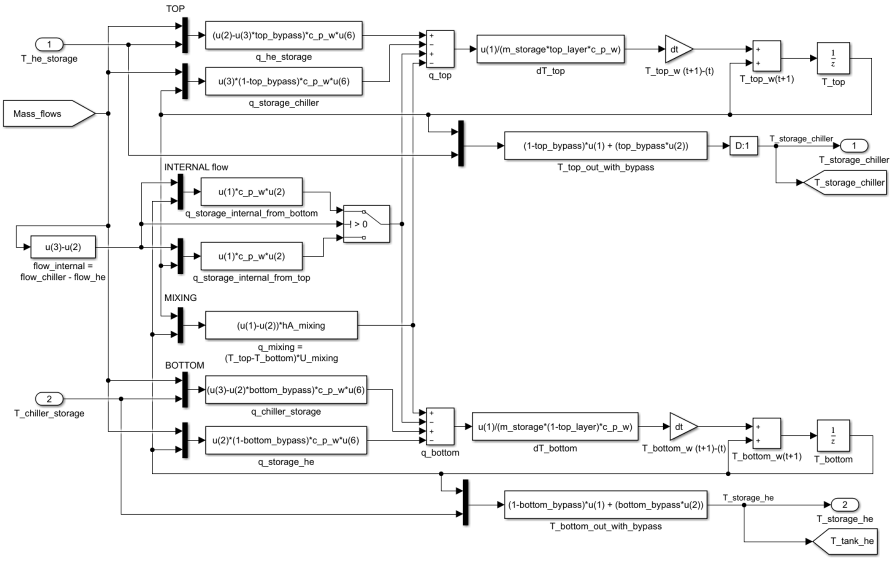

Figure A5: The storage tank equations realized in Simulink.

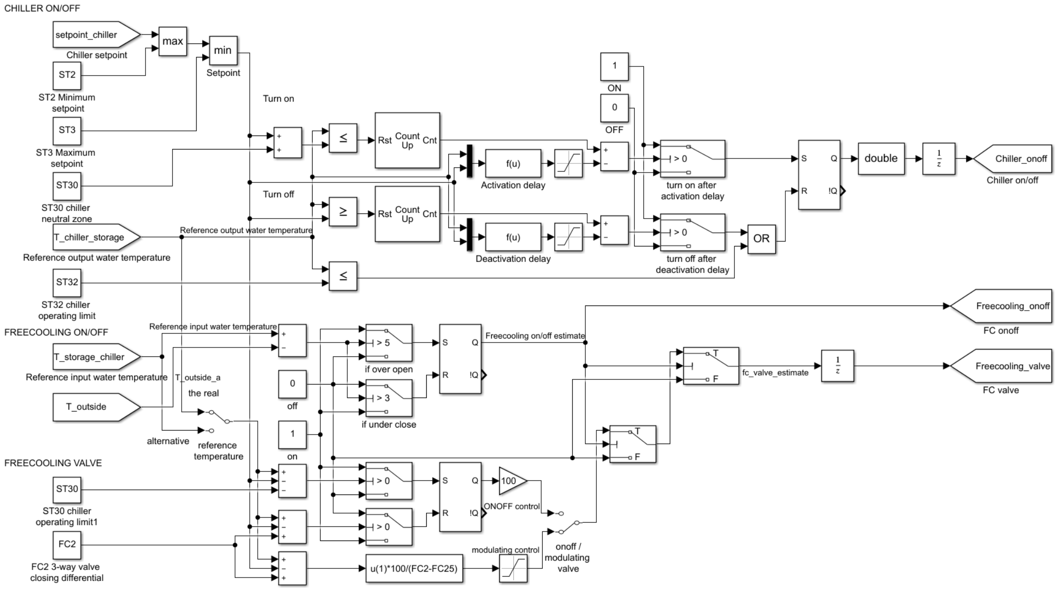

Figure A6: The chiller compressor On/Off, free cooling On/Off and the 3-way valve control logic signals realized in Simulink. Part 1 of 3 of the chiller model.

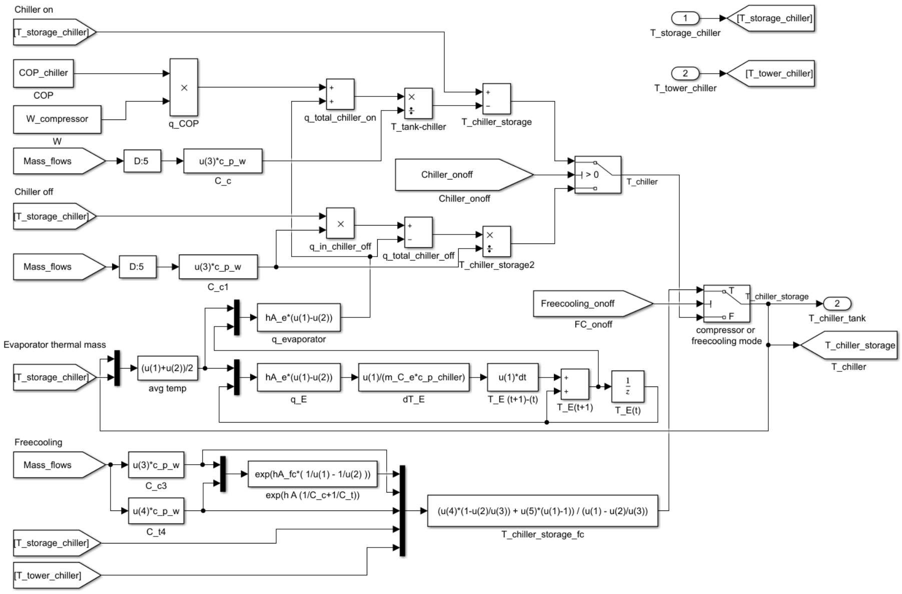

Figure A7: The chiller to storage tank output equations realized in Simulink. Part 2 of 3 of the chiller model.

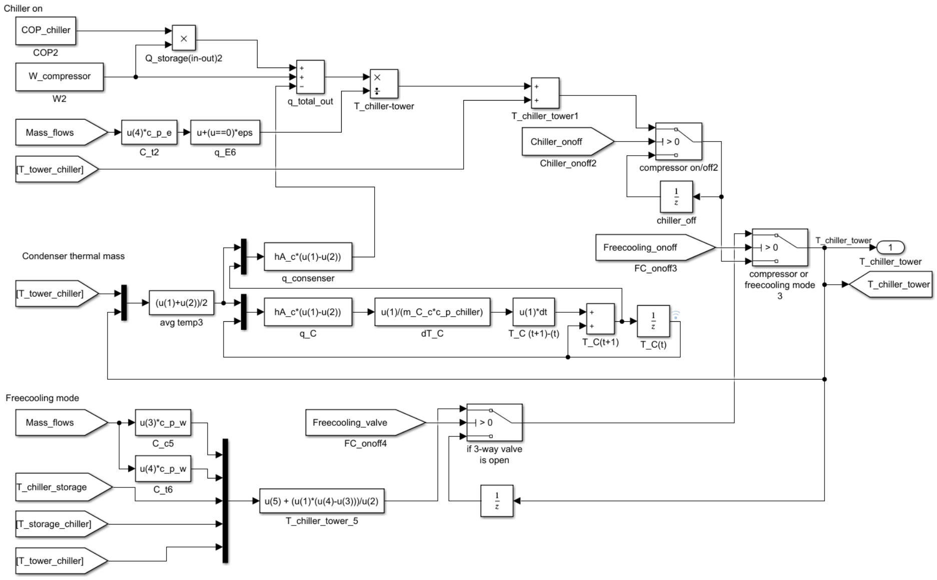

Figure A8: The chiller to cooling tower output equations realized in Simulink. Part 3 of 3 of the chiller model.

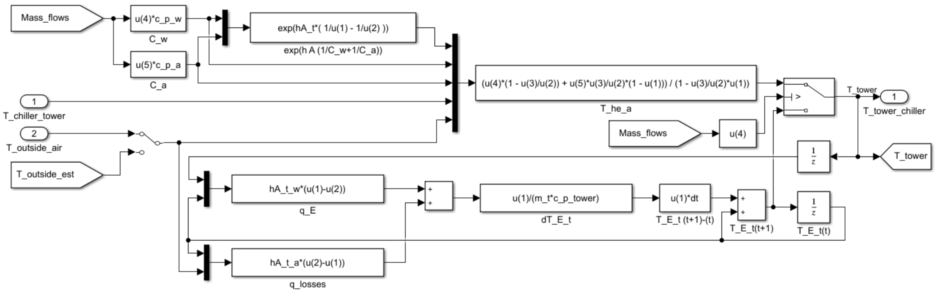

Figure A9: The cooling tower equations realized in Simulink.

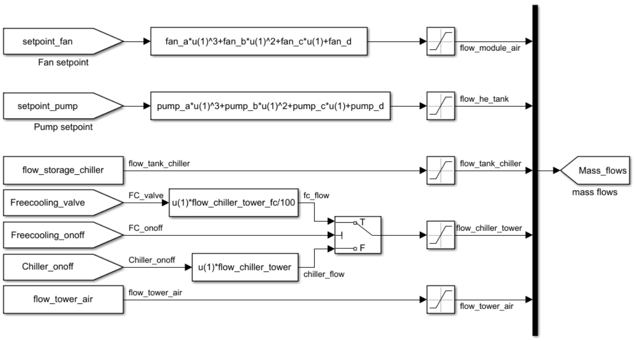

Figure A10: The mass flow equations realized in Simulink.

B Validation data inputs

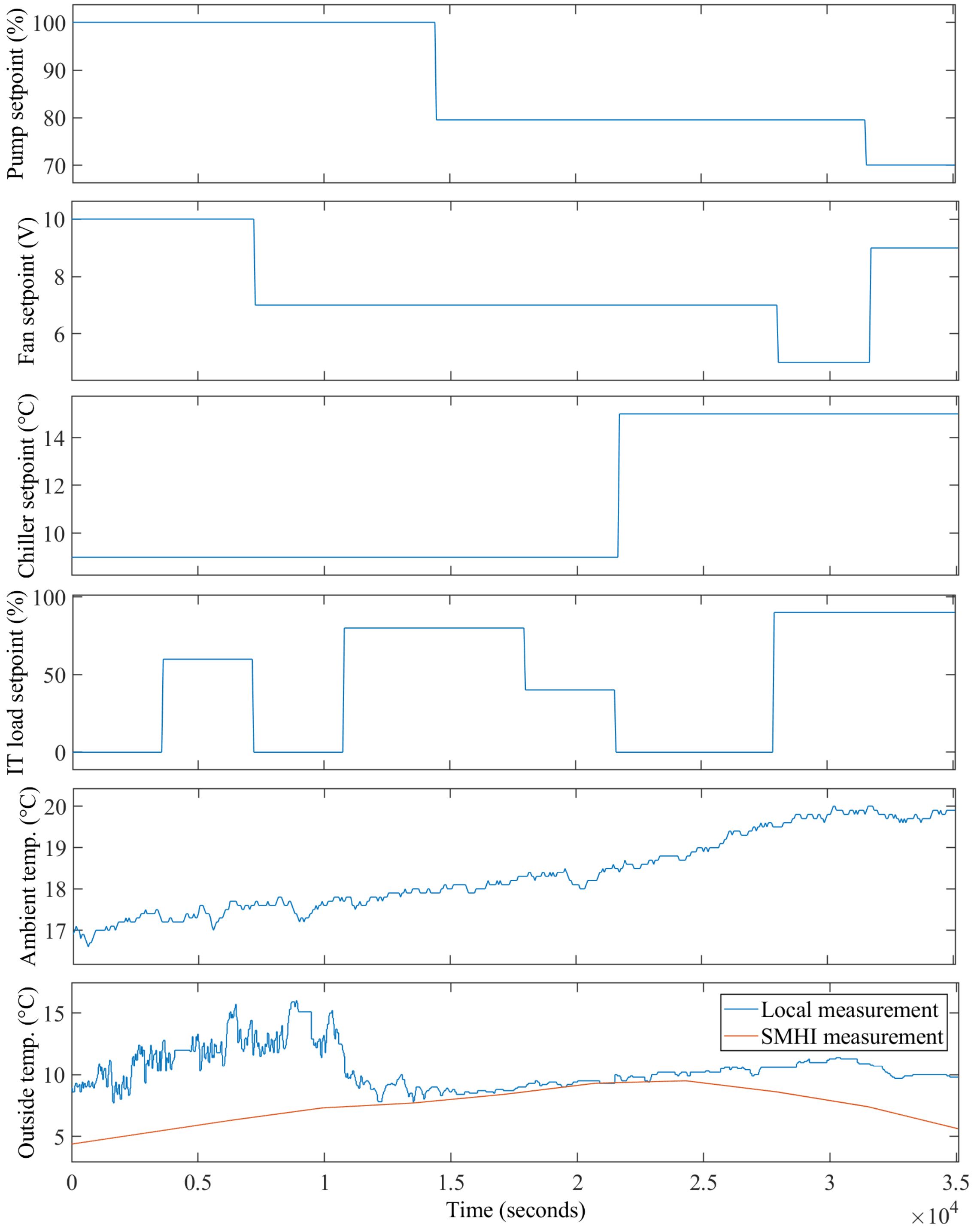

Figure B1: Model inputs during the validation period.

C Simulation error and error distribution graphs

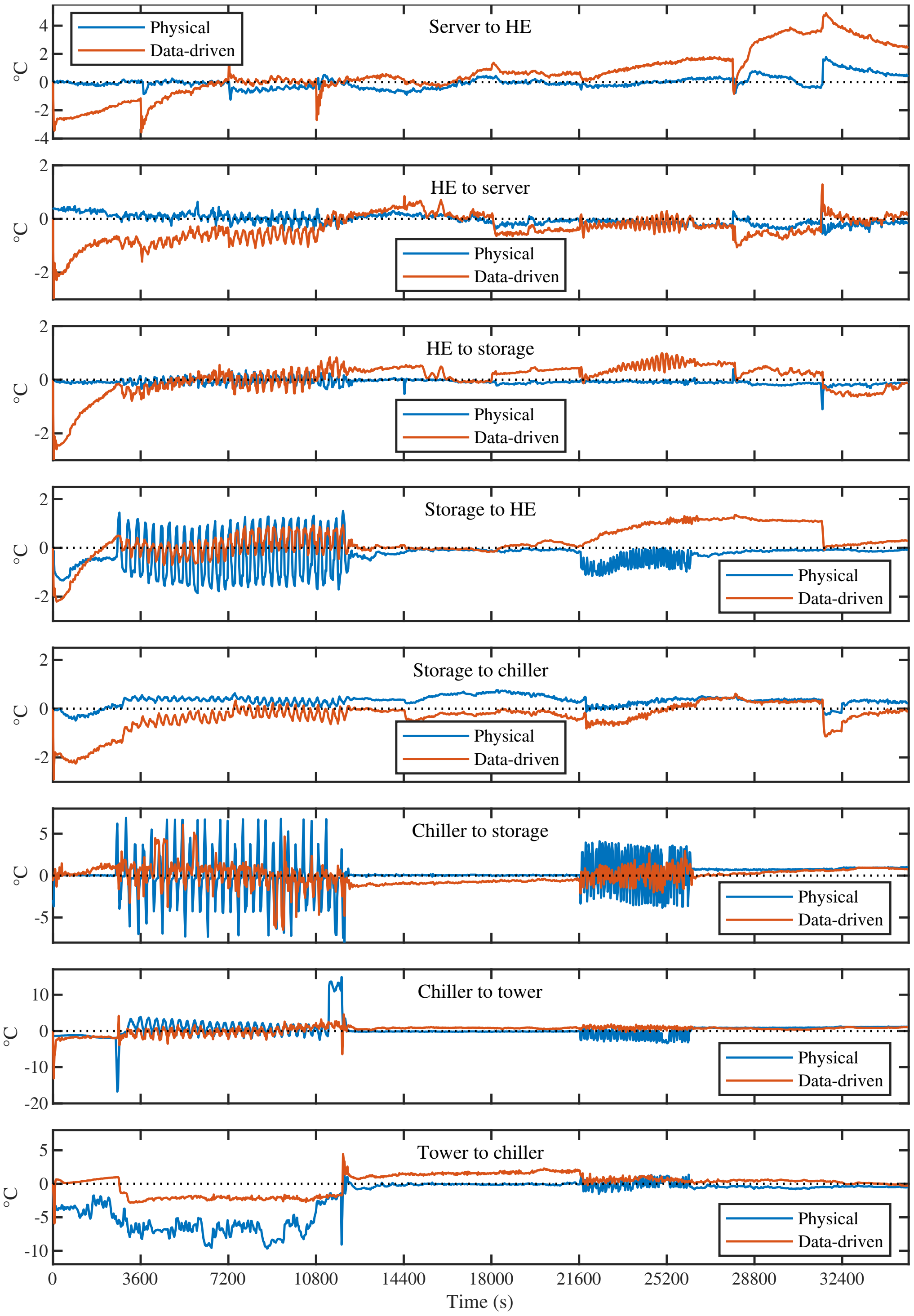

Figure C1: The prediction errors of the physical and data-driven model components when they receive accurate input data during validation.

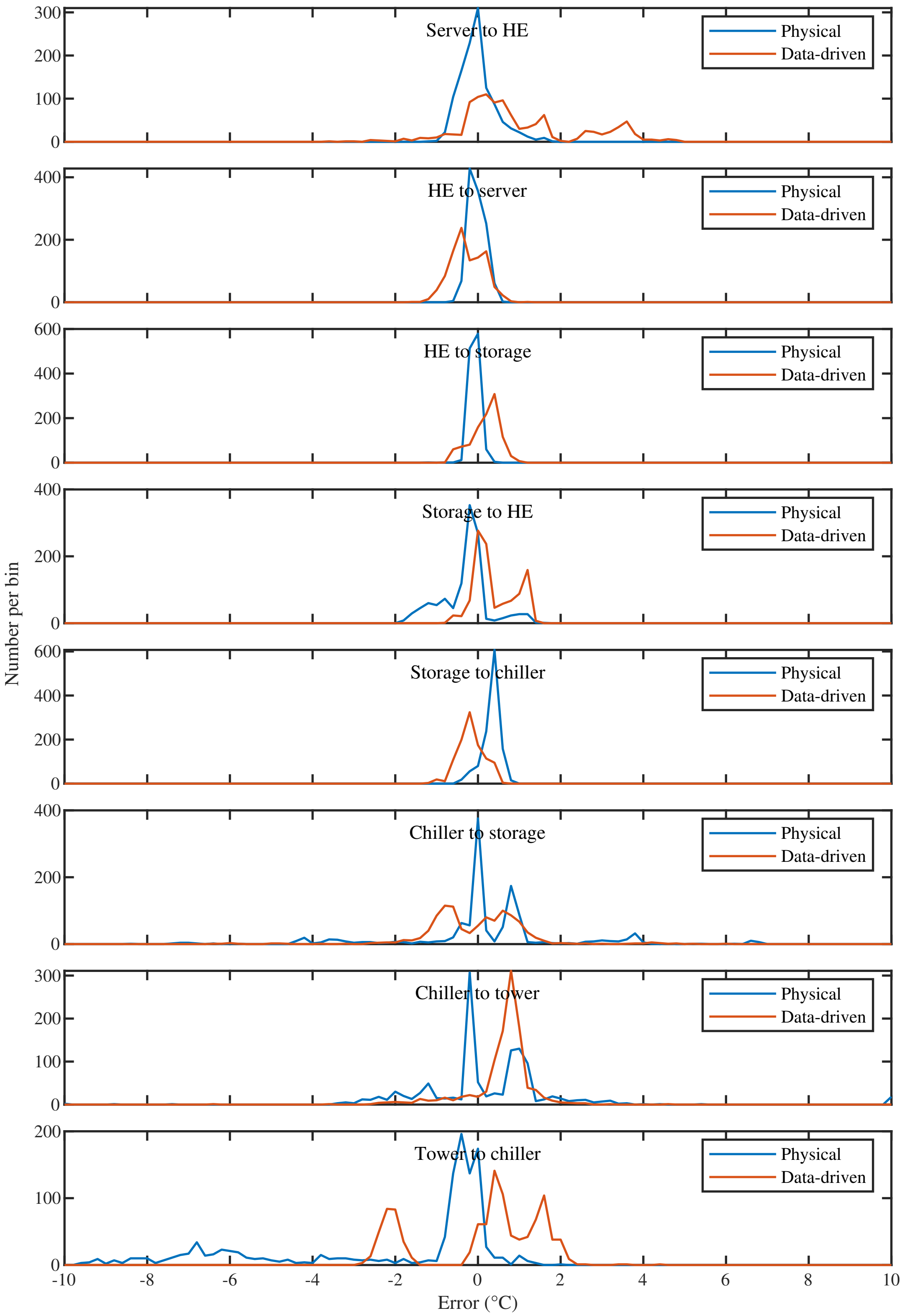

Figure C2: The error distribution of the physical and data-driven model components when they are validated with accurate input data. The values are gathered into 0.2 °C wide bins.

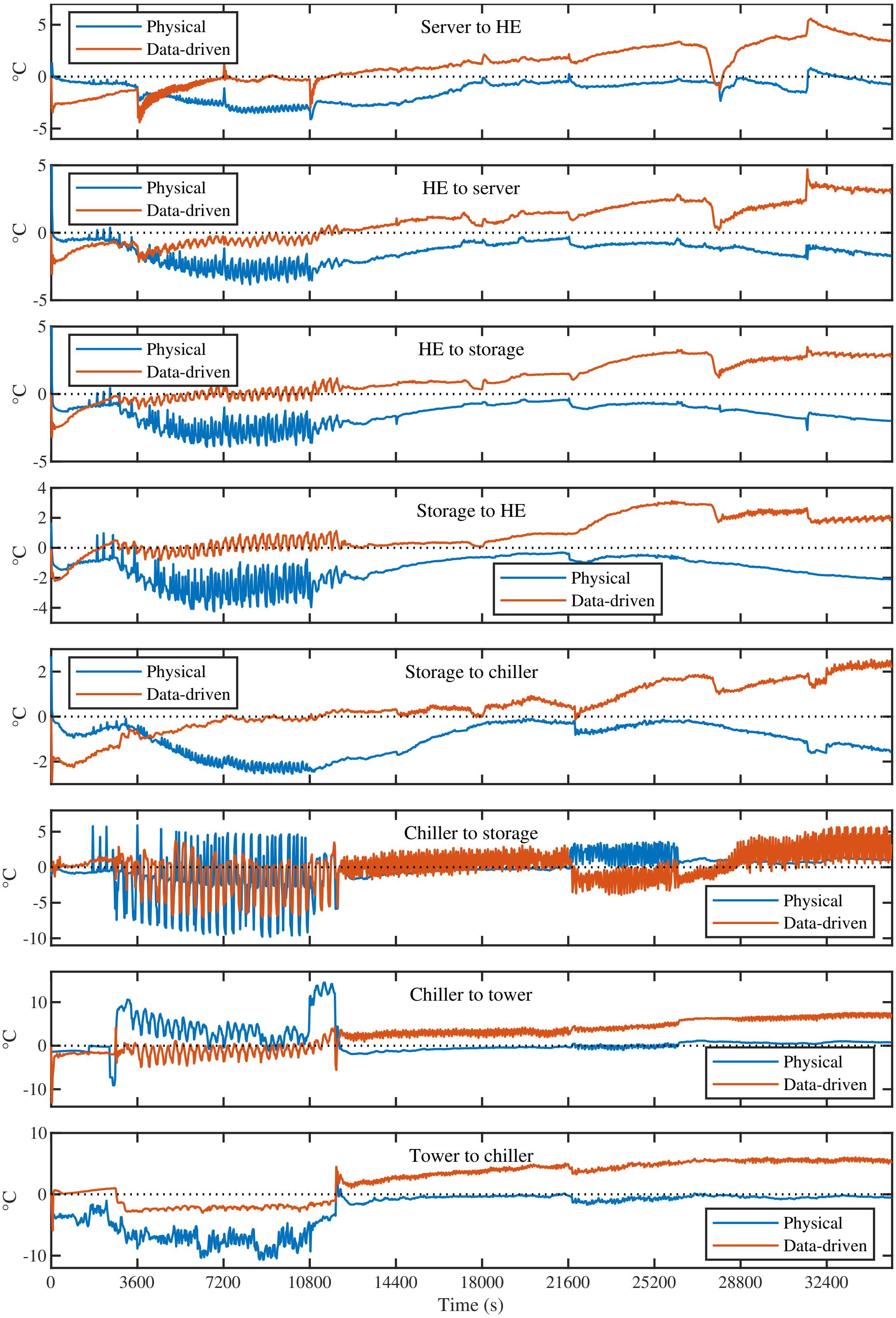

Figure C3: The prediction errors of the physical and data-driven model simulations.

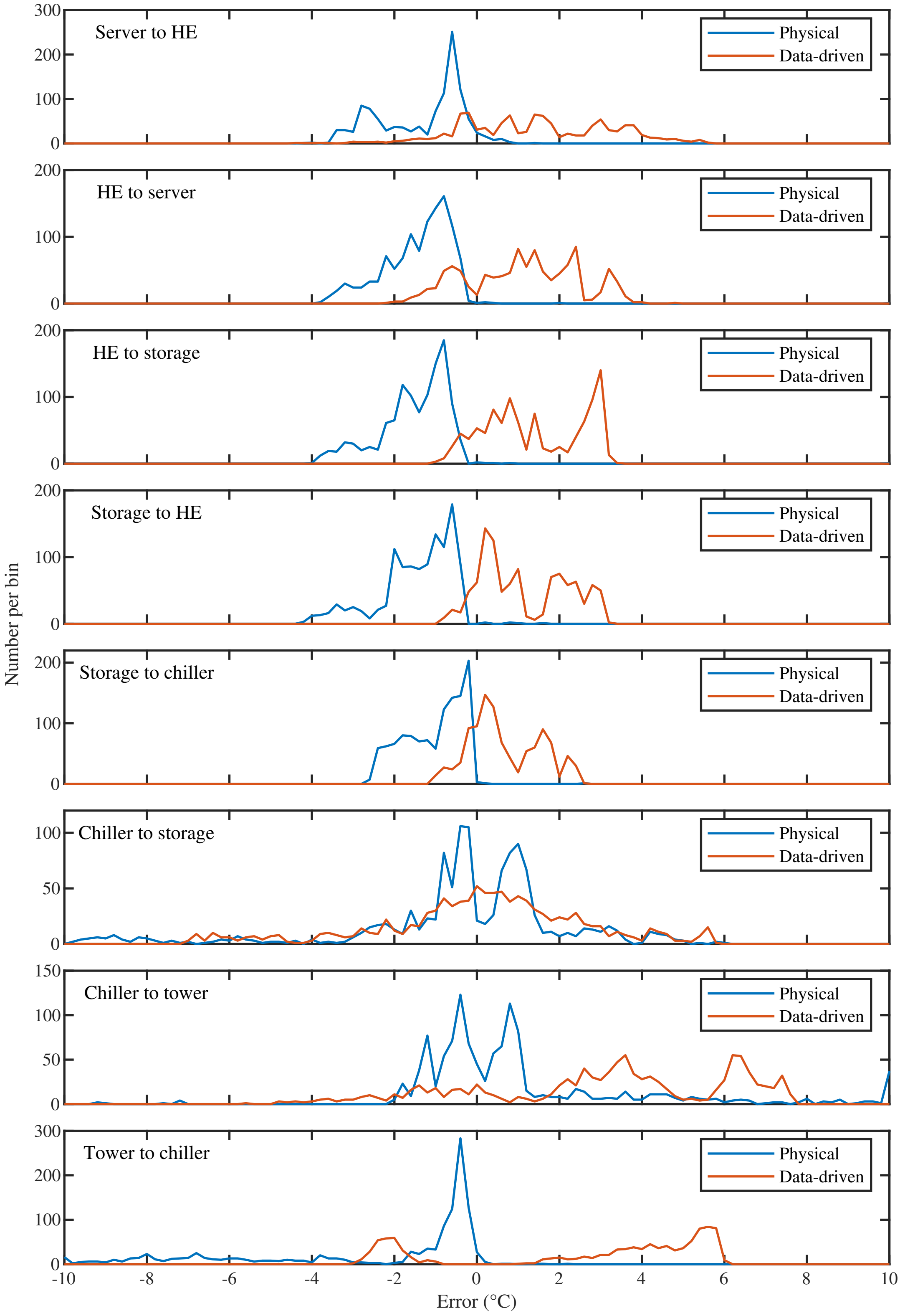

Figure C4: The distribution of the physical and data-driven model simulation errors. The values are gathered into 0.2 °C wide bins.