Reinforcement Learning for Building Heating via Mixing Loops

Overgaard, Anders

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

REINFORCEMENT LEARNING FOR BUILDING HEATING VIA MIXING LOOPS

BY
ANDERS OVERGAARD

DISSERTATION SUBMITTED 2019

AALBORG UNIVERSITY
DENMARK

Reinforcement Learning for Building Heating via Mixing Loops

Ph.D. Dissertation
Anders Overgaard

Dissertation submitted November, 2019

Dissertation submitted: November, 2019

PhD supervisor: Assoc. Prof. Jan Dimon Bendtsen
Aalborg University

Assistant PhD supervisors: Prof. Carsten Skovmose Kallesøe
Grundfos Holding A/S / Aalborg University

Ph.d. Brian Kongsgaard Nielsen
Grundfos Holding A/S

Ph.d. Hakon Børsting
Grundfos Holding A/S

PhD committee: Professor Jan Østergaard (chairman)
Aalborg University

Dr. Joaquin Blesa Izquierdo
Universitat Politècnica de Catalunya

Senior Manager Lars Finn Sloth Larsen
Danfoss

PhD Series: Technical Faculty of IT and Design, Aalborg University

Department: Department of Electronic Systems

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-544-4

Published by:
Aalborg University Press
Langagervej 2
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Rosendahls, 2019

Abstract

This thesis is concerned with designing data driven plug and play control for mixing loops used in heating for buildings. Designing energy efficient control for buildings has been a topic of interest for decades. This is mainly due the large operating cost and the often considerable potential for savings.

Classic industrial grade control of mixing loops for building heating is based on set point control. In the academic world control in heating for buildings, has been mostly focused on model predictive control. This approach however relies on models of the building and in the case of a mixing loop being sold as a standard prefabricated solution to different buildings a plug and play solution is to be desired. Therefore the attention has been on the data driven control methods within the field of Reinforcement Learning (Approximate Dynamic Programming). In its purest form no prior knowledge is needed for this controller to reach optimal control, but the training time becomes infeasible for an application such as a mixing loop.

With trends such as Internet of Things, big data and A.I. emerging a multitude of data becomes available for the control. Using this data in a self learning optimal control scheme can improve the control. However due to the curse of dimensionality the learning rate deteriorates as more sensor signals are used. A data driven algorithm based on partial mutual information has been developed to identify which sensor nodes should be used by the self learning controller for that specific building.

One of the major problems in mixing loop control is that long flow dependent transport delays occur. In this work a flow compensation is added to improve both the self learning control, but also taken into account in the initial selection of input variables.

Testing building control is difficult due to long testing time and difficulty of benchmarking. To combat this a hardware in the loop setup was designed to test the proposed control. Data was logged from an office building equipped with a multitude of sensors. This data was then used to form a load model. A hydraulic setup, including a mixing loop, was then controlled such that the mixing loop acted up against that load model. The control algorithms were embedded into microprocessors of the type that could

be applied in a mixing loop application to ensure feasibility of computation time. The proposed self learning control has been compared with an industrial grade controller. The results shows a promising improvement of performance within reasonable time leading to considerable savings.

The project has been carried out under the Danish Industrial PhD programme and has been financed by Grundfos Holding A/S together with the Danish Foundation of Innovation. Supervision of the project has been from Controls within Grundfos Holding A/S and the department of Automation and Control at Aalborg University.

Resumé

Denne afhandling omhandler design af et data dreven kontrolsystem til blandesløjfer anvendt til opvarmning af bygninger. I mange årtier har design af energi effektive kontrol systemer til bygninger vækket interesse. Det er hovedsageligt på grund af de store driftsomkostninger og det ofte store spare potentiale.

Klassisk har industriel kontrol af blandesløjfe til opvarmning af bygninger været baseret på setpunktskontrol. I den akademiske verden har fokus hovedsageligt været på model prædikativ kontrol. Denne metode afhænger dog af modeller af bygningen, hvilket i tilfælde af præfabrikeret blandesløjfer som sælges til mange forskellige bygninger, kan være svært. En selvindkørende løsning uden modelafhængighed er derfor at foretrække. Fokus er derfor på data drevne kontrol metoder indenfor feltet "Reinforcement Learning". I denne metode er der ikke behov for forhåndsviden omkring systemet for at opnå optimal styring. Træningstiden kan dog blive for lang for en applikation som blandesløjfer.

Med tendenser såsom "Internet of Things", "Big Data" og "A.I." i udvikling bliver der mere og mere data tilgængeligt til anvendelse i kontrol. At anvende disse data i et selvlærende kontrol system vil kunne forbedre ydeevne. Dog er den forhøjede dimensionalitet med til at forværre læringsraten for hvert ekstra sensorsignal der anvendes. En datadreven algoritme baseret på "Partial Mutual Information" er blevet udviklet til at identificere hvilket sensorer der med fordel kan anvendes i den selvlærende kontrol for en given bygning.

Et af de store problemer i blandesløjfekontrol er den lange flow afhængige transport forsinkelse der opstår i vandrørende. I dette værk vil en flow kompensering blive præsenteret for at forbedre både den selvlærende kontrol og den initiale udvælgelse af sensorer.

Det er svært at lave test på bygninger grundet lange testtider og et besværligt sammenligningsgrundlag. En "hardware in the loop" test opstilling til test af kontrolalgoritmen er udviklet for at overvinde dette problem. Der er blevet opsamlet data fra et væld af sensorer i en kontorbygning. Disse data er blevet anvendt til at forme en belastningsmodel. Et hydraulisk system, indehold-

ende en blandesløjfe, blev designet således at blandesløjfen spiller op imod denne belastningsmodel. Kontrolalgoritmerne blev implementeret på mikroprocessorer af sammenlignelig type med dem der findes i blandesløjfer for at anskueliggøre overholdelse af processeringstid. Den foreslåede selvlærende kontrol er sammenlignet med industrial klasseficeret kontrol algoritmer. Resultaterne viser en lovende forbedring indenfor en overkommelige tidshorisont hvilket fører til betydelige besparelser.

Projektet af lavet under det danske erhvervs PhD program og er blevet finanseret af Grundfos Holding A/S samt den danske innovationsfond. "Controls" afdelingen hos Grundfos samt afdeling for Automation og Kontrol på Aalborg Universitet har stået for vejledning under projektet.

Contents

Abstract iii

Resumé v

Preface xi

I Introduction and Summary 1

Introduction 3

1 Background and Motivation . . . 3
- 1.1 Value Driven Motivation . . . 3
- 1.2 Adding Customer Value . . . 4
- 1.3 Technology Enablers . . . 6
- 1.4 Business Driven Motivation . . . 7

2 Brief Introduction to building energy systems using mixing loops 9

3 State of the art and related work . . . 13
- 3.1 Industrial solutions . . . 13
- 3.2 HVAC control . . . 14
- 3.3 Reinforcement Learning . . . 15
- 3.4 Input Selection . . . 18
- 3.5 Delay compensation . . . 19

4 Objectives . . . 20

5 Contributions . . . 21

6 Outline of thesis . . . 22

Summary of work 25

7 Summary of Work . . . 25
- 7.1 Input variable selection . . . 26
- 7.2 Reinforcement learning . . . 35
- 7.3 Plug & play control scheme . . . 44
- 7.4 Results on performance of control scheme . . . 50

8 Conclusion and recommendations . . . 54
8.1 Conclusion . . . 54
8.2 Recommendations . . . 56
References . . . 57

II Papers **67**

A Input Selection for Return Temperature Estimation in Mixing Loops using Partial Mutual Information with Flow Variable Delay **69**
1 Introduction . . . 71
2 Preliminaries . . . 72
3 Application . . . 74
4 Methodology . . . 76
5 Results & discussion . . . 80
6 Conclusion . . . 84
References . . . 84

B Mixing Loop Control using Reinforcement Learning **87**
1 Introduction . . . 89
2 Preliminaries . . . 90
3 Building Heat Supply via Mixing Loop . . . 92
4 Q-learning with Gaussian Kernel Backup . . . 95
4.1 Choosing Reward . . . 95
4.2 Choosing States and Actions . . . 96
4.3 Gaussian Kernel Backup and pre-simulation . . . 97
5 Simulation Setup . . . 98
6 Results & Discussion . . . 100
7 Conclusion . . . 103
References . . . 103

C Reinforcement Learning for Mixing Loop Control with Flow Variable Eligibility Trace **107**
1 Introduction . . . 109
2 Building Heat Supply via Mixing Loop . . . 110
3 Preliminaries . . . 110
3.1 Basics . . . 111
3.2 Temporal Difference . . . 111
3.3 Eligibility Traces . . . 112
3.4 The method $Q(\sigma, \lambda)$ . . . 113
3.5 Radial Basis Function Approximation . . . 113
4 Proposed Method . . . 113
4.1 Flow Dependent Eligibility Trace . . . 114

4.2 Flow dependent $Q_\phi(\sigma, \lambda)$ . . . 114
5 Test . . . 115
5.1 Hydraulics . . . 116
5.2 Building Model . . . 117
5.3 Controllers . . . 118
5.4 Determining ϕ^* . . . 119
6 Results & Discussion . . . 120
7 Conclusion . . . 123
References . . . 123

D Reinforcement Learning for Building Heating via Mixing Loop with Data Driven Input Variable Selection 125
1 Introduction . . . 127
2 Building Heat Supply via Mixing Loop . . . 128
3 Preliminaries . . . 131
4 Method . . . 134
4.1 Reinforcement Learning Control of Mixing Loop . . . 134
4.2 State Selection . . . 138
4.3 Method overview . . . 140
5 Test Setup . . . 142
5.1 Building Model . . . 143
5.2 Hydraulics . . . 143
5.3 Controllers . . . 143
6 Results . . . 144
7 Conclusion . . . 150
References . . . 150

E Technical Report on instrumentation of office building for data collection 153
1 Introduction . . . 154
2 Office Building . . . 154
3 Data logger . . . 156
4 Signals . . . 157
5 Conclusion . . . 159

F Technical Report for experimental setup for mixing loop control 161
1 Introduction . . . 162
2 Hardware-in-the-loop . . . 162
3 Hardware . . . 163
4 Building Model . . . 165
5 Conclusion . . . 167

G Technical Report for simulation driven test **169**
1 Introduction 170
2 Dymola 170
3 Buildings Library 171
4 Models 173
5 Controls 174
6 Conclusion 175
References 175

Preface

This thesis is submitted as a collection of papers in partial fulfillment of the requirements for a Doctor of Philosophy at the Section of Automation and Control, Department of Electronic Systems, Aalborg University, Denmark. The work presented in this thesis has been supported by the Danish Ministry of Science, Technology and Innovation under the Industrial PhD programme. The work was carried out in the period spanning from June 2016 to Juli 2019 at Grundfos Holding A/S - Core Technology Controls and at the Section of Automation and Control, Aalborg University. I would like to thank my supervisors Prof. Jan Dimon Bendtsen, Prof. Carsten Skovmose Kallesøe, Ph.D. Brian Kongsgaard Nielsen and Ph.D. Hakon Børsting for their support and for the numerous of interesting discussions. I have been lucky to have this many supervisors all providing valuable guidance and inspiration within their different areas of expertise. I could not have hoped for a better support team. I would also like to thank all of my colleagues in the Controls department at Grundfos for discussion and general support within all related areas of pump control systems. A special thanks goes to Erik B. Sørensen for his help within building HVAC simulation. I also owe a thanks to colleagues outside Controls. Casper L. Mogensen for his expertise within various computer systems and without whom the test installations would never have functioned as well as they do. Jan Bæk Rohde for his crucial help with hardware and PCB design. During the course of this project I had the privilege of visiting the Department of Building Systems Engineering at the University of Colorado Boulder. I would like to express my gratitude to Prof. Gregor P. Henze for giving me this opportunity, and for some very insightful discussions during my stay. Also, I would like to thank the PhD students at the Department of of Building Systems Engineering, who helped making it an enjoyable and memorable stay. Finally, I would like to thank my family their love and support. Mathilde, my wife, has been incredible understanding and patient in times when work-life balance has been non existent.

Anders Overgaard
Aalborg University, November 12, 2019

Part I

Introduction and Summary

Introduction

1 Background and Motivation

This project was initiated by The Controls department at Grundfos Core Technology. Grundfos is a global company, employing approximately 19,000 people in 56 countries, with headquarters located in Denmark. The main business area for Grundfos is pumps and pump systems for heating and cooling. Grundfos is market leading for circulators worldwide. There are multiple motivation drivers for this project which are discussed here

1.1 Value Driven Motivation

The intrinsic motivation is driven by the Grundfos value of sustainability. This is achieved by providing energy efficient solutions that reduce the energy consumption leading to a reduction in CO_2 emissions. Increasing efficiency of Heating, Ventilation and Air condition (HVAC) systems has a large potential. Energy consumption is increasing in the world due to larger population and wealth. Improved efficiency has been used to combat the increase in energy consumption. In Fig. 1 the rise in energy consumption with and without the improved efficiency measures in the three largest energy consuming sectors throughout countries that participate in the International Energy Agency (IEA) can be seen.

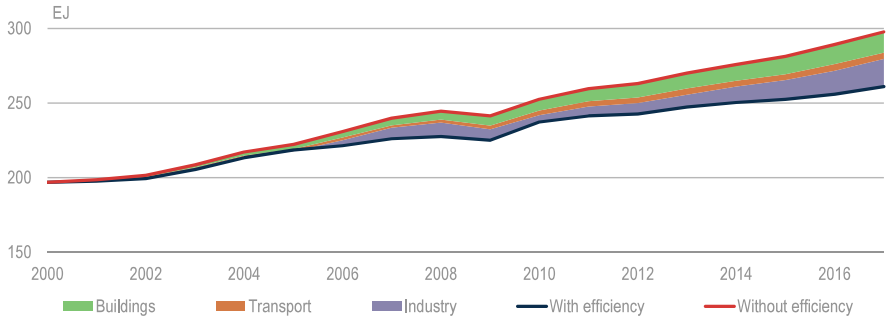

Fig. 1: Energy use in IEA countries and other major economies with and without energy savings from efficiency improvements, by sector, 2000-17. [46]

With increased focus on energy consumption, heating and cooling comes in focus, as these systems account for a large amount of the worldwide energy consumption. According to the United Nations Environment Programme the energy consumption for buildings corresponds to 40% of the overall energy consumption and 30% of the released green house gasses in the world [108]. According to [81] 34% of the building energy goes to HVAC in a typical office building as seen in Fig. 2.

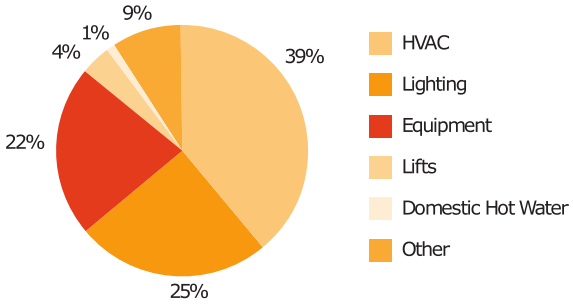

Fig. 2: Typical energy consumption breakdown in an office building [81]

This means that a large amount of energy is worldwide being used on HVAC, and thus the potential for energy savings is huge. This potential increases the motivation to do research within energy efficient control for HVAC systems.

1.2 Adding Customer Value

To achieve energy savings a proposed solution has to be sold and actually utilized in as many buildings as possible. By developing solutions that adds

customer value, the odds of selling the product and getting it adapted on large scale increases. Some ways of increasing customer value of HVAC systems are; Increasing comfort, minimizing energy consumption, reducing operational cost and limiting commissioning time.

Reducing operational cost is mainly achieved through energy savings. In [60] heating systems in UK was analysed and it was found that there was a potential for 20% less energy consumption through improved heat control. Tuning and proper set point selection in existing standard control structures can lead to reduced operational cost. Another way is changing control structure. Optimal control where the controls is often optimized towards high comfort and low operational cost can lead to considerable savings. In [9] up to 56% savings in operational cost was shown using model predictive optimal control. Here a model of the specific building had to be developed, contributing to a higher initial cost.

Minizing energy consumption does not only lead to a reduced operational cost, but can create value for the customer on its own. In many countries buildings code, policies or incentives are made for buildings with low energy consumption. In Denmark as an example the building code for residential buildings is demanding an increasingly lower energy consumption per area as seen in Fig. 3.

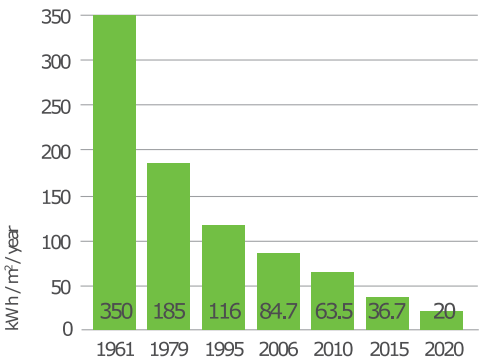

Fig. 3: Danish building codes from 1961 to present: Maximum allowed energy demand per year and m^2 heated floor space in a new 150 m^2 residential building. The limit is on the total amount of supplied energy for heating, ventilation, cooling and domestic hot water [26].

Commissioning is the phase where the building equipment is fit to the specific building. A large part of this phase involves preparing the HVAC equipment to meet the demands of the building. The challenge is that HVAC systems often come in the form of prefabricated systems that are sold to a variety of buildings. Fig. 4 illustrates how different buildings can be in terms of materials, window to wall ratio and purpose of the building. The function of the building might also require specific standards such as high fresh air

change rate in hospitals.

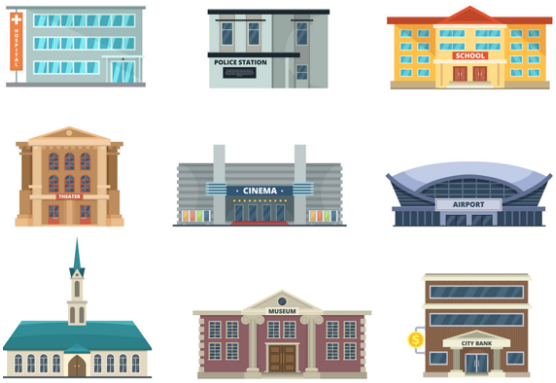

Fig. 4: Different buildings require different control. Differences in materials, window ratio, pipe layout, HVAC equipment, heat source, regulations for the type of building or other aspects change how the buildings is optimally controlled [96].

In HVAC systems that often means tuning the controllers or providing the right set points. This phase is time consuming and thereby expensive making the initial cost higher, but also has the effects of making the operational cost lower. In the case of one party constructing the building and another party buying and operating the scale often tips towards reducing initial cost and therefore limiting commission time. This might be the explanation why many systems are ill commissioned. In [70] 150 newly constructed buildings was tested with the conclusion that the average energy saving that could be achieved by doing a proper commissioning of the existing equipment was 18%. After doing the commissioning, improvement of thermal comfort was reported in 19% of the cases.

1.3 Technology Enablers

Many digital technology enablers are being developed within Intelligent systems and connected solutions these years. Areas such as Artificial Intelligence (A.I), Machine Learning, Model Predictive Optimal Control, Internet of Things (IoT) and Big Data is recieving a lot of focus in academia and in industry. An illustration of the increase in papers within artificial intelligence can be seen in Fig. 5.

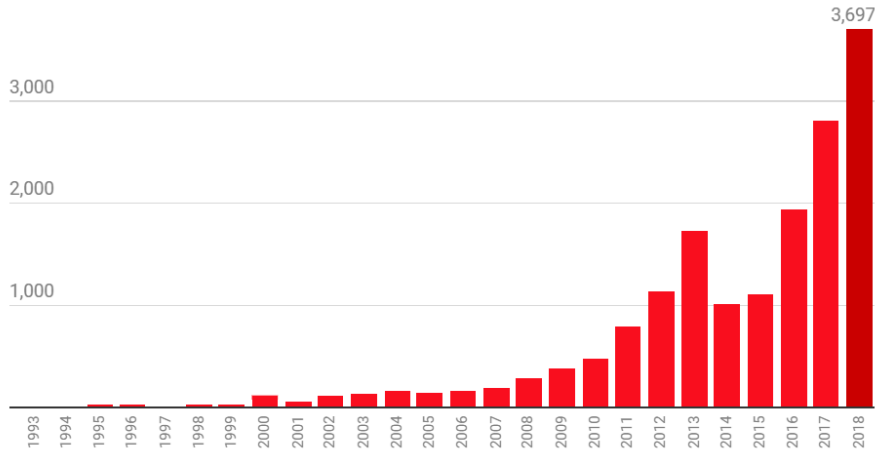

Fig. 5: Amount of papers in the Artificial Intelligence section of the open source database of scientific papers "arXiv" per year [71].

Advances within the before mentioned technology areas opens up for the possibility of developing self learning optimal control for systems. By measuring, learning and optimizing the HVAC system directly for a specific building the hope is to provide a control solution that delivers on all the presented customer values.

1.4 Business Driven Motivation

Another motivation for this research is staying ahead of the competition. Today companies, which have data as the main business, start entering the market for heating and cooling, with Google's acquisition of the thermostat company Nest as the most prominent example. Danfoss has taken up the competition with its smart valve system. Also small start-up companies are entering the market such as Building Robotics, AVOB, Kiltech Controls, and for example a Swedish company Nordiq has shown a good business case by dynamic control of the supply temperature to buildings. Other companies, as German Tado have combined whereabouts information from mobile phones with the temperature control, and thereby enabled huge savings, while still maintaining the high comfort for the users. It is believed that by combining the large system knowledge within Grundfos and the large knowledge about control and optimization at Aalborg University it is possible to build a system that can harvest the huge energy savings expected to be available in the building business without compromising user comfort. Such a solution will have great impact in the market. In this project we are concerned with the use of connection between digitalized products, from both Grundfos

and other companies, to optimize the control of heating systems. It can be stated thus: It is not a project that works on how to make products communicate, but instead this project focuses on what to communicate about. The technologies that will be developed in the project are in line with the main business area for Grundfos. As a step towards supplying system solutions Grundfos launched MIXIT. MIXIT is a mixing loop which is often used in hydronic based building heating and cooling for proper pressure and temperature control. Traditionally a mixing loop is put together from multiple components from different suppliers. In MIXIT a mixing loop system is offered in one unit, uniting a electronic 3 way regulating valve, balancing valve, one-way valve, sensors and controls.

Fig. 6: Grundfos mixing loop solution MIXIT

MIXIT is therefore the platform for the implementation of the control algorithm and methods developed in the project.

Grundfos provides and will in the future provide system solutions for many thermal hydronic applications. The learnings and the technology developed in this project within self training optimal controllers will hopefully have a spillover effect to these other applications.

2 Brief Introduction to building energy systems using mixing loops

This chapter is intended to provide a brief introduction to building energy systems using mixing loops. Building energy systems is the subject of heating and cooling in buildings. Many aspects and subsystems are the same. In this brief introduction a heating system is described, but many of the challenges and local control structures are the same.

Heating systems in large buildings are often designed with a main supply line delivering heat to a number of building sections. Mixing loops connect the main supply line to different sections of the building and to air handling units that ensure the air quality in the sections. The mixing loops connecting the building sections control the supply temperatures and pressure to ensure satisfying heat supply and room temperature control. A number of heat emitters, typically radiators, are connected to a mixing loop supplying heat to the rooms of the building. The structure with mixing loops controlling the supply to different sections of the buildings makes it possible to adjust the settings to the demands of the sections individually, e.g. north, south, east, and west zones of a building. Fig. 7 presents a sketch of such a system where the heat source is a heat exchanger connected to a district heating system.

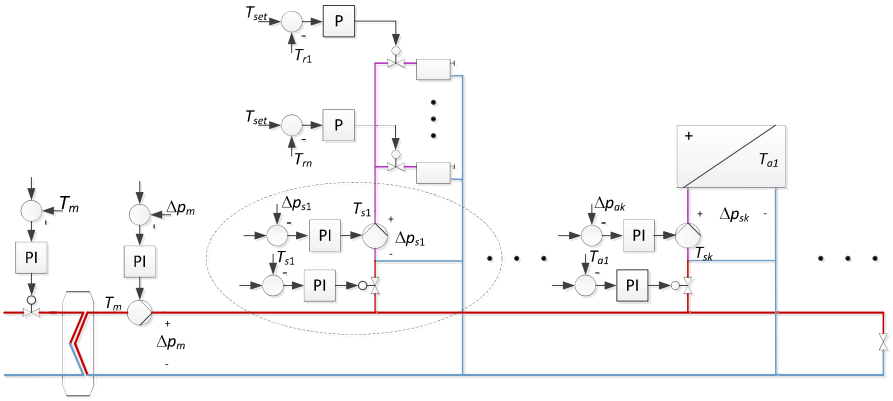

Fig. 7: Typical heating system for commercial buildings in its most simple form, with k rising mains and n local thermostats connected to each of the rising mains. A number of local P or PI controllers, controlling the temperatures and pressure in the network, forms the control architecture.

The mixing loop that is in focus in the project is marked with the dashed oval in Fig. 7. Large buildings typically contains a number of these dividing the building into heating zones. The second branch is an airhandling unit, which supply fresh air to the heating zones. To control the temperature

of the water leaving the mixing loop and the zone pressure, local setpoint controllers are used. A number of disconnected typically P or PI controllers takes care of the local control in such a systems. On paper, this makes the system easy to set into operation and enable set point control of important variables in the system. Unfortunately, often the controllers are not well tuned and it is hard to decide the optimal set points for the different parts of the system.

Pump Control

By controlling the speed of the pump, the zone pressure can be controlled to allow for the needed flow. On the other hand if the speed is set too high unnecessary power is consumed by added pressure losses. The affinity laws are often used to predict pump head, flow rate and power consumption as a function of changing between two rotational speeds [3].

$$\frac{\Delta p_2}{\Delta p_1} = \left(\frac{\omega_2}{\omega_1}\right)^2 \qquad \frac{q_2}{q_1} = \frac{\omega_2}{\omega_1} \qquad \frac{P_2}{P_1} = \left(\frac{\omega_2}{\omega_1}\right)^3 \tag{1}$$

Here the subscript 1 is before the change and 2 is after. Where ω is the rotation speed of the pump, Δ_p is the differential pressure (pump head), q is volumetric flow rate, and P is power consumption. From this it can be noticed the cubic relation between a change in pump speed and power consumption. Pump curves describe the differential pressure and the power consumption. These pump curves are often approximated by polynomials [116] [107].

$$\Delta p = a_{h0}\omega^2 + a_{h1}\omega q + a_{h2}q^2 \tag{2}$$

$$P = a_{p0}\omega^3 + a_{p1}\omega^2 q + a_{p2}\omega q^2 \tag{3}$$

Here $\mathbf{a_h}$ and $\mathbf{a_p}$ are pump specific constants, often measure and supplied by the manufacturer.

In building heating, thermostatic valves are often used to regulate the zones temperatures. By reducing the opening of a valve, the flow can be reduced and thereby the heat power being consumed at the terminal unit. The same reduction in flow could have been achieved by lowering the differential pressure of the pump. This would have saved pump power, but since the pump is supplying the whole zone this would affect other parts of the zone. This becomes even more complex when realising that the variable temperature, that the mixing loop offers, also affects the dissipated heat power at the terminal units. In industrial applications pumps are often controlled by constant speed, constant pressure or proportional pressure, see Fig. 8. The system curve caused by the system resistance giving the relation between flow and pressure. This curve is for constant resistance, meaning no changes in the form of valves opening and closing. The system curve at higher load is

where the valves are more opened than under design condition due to higher load.

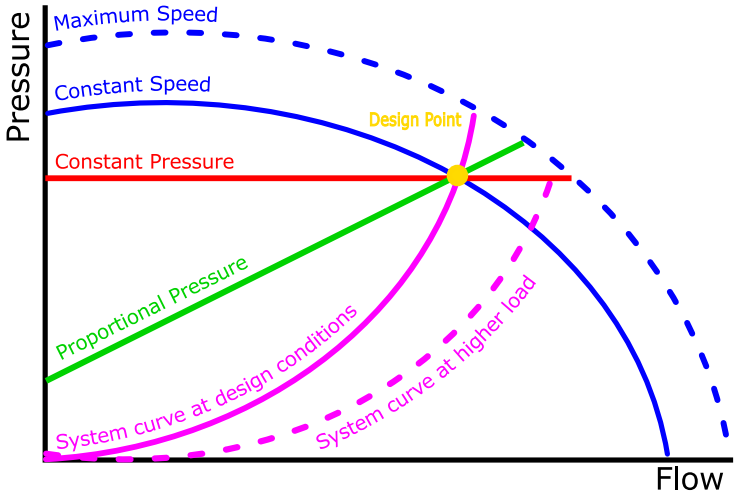

Fig. 8: Pump curves under different control schemes and at maximum speed. System curve at design condition and at higher load.

For all of these control schemes the design point (-12^oC outside temperature in Denmark) has to be met to ensure the needed pressure. Due to the shape of the system curve there is a potential to save pump energy by moving from constant speed to constant pressure. To go even further many industrial applications use proportional pressure control where the pressure is controlled proportional to the flow. Even with proportional pressure control where the pressure curve is fitted well, there is a potential here for the pump to match the system even better. Furthermore the system curve will change as a function of temperature since this influences the heat dissipation and therefore the opening of the thermostatic valves.

Temperature Control

In most heating systems the temperature reference on the supply temperature (T_m in Fig. 7)and the water temperature at the rising mains (Ts_1 in Fig. 7) is compensated with the outdoor temperature, such that; low outdoor temperature means higher supply temperature and vice versa. An example of this can be seen in Fig. 9, where four points define the compensation curve.

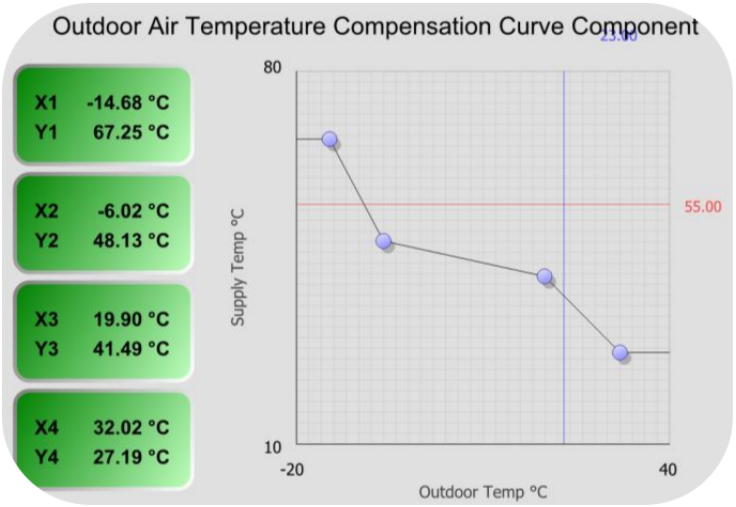

Fig. 9: Screenshot from a building management system of the outside temperature compensation.

However, due to the slow dynamics of the house and delays in the piping, this is not a very precise compensation. This is often refined by applying a first order filter approximation to this compensation, where the time constant should match that of the building dynamics. The outdoor temperature is not the only parameter that affects the heat of the rooms. In many buildings, the sun radiation has a larger impact. This is especially true, in houses and offices building with large window areas and good insulation. These effects are typically called disturbances. Other examples of disturbances that impacts the heat load are occupancy, electric appliances or wind speeds. When standard thermostats control the room temperatures, and the mixing loop temperature control does not compensate the disturbances very well, the result is variations in the room temperatures. In many buildings, the heat set point is lowered during night when the building is not in use, this is denoted night setback. This is often implemented as a constant offset on the temperature curve. Other implementations uses calendar modules for setback periods for buildings that has off time such as school vacation. The load of the heating system is in its extremes when the rooms are reheated after a setback, which results in high return temperatures from the radiators. This often leads to too high return temperature at the heat source, which again leads to poor efficiency at the heat source. To combat this challenge industrial controllers sometimes apply a return temperature limit which is a slow feedback loop which enables when the limit is reached and slowly brings down the supply temperature. Due to the long and often unknown flow variable delay this feedback loop has to be tuned very conservatively to avoid an unstable

control loop.

Here the classic control strategies used in building energy systems via mixing loops was introduced. Next the advances in control of building energy systems within academia and frontrunners in the industry is presented in a state of the art analysis. Afterwards the research objectives of this work are described.

3 State of the art and related work

The focus in this project is on self learning optimal control with data driven input selection and experimental validation for building heating using mixing loops. To provide an overview of current practices within the HVAC industry, as well as related work within the computer science and control engineering community, this chapter is divided into five parts. The first part covers industrial (commercial) available solutions. The next parts cover research advances done in academia within HVAC control, reinforcement learning, input selection and delay systems.

3.1 Industrial solutions

A mixing loop is typically build up from multiple components and local temperature and pressure controllers. In this project it is the set point control for the temperature of the water leaving the mixing loop and the pressure or pump speed that is of interest. Controlling mixing loop temperature and pressure is too specific an area to find content, but the same methods are also used in broader HVAC control, which is therefore the area of interest.

There is a gap between industrial applied controllers and the ones researched in academia. This is of cause the natural development process, but even research topics such as model predictive control that has been researched for many years has not been widely applied in the industry. Perhaps that is due to a focus on easy commissioning and reliability through simplicity.

In chapter 2 the basic control schemes for temperature and pressure was presented. Most commercial solution uses these methods, variations or parts hereof. In Danfoss ECL Comfort controller [25] weather compensation is implemented as a 6 point slope. A return temperature limitation is implemented as an offset on the weather compensation curve. It also has the feature of being able to autotune the parameters of the local PI controllers. Another example is the Sauter Flexotron controller [89] where an 8 point outside temperature compensation curve is used. A return temperature limit can be used that offsets the supply temperature by the amount of a chosen parameter. On

top of this a wind speed compensation can further be used on the supply temperature where a shift factor (^0C *per* m/s) has to be chosen. A night reduction functionality is further used where the temperature error along with a user specified heat capacity of the zone determines how early the reheating starts when coming out of night reduction. Pumps are controlled by constant pressure.

Some more advanced control solutions are commercial available, but not as wide spread. BuildingIQ [18] offers savings by using a model predictive control scheme. Machine learning techniques such as support vector machines (SVM) and K-means clustering is used to segment data and defer the contributions from different energy sources to heating and cooling. This solution is based on having engineers analysing 3-12 months of data and from this building the algorithm. This makes the initial price costly, but is earned through improved savings over time. NordIQ [80] has the SoftControl method where the supply temperature is control optimally. The savings stems from better control where room temperatures are kept from going higher than the setpoint at warm periods keeping comfort, while reducing the average temperature. NeuroBat [76] is a startup which offers a model predictive HVAC control solution. Here the model is a neural network which is trained and adapted to the specific building of operation. BrainBoxAI offers an A.I. solution build on deep learning. A deep neural network is trained from multiple data points such as outside temperature, sun/cloud positioning, fan speed, duct pressure, heater status, humidity levels and occupant density to predict the energy leak of the zones. A non-linear solving algorithm is then used to find the control actions leading to highest comfort at lowest cost. This solution is very computational heavy and is therefore run as a cloud solution [17].

3.2 HVAC control

The most active research topic within building HVAC control is optimal control in various forms.

In [93] a review on optimized control systems for building energy and comfort management is done. Research done in 121 works was analysed. In those works the most researched control schemes are model predictive control and multi-agent control. Another review of advanced control methods for building energy and comfort management can be read in [30].

In model predictive control a model of the system is used to compute an optimal trajectory of the system given selected control actions. Examples of building energy system control using model predictive control is [86] [33] [92] [97]. The performance of model predictive control depends on the accuracy of the

model. Different model approaches are used to capture the buildings and heating system dynamics. A review of thermal buildings models for control is in [10].

There are many solutions to overcome the modelling of the building from prior knowledge. A model was found for model predictive control using subspace identification in [20]. In [76] a neural network is trained from data and used as thermal model of the building. This model is then used to compute the optimal control trajectory. In [2] a review of artificial neural network (ANN) based model predictive control (MPC) is done. Backpropagation is used for training the neural networks. The challenge is finding the optimal control with a highly nonlinear function such as a neural network. Different nonlinear solvers has been proposed to solve this. In [5] Lagrangian dual method is used, while a multi-objective genetic algorithm was used in [37] and [84] utilised the method simplex.

Another popular advanced control method for HVAC is multi-agent control. A multi-agent system tries to accomplish objectives according to a set of rules and regulations. A multi-agent system consists of a set of agents that interact, communicate and coordinate themselves to achieve the established objectives. Self-coordination refers to the way in which the agents that make up the system cooperate to reach the objective of consuming less resources. Examples of multi-agent control in building energy systems are [27] [50] [88]. In [115] a review is done on HVAC multi-agent control.

Other topics of advanced building energy control include fuzzy logic [4] [29], scheduling [83] [8], plug and play control [100] and reinforcement learning, which is the focus of this work.

3.3 Reinforcement Learning

Reinforcement learning, also known as approximate dynamic programming, is a self learning optimal control scheme. For a basic introduction into reinforcement learning see section 7.2. Here some results from general developments within reinforcement learning is presented. This is followed by examples of HVAC implementations in the literature.

Research interest in reinforcement learning increased when deep Q-learning combining classic Q-learning with deep neural networks was presented in [72]. The most applied benchmark for reinforcement learning algorithms is Atari games, where soon after double-q learning was shown to improve performance by removing the bias of the max selection in in Q-learning by adding a second estimator [44]. Whereas Q-learning is value based, mean-

ing that a value function is used that maps an action to a value. The higher value the better action. Other reinforcement learning methods are policy based meaning that an optimal control policy is found directly without the use of a value function. In [73] actor-critic deep reinforcement learning was proposed to improve performance further. Here the actor is a policy based agent and the critic holds a state action value function that evaluates how well the action chosen by the actor performs. These advances are as stated benchmarked on Atari games. Another benchmark used to test algorithms has been a set of continous control tasks such as cart-pole balancing, cart-pole swingup, double inverted pendulum balancing and mountain car [117]. For this benchmark the methods truncated natural policy gradient [12], trust region policy optimization [90] and deep deterministic policy gradient [105] was shown to have good performance. Creating a wider benchmark with a large set of different environments to improve on generalization is still a work in progress in the research community, with [43] being an example of a community working on this.

Function approximation is an important area of reinforcement learning. A function approximation is often used to represent the value function. Different functionalities can be achieved by choosing the correct function approximation. For the methods which has a proof of convergence this is often build from theory that relies on linear or linear in the weights function approximation [99] [119]. When deciding basis function one might be chosen to achieve good performance in a specific domain due to a natural fit of structure. Examples of this are; using fourier basis [52], polynomial basis [106], coarse coding [104], tile coding [95], radial basis [53] and sigmoid basis [31]. A lot of research is also being done into nonlinear function approximators such as artificial neural networks. Here deep consists of a succession of multiple processing layers. Each layer consists in a non-linear transformation and the sequence of these transformations leads to learning different levels of abstraction [34] [82]. Different layer structures have been proposed with specific capabilities. Using recurrent networks it is possible for the network to exhibit temporal dynamic behavior by having an internal memory state [78]. To be able to have both long and short memories a variant of the recurrent neural network called long short term memory is proposed [40]. Convolutional neural networks uses convolution in layers to take advantage of hierarchical patterns in data. In this way more complex patterns can be assembled using smaller and simpler patterns. This makes it especially suitable for analysing visual imagery [74].

Eligibility Trace is a strong mechanism of reinforcement learning that provides a computational efficient way of implementing multi step behaviour. Instead of saving in memory all state transitions and rewards a trace vec-

tor is utilized. In [110] the true online TD(λ) was proposed using a dutch trace to improve on the popular TD(λ) method employing an accumulating eligibility trace. In [65] a gradient Q-learning method was proposed with guarantees of stability for off-policy learning using eligibility traces. The same guarentee of stability in off-policy training using eligibility traces has also been achieved via gradient-TD(λ) [102] and emphatic-TD(λ) [1]. In [77] the conditions required to learn efficiently and safely with eligibility traces from off-policy experience are provided and the novel method retrace(λ) is introduced. In [118] a unified approach for multi-step temporal-difference learning with eligibility traces in reinforcement learning is proposed.

Reinforcement Learning has also been implemented on HVAC systems. In [24] reinforcement learning was used for energy conservation and comfort in buildings. [19] used reinforcement learning to control HVAC and windows to provide natural ventilation at minimum operational cost. In [35] reinforcement learning is used to optimize occupant comfort and energy usage in HVAC systems. In [28] an adaptive Critic-Based Event-Triggered Control for HVAC System was proposed. An apprenticeship approach is used where the intial control is done by an LQR control scheme while the reinforcement learning agent trains in the background. After a while the control is shifted to the reinforcement learning controller. In [32] adaptive control for building energy management using reinforcement learning. Here a neural network was proposed for state approximation to help with the curse of dimensionality. Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage is shown in [61]. Here pre-simulation was used to help improve the initial performance of the controller. [109] proposes a learning agent for heat-pump thermostat control. Here the challenge is that in standard heat pump control the auxiliary electric heating is being used after setback due to the large energy demand. This decreases efficiency and is avoided by applying a control scheme that learns to slowly return from setback to avoid using the auxiliary electric heater. The same control problem is investigated in [87] where another learning agent for a heat-pump thermostat with a set-back strategy is proposed. In [114] deep reinforcement learning was investigated for for Building HVAC Control. [75] proposes an on-line Building Energy Optimization using Deep Reinforcement Learning, where the focus is on having the building perform in a smart grid. Both deep Q-learning and deep policy gradient approaches are investigated. In [21] load control of a residential area using convolutional neural networks for automatic state-time feature extraction in reinforcement learning is shown to reduce the electrical cost.

3.4 Input Selection

Input variable selection is the domain of selecting a set of inputs that leads to the optimal prediction model, with optimality being maximising some cost function. This closely relates to feature selection, but is more focused on the individual inputs rather than features generated by a set of inputs. Another close area which will not be analysed is dimension reduction where principle component analysis is often used in many forms such as singular value decomposition or eigenvalue decomposition. Here input variable selection is the focus, but there will be some overlap to feature selection and dimension reduction methods. In [68] a review of multiple input variable selection methods is done. The methods within input variable selection can generally be divided into three categories; wrapper, embedded or filter methods.

In wrapper methods an iterative approach is used where models are trained with the different inputs and evaluated. Here the result depends on how well the relation between variables that is sought is represented in the testing models. In [42] a single variable regression was used where models are created using each candidate variable and ranked according to performance. Another approach to single variable regression was proposed in [16] where a general regression neural network is utilised. In [15] a genetic optimisation algorithm was used in a general regression input variable selection scheme to find the optimal inputs.

In embedded methods the evaluation of the input variables takes place during the training of a single model. The input selection is done while the inputs are embedded in the model. [42] uses an embbedded input selection method where recursive feature elimination is done durnig an iterative model training. Here all inputs are used at the beginning and then removed based on rank magnitude of the weights impacted by the inputs. Multiple optimality driven methods adds penalty to model complexity onto the cost function to reduce weights. In [22] L^2 regularisation was used for learning kernels. [51] utilised L^1 regularisation for temporal difference learning. Pruning is another method where weights are analysed and those having little importance are removed [41]. Another approach has been proposed in [6] where a genetic algorithm was used for training the model using a decision variable to choose inputs. In [111] a genetic algorithm was used to choose the horizon and length of the input variables for prediction of air temperature.

Whereas wrapper and embedded methods relied on models, filter methods are model free. Filter methods is the category of methods where a measure of importance of the input candidates is used to choose the input set. Different input output relation measurements can be used to establish importance of

the input. The most well know relation measurement used for input variable selection is perhaps Pearsons correlation (linear correlation) [68]. Another popular relation measurement is mutual information. In [13] the mutual information criterion was used to find a subset of inputs for a neural network. In [48] a neural network was used to predict broiler weight as a function of various inputs such as indoor climate data. A subset of these input was selected using mutual information. Partial mutual information is an input variable selection method first proposed in [94]. Here an iterative procedure of choosing inputs via mutual information and then removing the added information is used to remove redundancy. Partial mutual information is later used with success in [69] and [59]. Mutual information is often implemented via kernel density estimation [39]. In [36] a shifted histogram implementation of mutual information was proposed.

Above a collection of data driven methods of selection inputs have been presented. Another approach is to use domain knowledge to select the inputs as in [11]. Here a procedure is suggested for identification of suitable models for the heat dynamics of a building. Grey-box models based on prior physical knowledge and data-driven modelling are applied. A hierarchy of models of increasing complexity is formulated based on prior physical knowledge and a forward selection strategy is suggested enabling the modeller to iteratively select suitable models of increasing complexity.

3.5 Delay compensation

In building HVAC pipe systems transport the energy to the different zones and back. Due to this flow dependent transport delays occur making the system highly non-linear. The problem that transport delays in systems incurs to control is well researched topic in control litterature - see [85] or the more recent [38] for an overview. The Smith predictor where a constant delay can be compensated for in a stable linear system is perhaps the most well known approach in classic control [98]. In [7] [57] [66] finite spectrum assignment is used to compensate in unstable systems using prediction of future states. Using feedback of predictions has also been extended to nonlinear systems with constant input delays [54] [56] and state delays [62] [63] [47]. Compensation for linear system with time varying input delay was proposed in [64] [79] [55]. In [14] a predictor feedback for nonlinear system with time varying input and state delay was proposed. Only few works consider delay compensation for reinforcement learning control schemes. In [49] a model-free H∞ control design for unknown linear discrete-time systems via Q-learning with LMI was proposed. While [119] suggested a nearly data-based optimal control for linear discrete model-free systems with delays via reinforcement learning. For estimating the time delay in non-Linear systems [67] presented a method

using average amount of mutual information.

4 Objectives

The overall scope of this project is to clarify potential savings and implications as a consequence of introducing a plug & play control scheme to the mixing loop control for building heating. To this end, four research objectives have been identified to form the basis for the work presented in this thesis.

In this work, a self learning control approach is chosen to effectively deal with the challenges that the problem poses in terms of variation of systems, delays and disturbances. To be able to learn the disturbances a lot of data points containing information of these need to be used. With the increased availability of sensor data the task becomes finding the correct data that gives information about the problem. Since some data holds value for some building, but not others, a data driven solution is desired. The first objective is therefore to develop input selection that determines the data points that holds information for the mixing loop application in a specific building.

The second objective is to demonstrate self learning optimal control on the mixing loop application. Self learning meaning that no extensive modelling of the system is needed, but is instead learned. Optimal in the sense that control actions are chosen to maximise some cost function.

Having demonstrated self learning optimal control and input selection on a mixing loop application the third objective is combining it. The objective is to utilize the input selection to decrease training time, while maintaining performance of the self learning optimal controller.

The background for this project is founded by the idea and intention of reducing operating cost while maintaining comfort of the building energy system using mixing loops. Verification of actually obtained cost reductions by the implementation of new control laws compared to industrial grade controllers is the fourth and last research objective. To reach this objective an the proposed control scheme should be implemented and tested on an experimental system.

Summarized, the four research objectives are:

Research Objective 1: Input selection methodology
Develop input selection method suitable for the application area of building heating via mixing loop.

Research Objective 2: Control design methodology
To demonstrate a self learning optimal control design methodology applicable to the mixing loop application

Research Objective 3: Plug & Play control
Develop a plug & play control scheme combining the methods derived in objective 1 and 2.

Research Objective 4: Experimental Verification of savings
To verify operational savings obtained through the implementation of the proposed control scheme on an experimental test setup.

Together, these research objectives make up the scope of this project, and all the work presented in this thesis can be related to one or more of these objectives.

5 Contributions

The main contributions in this project are divided into three categories which are closely related to the research objectives in Chapter 4.

Input Selection

- Proposal of a method for input selection for estimation of the return temperature. This method is based on partial mutual information due to the ability of dealing with non linear relations. The return temperature is one of the important states of the mixing loop application. It is however also one of the harder to estimate due to flow variable delay. The proposed method uses the flow measurement to find the variable delay that holds the most information between the inputs and the return temperature [Paper A].

Control Design

- Proposal of reinforcement learning based control strategy for mixing loops for building heating. To ensure convergence the state-action value function is a table implementation. A Gaussian kernel backup is proposed to increase training speed. This implementation showed the ability to increase performance compared to industrial standard controller that is well tuned. This work however also showed the necessity of increasing training speed further due to slow convergence caused by limited amount of data [Paper B].
- A generic physical model describing only basic relations is proposed for pre-training. This model has to add information in all system variations for it to add value and is therefore proposed as a basic generic model.

The pre-training was shown to provide a better initial performance of the control scheme [Paper B].

- Proposal of a reinforcement learning control scheme that improves on performance and training speed. Here function approximation is used to improve training speed. An method for varying the eligibility trace as a function of the flow is used to improve performance [Paper C].
- Proposal of a plug & play control scheme. A reinforcement learning agent [Paper C] is combined with input variable selection [Paper A] to create a solution that adapts to the specific building without manual tuning. Input selection is used to find the input variables that holds the most mutual information with respect to the return [D].

Validation

- Input selection performance for the method proposed in [Paper A]. This was done using experimental data from an office building supplied by mixing loop as described in [Appendix E].
- High fidelity simulation driven test of control scheme proposed in [Paper B]. Here the performance was compared against an industrial controller on three different buildings. The generated simulation models are described in [Appendix G]
- A hardware in the loop experimental setup is proposed to validate the methods in [Paper C] [Paper D]. The setup i described in [Appendix F]

6 Outline of thesis

This thesis is based on a collection of publications written throughout the course of the PhD project. Consequently, the thesis is divided into two parts:

The first part provides an introduction, summary of work and conclusions for this PhD project, while the second part contains all the related publications. More specifically, the structure of the remaining thesis is as follows:

Part I

Chapter 7 provides a summary of the work and results on input selection, control design and experimental validation. This chapter is intended to give a coherent overview of the problems and solutions that are considered in this work. Conclusions and recommendations, including suggestions for future work, are presented in Chapter 8.

Part II

This part contains the publications written during the PhD project. These are included in the following order:

[A] Anders Overgaard, Carsten Skovmose Kallesøe, Jan Dimon Bendtsen and Brian Kongsgaard Nielsen, "Input Selection for Return Temperature Estimation in Mixing Loops using Partial Mutual Information with Flow Variable Delay",
Proceedings of 2017 IEEE Conference on Control Technology and Applications (CCTA), pp. 1372–1377, 2017.

[B] Anders Overgaard, Carsten Skovmose Kallesøe, Jan Dimon Bendtsen and Brian Kongsgaard Nielsen, "Mixing Loop Control using Reinforcement Learning",
CLIMA 2019 REHVA HVAC World Congress. E3S Web of Conferences Vol. 111, 2019

[C] Anders Overgaard, Brian Kongsgaard Nielsen, Carsten Skovmose Kallesøe and Jan Dimon Bendtsen, "Reinforcement Learning for Mixing Loop Control with Flow Variable Eligibility Trace",
Proceedings of IEEE Conference on Control Technology and Applications 2019

[D] Anders Overgaard, Carsten Skovmose Kallesøe, Jan Dimon Bendtsen and Brian Kongsgaard Nielsen, "Reinforcement Learning for Building Heating via Mixing Loop with Data Driven Input Variable Selection",
This paper has been submitted to IEEE Transaction on Neural Networks and Learning Systems

The layouts of the above publications have been revised from their original form to fit the layout of this thesis. In addition to the above publications, unpublished technical reports outlining the work in relation to experimental validation has also been prepared as a part of this PhD thesis:

[E] Anders Overgaard, "Technical Report on instrumentation of office building for data collection", *Unpublished technical report 2017*

[F] Anders Overgaard, "Technical Report for experimental mixing loop control", *Unpublished technical report 2018*

[G] Anders Overgaard, "Technical Report for simulation driven test",
Unpublished technical report 2017

Introduction

7 Summary of Work

This chapter summarizes the contributions from this PhD project on the subjects of input variable selection, flow compensation, reinforcement learning and experimental validation. The final outcome is a plug and play control scheme that builds on all the proposed elements. Fig. 10 shows an illustration of the final plug and play control scheme.

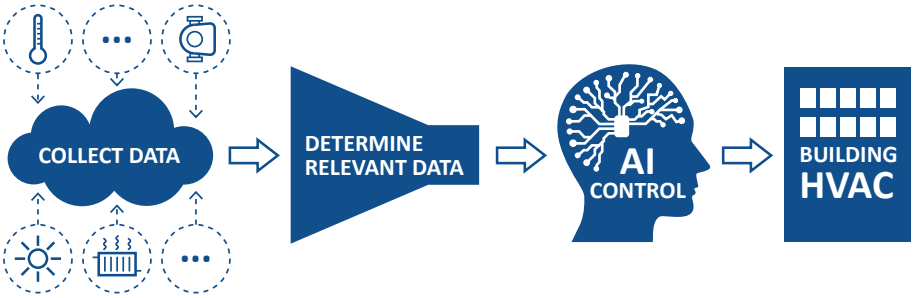

Fig. 10: Illustration of the plug and play control scheme

The chapter is based on papers A-D and appendix E-G but to add to the coherency of the summary, the contributions are presented in the order of topics rather than in chronological order. A quick reference list on test form when an appendix is mentioned is here given:

E Test setup where data is gathered from an office building located in Bjerringbro, Denmark.

F Hardware in the loop test setup where multiple mixing loop systems are installed and feeding into heat exchangers that are controlled to emulate a load model generated from the data gathered in the office building.

G Simulation environment where control of different building types and HVAC equipment can be tested.

7.1 Input variable selection

The purpose of input variable selection considered in this PhD project is to establish a method for deriving which sensors contribute with information to improve building control. By selecting a subset of inputs that holds the most information the dimension of the learning problem can be reduced. By reducing dimension a reinforcement learning control scheme can achieve a training time that is feasible for mixing loop applications.

The method proposed treats the input variable selection for reinforcement learning as a prediction task. In reinforcement learning, see chapter 7.2, the controlling agent seeks to maximise the return, where the return is the sum of rewards given over a time horizon. The reward function, outputting a scalar reward at every time step, is comparable to the cost function in other optimisation schemes. By analysing which input variables that has the strongest relation to the return a subset of these can be chosen to represent the agents state knowledge. In the later plug and play control scheme a filter approach will be used based on partial mutual information to determine relation between a set of inputs $\mathbf{x}$ and the return G. The input variable selection method proposed in this work will here be summarised.

Mutual Information

The basic relation criteria used in partial mutual information is mutual information that for two contionusly joint random variables $\mathcal{X}$ and $\mathcal{Y}$ is defined as [23].

$$I(\mathcal{X};\mathcal{Y}) = \int\int p(\mathcal{X},\mathcal{Y}) \log\left(\frac{p(\mathcal{X},\mathcal{Y})}{p(\mathcal{X})p(\mathcal{Y})}\right) d\mathcal{X}d\mathcal{Y}. \tag{4}$$

Here $p(\mathcal{X}), p(\mathcal{Y})$ are the marginal probability density functions with $p(\mathcal{X},\mathcal{Y})$ being the joint probability density function. When the log function is used in base 2, the unit of mutual information is "bits". In the case of independent variables $p(\mathcal{X},\mathcal{Y}) = p(\mathcal{X})p(\mathcal{Y})$ the fraction $\frac{p(\mathcal{X},\mathcal{Y})}{p(\mathcal{X})p(\mathcal{Y})}$ becomes 1 and the mutual information 0. The mutual information is a measurement of how much uncertainty about $\mathcal{Y}$ is removed by knowing $\mathcal{X}$. Mutual information also exists for the multivariate case $I(\mathcal{X}_1;\mathcal{X}_2;...;\mathcal{X}_n)$. In this work only second order relations will be examined due to computation limitations. This means that inputs that holds most information in third order will be neglected. Examples of this could be wind speed holding most information about the cooling of the building when coupled with the wind speed. Another example of a third order relation could be the solar radiation giving most information towards the free heat when the blinds position is added. In the second order relation

information can still be present for each of the examples on their own, but maybe in a lower amount.

Another important aspect to consider is that of time delay. The return is often a weighted summation of future rewards and is therefore a function of the time in the horizon $G(t, t+1, ..., t+n)$. For readability the n-step return at time t is written as G_{t+n}. An input might hold most information about G_{t+n} at time t, but other delays might hold more information. An example of this could be the outside temperature holding most information with respect to the return temperature at a delay due to slow dynamics of the building. In the proposed framework the same input variable can be used at multiple time delays. Inputs might hold information (that is not redundant) at multiple delays. This can as an example be due to higher order dynamics where a row of time instances represents the dynamic state. This means that the problem of finding the input variable that has highest mutual information with relation to return can be formulated as

$$\max_{j,k} \quad I\left(x^j_{t-d}; G_{t+n}\right), \tag{5}$$

where d is the delay and j is the index of the input in the full set of inputs.

To estimate the mutual information between two sampled variables a local gaussian approximation is used [39]

$$I(\mathcal{X};\mathcal{Y}) \approx \frac{1}{n_s}\sum_{i=1}^{n} \log\left(\frac{f(x_i, y_i)}{f(x_i)f(y_i)}\right), \tag{6}$$

where f is the estimated probability density from n_s samples of $\mathcal{X}$ and $\mathcal{Y}$. Kernel density estimation with parzen window is used for the probability density estimations. For the joint probability density this is [91]

$$\hat{f}(x,y) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}\left(\begin{bmatrix}x\\y\end{bmatrix} - \begin{bmatrix}x_i\\y_i\end{bmatrix}\right) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}\left(\mathbf{x}\right) \tag{7}$$

For $K_{\mathbf{H}}$ the Gaussian kernel is used on the form [59]

$$K_{\mathbf{H}}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^m |\mathbf{H}|}} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{H}^{-1}\mathbf{x}\right) \tag{8}$$

Here m is the dimension of $\mathbf{x}$ and $\mathbf{H}$ is the bandwidth matrix that controls orientation (off diagonal terms) and shape (diagonal terms). An often used bandwidth matrix for the bivariate case (only second order mutual information relations are investigated) is [112]

$$\mathbf{H} = h^2 \begin{bmatrix} S_x^2 & S_{xy} \\ S_{xy} & S_y^2 \end{bmatrix}. \tag{9}$$

Here S_x^2 and S_y^2 are sample variances. S_{xy} is the covariance between x and y, with h being the bandwidth parameter.

Fig. 11 shows mutual information, computed for the input variables as seen in table 1 with relation to return temperature in a mixing loop heating system as described in [Appendix E]. For an easier overview mutual information is only shown for the different inputs at the delay that yields the highest mutual information.

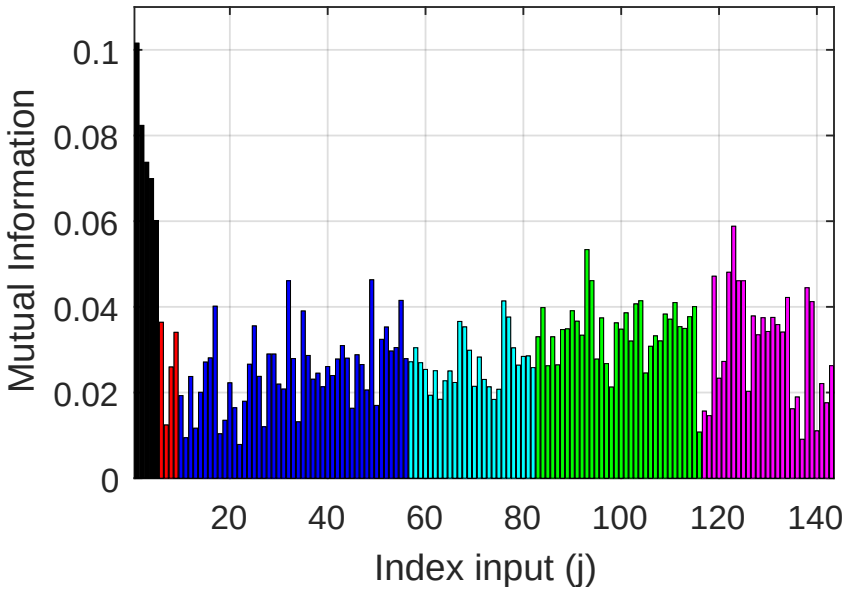

Fig. 11: Mutual Information as a function of input index. For each input only mutual information at the delays yielding highest mutual information is shown. See Table 1 for input indexes [Paper A].

Index (j)	Input Description	Name
1	Supply Temperature	(T_s)
2	Diff. pressure	(dp)
3	Primary Flow	(q_p)
4	Mixing Valve Opening Degree	(OD_{mv})
5	Heat Power Mixing Loop	(P_m)
6	Outside Temperature	(T_o)
7	Solar Radiation	(S_o)
8	Wind Direction	(Wd)
9	Wind Speed	(Ws)
10-13	Heat Power Ventilation Systems	(H_{v1-4})
14-18	Ventilation Air Temperature	(T_{v1-5})
19-51	Ventilation Ducts Opening Degrees	(OD_{d1-33})
52-56	Ventilation Fan Speeds	(VAV_{1-5})
57-82	CO_2 level in zones	(C_{1-26})
83-116	Zone Temperatures	(T_{s1-34})
117-143	Radiator Valve Opening Degrees	(OD_{r1-27})

Table 1: Inputs indexes [Paper A].

Partial Mutual Information

One way of choosing an input set would be to choose a subset of the inputs holding highest mutual information. This however might lead to input supplying redundant information. To handle this partial mutual information was suggested in [94]. In this method a search for the input providing the highest mutual information is done first. Afterwards the information provided by this input variable is removed from and a new search is done. In this way redundant information is removed. Another way of formulating this is finding the remaining mutual information between $\mathcal{X}$ and $\mathcal{Y}$ variables when $\mathcal{Z}$ is already given $I(\mathcal{X};\mathcal{Y}|\mathcal{Z})$. This is done iteratively until some stopping criteria. Partial mutual information uses estimators to remove mutual information from the inputs $\mathbf{x}$ and the output $\mathbf{y}$ and create the residuals $\mathbf{v}$ and u.

$$\begin{aligned} u_{t:T} &= y_{t:T} - E[y_{t:T}|z_{t:T}] \\ \mathbf{v_{t:T}} &= \mathbf{x}_{t:T} - E[\mathbf{x}_{t:T}|z_{t:T}]. \end{aligned} \tag{10}$$

The subscript $t:T$ means that it is a time series going from time t to time T.

The partial mutual information is then computed from the residuals as

$$I(\mathbf{x};y|z) = I(\mathbf{v};u) \tag{11}$$

A pseudo code algorithm for partial mutual information would be

repeat

Determine input variable z with highest mutual information as

$z \leftarrow \underset{j,d}{\text{argmax}} \quad I\left(x^j_{t-d}; y_t\right)$

Create the estimators $E[y|z]$ and $E[\mathbf{x}|z]$

$u \leftarrow y - E[y|z]$

$\mathbf{v} \leftarrow \mathbf{x} - E[\mathbf{x}|z]$

$y \leftarrow u$

$\mathbf{x} \leftarrow \mathbf{v}$

Move z to the subset of inputs $\mathbf{z}$

until *Stop Criteria*;

The stopping criteria used is based on the root mean square prediction error (RMSE) of the prediction model $E[y|\mathbf{z}]$. When the prediction error improves less than a set tolerance *tol* the algorithm is stopped

$$tol > \frac{RMSE_{i-1} - RMSE_i}{RMSE_{i-1}} \tag{12}$$

where i is the algorithm iteration counter.

In this work a generalized regression neural network is used to generate the estimators. These networks uses the radial basis function with the feature point center x_i and width σ.

$$\phi_i = \exp\left(-\frac{(\mathbf{x_i} - \mathbf{x})^T(\mathbf{x_i} - \mathbf{x})}{2\sigma^2}\right) \tag{13}$$

In the case of a single layer network with d weights (w) this means that $E[y|x]$ would be modelled by

$$\hat{y} = \sum_{i=1}^{d} w_i \phi_i(\mathbf{x}), \tag{14}$$

Flow Compensation

A flow variable delay compensation scheme will now be presented. The purpose of this delay compensation is twofold. First it is applied to the input variable selection and secondly it is used to determine the horizon of the return as seen in chapter 7.2.

In building heating thermal delays occurs due to long piping networks. An example of this is the propagation of the supply water throughout the system before ending up at the return. While most variable relations in the mixing loop application is subject to this longer delay, the pressure relations act on a much faster time scale.

Due to variable transport delays the input variable delays that holds most information will change with flow. In a simplified system containing only a single pipe between the supply and return, no heat losses, no mixing of the fluid and constant flow the relation between supply and return temperature can be described by a constant delay d

$$T_r(t) = T_s(t - d) \tag{15}$$

In this simplified system an estimator of the delay d based on mutual information could be

$$\underset{d}{\text{argmax}} \quad I\left(T_s(t-d); T_r(t)\right), \tag{16}$$

Flow does however change over time in the system. The first assumption that will be applied is that the flow is quasi static meaning that it stays constant in the time frame of the delays. In Fig. 12 flow data from the experimental setup installed in the office building in Bjerringbro [Appendix E] can be seen. Here the flow changes slowly enough to be considered quasi static within the thermal delay between supply and return.

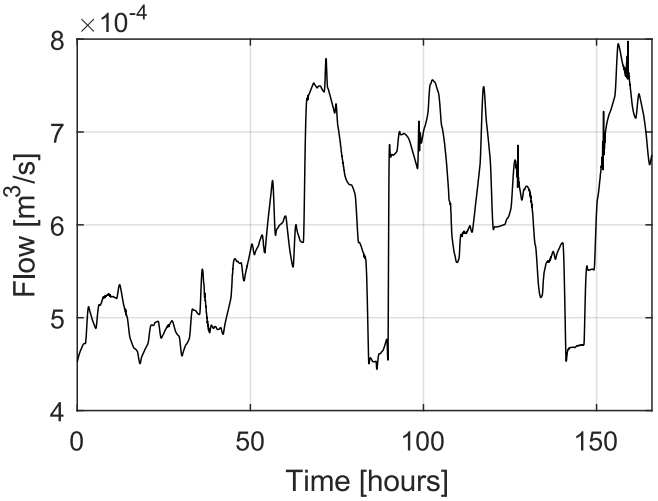

Fig. 12: Flow time series from mixing loop installed in office building [Appendix E]. Flow changes over whole period, but is quasi static within the delay times between supply and return [Paper A].

With the assumption of quasi static flow, while still maintaining the rest of

the simplifications from before the flow variable delay can now be estimated as

$$\underset{V}{\text{argmax}} \quad I\left(T_s\left(t-\frac{V}{q}\right);T_r(t)\right). \tag{17}$$

Here measurements of the flow (q) is used to find the volume of the pipe (V). When the pipe volume has been established the flow variable delay can be calculated using measurements of the flow. This is however still only for a single pipe system. For a system with multiple pipes routes the relation is extended to

$$T_r(t) = h\left(\mathbf{T}_s, \mathbf{q}\right), \tag{18}$$

where

$$\begin{aligned} \mathbf{T}_s &= \left[T_s\left(t-\frac{V_1}{q_1}\right), ..., T_s\left(t-\frac{V_n}{q_n}\right)\right] \\ \mathbf{q} &= [q_1, ..., q_n] \end{aligned} \tag{19}$$

Usually there will only be measurements of the total flow available and not how the flow divides into specific pipe routes. A flow ratio β is introduced to relate the specific pipe flows to the total flow.

$$\begin{aligned} \sum_{n=1}^{p} \beta_n &= 1 \\ q_n &= \beta_n q \end{aligned} \tag{20}$$

A lumped parameter v, here named the lumped volume, is defined as

$$v_n = \frac{V_n}{\beta_n}, \tag{21}$$

The approximation used here is that the flow ratios β stay constant. The preciseness of this assumption is discussed later. When inserted into (19) this becomes

$$\mathbf{T}_s = \left[T_s\left(t-\frac{v_1}{q}\right), ..., T_s\left(t-\frac{v_n}{q}\right)\right] \tag{22}$$

To demonstrate how partial mutual information, as described in chapter 7.1, is used to find the lumped volume, a simulation of the system in (22) was done. Three pipe routes with the lumped volumes ($v_1 = 0.02$, $v_2 = 0.04$, $v_3 = 0.12$) was simulated with the input of total flow being the time series seen in Fig. 12. The supply temperature was changed multiple times to induce a change of return temperature as seen in Fig. 13. Partial mutual information where the total flow is measured was applied to find the lumped volumes that holds most mutual information. The resulting mutual information as a function of v at the different partial iterations can be seen in Fig. 14. Here the lumped volumes was successfully found at each itera-

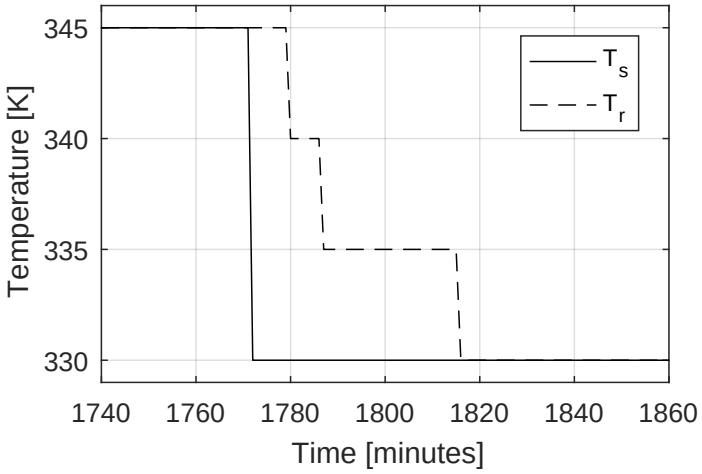

Fig. 13: One edge of the pulses given in the simulation. The delays between supply and return are a function of the flow given at that time [Paper A].

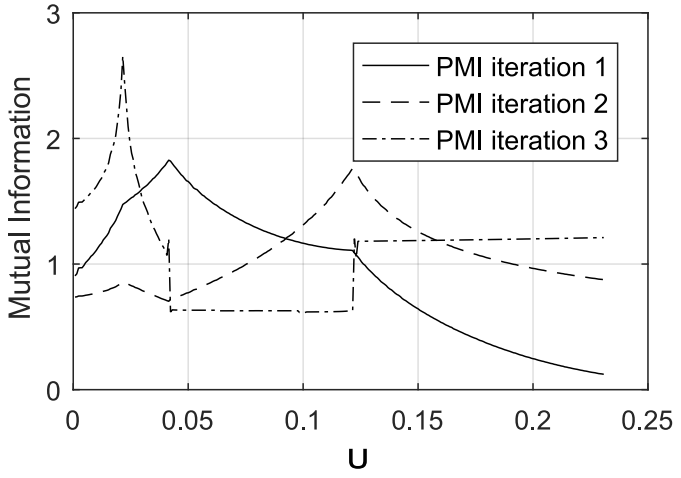

Fig. 14: Mutual Information between T_s and T_r as a function of v [Paper A].

tion. The above analysis have been done on the relation between supply and return temperature, however flow compensation makes sense for other input variables as well.

Somewhere between the supply and return a terminal unit will emit thermal power. The amount of thermal power dissipated relies on zone temperature, so a relation between zone temperature and return temperature would also be subject to flow variable delay. In this work input variable selection is

used with relation to the return, the performance measure, that the reinforcement learning agent receives. In this work the reward function is shaped to maximise comfort and minimise operational cost, see chapter 7.3. Relation between input variables and the return will be subject to flow variable delay. By using the lumped volume approximation for flow variable delay compensation each partial step of selection input variables become

$$\underset{j,v}{\operatorname{argmax}} \quad I\left(x\left(t-\frac{v}{q}\right);y(t)\right) \tag{23}$$

A minimum flow has to be utilised to secure a maximum delay, since zero flow would have to delay go towards infinity. The flow compensation proposed here is derived from a simplified system and is as such an approximation. One approximation that was used is that the flow ratios β stay constant and can therefore be as in the constant lumped parameter v. The terminal units are controlled by regulating valves which will change how the flow ratios are distributed. Changes to outside temperature might change little in ratios due to affecting all zones, were solar radiation only hitting one side of a building might change the ratios more depending on the specific building. Sampling of the continuous system is another approximation error since not all values of delay can be used.

To illustrate the improvement data from an mixing loop heated office building [Appendix E] was analysed with and without flow compensation. Mutual information between return temperature and the three input variables supply temperature and two different zone temperature at the constant delay or flow variable delay that yields the highest mutual information can be seen in table 2.

Model/Input	T_s	T_{z1}	T_{z11}
Constant Delay	0.091	0.032	0.047
Flow Variable Delay	0.102	0.034	0.054
Improvement	12%	6%	15%

Table 2: Highest Mutual Information using Constant or Variable Delay [Paper A].

The mutual information as a function of v can be seen in figure Fig. 15

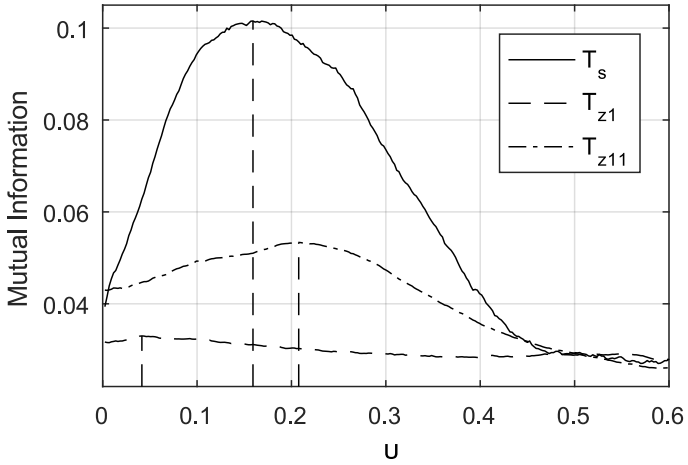

Fig. 15: Mutual Information between T_r and the three inputs T_s,T_{z1} and T_{z11} as a function of v [Paper A].

Here the input variable selection with flow compensation that is used to determine the controlling agents state domain in reinforcement learning was proposed. How this is used in greater detail with relation to reinforcement learning to create a plug & play control scheme can be seen in chapter 7.3.

7.2 Reinforcement learning

Here a reinforcement learning control scheme with flow compensated eligibility trace will be presented. A basic introduction to reinforcement learning and the advantage for the mixing loop application is given first. A broader introduction into reinforcement learning is given in [101].

Basics of reinforcement learning

In reinforcement learning, see Fig. 16 a controlling agent observes which state the environment is in a what reward this yields. The agent over time learns to choose an action on the environment that maximises some sum of rewards. The reinforcement learning theory builds on the assumption that the environment holds the Markov property such that the probability of ending in state s' only depends on the current state s and the action a

$$\mathcal{P}^a_{ss'} = Pr(s_{t+1} = s' | s_t = s, a_t = a). \tag{24}$$

The expected reward to be received can then be described as

$$R^a_{ss'} = \mathbb{E}[r_{t+1} | s_t = s, s_{t+1} = s', a_t = a]. \tag{25}$$

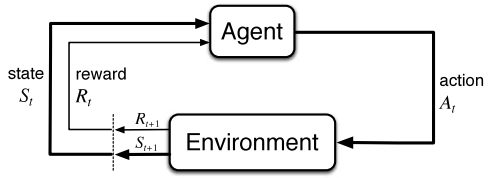

Fig. 16: Agent-environment interaction [103].

The series of rewards the that the controlling agents seeks to maximise is called the return G. An often used return is the weighted ($0 \leq \gamma \leq 1$) sum of rewards

$$G_{t+n} = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... + \gamma^n r_{t+n+1} = \sum_{k=0}^{n-1} \gamma^k r_{t+k+1}, \qquad (26)$$

The agent consists mainly of two parts. A policy π that determines what control actions should be taken and a value function describing the expected return given a state, action and policy.

$$Q_\pi(s,a) = \mathbb{E}[G_{t+\infty}|s_t = s, a_t = a] = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s, a_t = a]. \qquad (27)$$

For the agent to perform optimal control both of these need to be optimal, that is the value function needs to perfectly fit the system and the policy to choose the best action. However they are co-dependent so updating the approximation of the value function (value iteration) is a function of the policy. While updating the policy to yield maximum return (policy iteration) relies on the value function. In most reinforcement learning methods this is done by running both the two processes directly or indirectly in succession which can be illustrated as in Fig. 17.

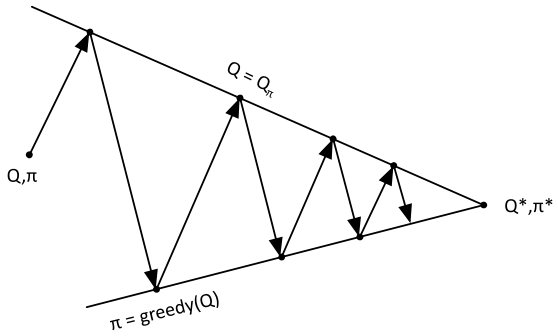

Fig. 17: Value- and policy iteration in succession converging to the optimal value and policy functions [101].

The greedy policy as used in Fig. 17 chooses that action as

$$a_t = \arg\max_a Q_\pi(s_t, a). \tag{28}$$

Reinforcement learning is a self learning optimal control scheme, where optimal refers to the bellman equation being satisfied, which is a necessary conditions for optimality. The bellman equation which for the action value function can be derived from (27) ends up on the form

$$Q_\pi(s,a) = \sum_{s'} \mathcal{P}^a_{ss'} \left[R^a_{ss'} + \gamma \sum_{a'} \pi(s',a') Q_\pi(s',a') \right] \tag{29}$$

If the greedy policy is used then it becomes

$$Q_{\pi^*}(s,a) = \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma \max_{a'} Q_{\pi^*}(s',a') \right] \tag{30}$$

Probably the most used classical reinformcent learning method Q-learning [113], which many later methods builds upon, approximates the bellman equation by at each iteration updating the value function as

$$Q_(s,a) \leftarrow Q(s,a) + \alpha \left[r_{t+1} + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]. \tag{31}$$

This implementation is a one step method since only one value of reward in the update. The value function however still represents the predicted return. Since this is an approximate solution a learning rate ($0 < \alpha \leq 1$) is utilized.

In [Paper B] Q-learning was combined with a gaussian backup scheme onto a tabular representation of the value function. In Fig. 18 results from simulation test [Appendix G] can be seen. In this test the actions that is controlled is pump differential pressure and supply temperature and the state space only contains the outside temperature and time of day. The reward function contained comfort and operational cost, for more specific representation see Chapter 7.3. The compared controllers are classic Q-learning, Q-learning with Gaussian backup, Q-learning with Gaussian backup and presimulation on generic physical model and a industrial controller that is tuned for the specific building.

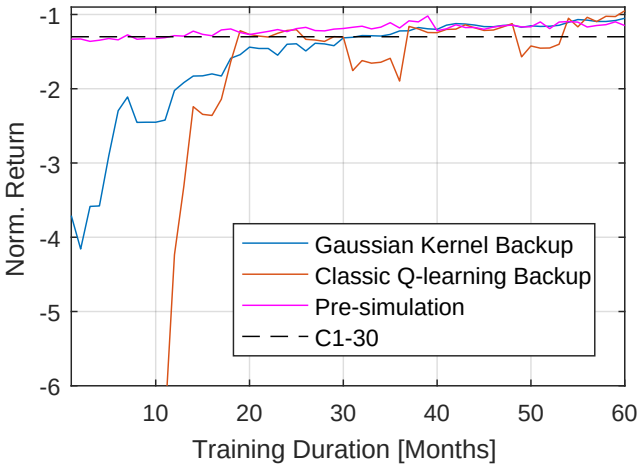

Fig. 18: Norm. returns for a year without further training as a function of training duration. Pre-simulation is on the method Q-learning with gaussian kernel backup. C1-30 is an industrial grade controller that is tuned to the specific building [Paper B].

Table 3 shows the results different controllers on different buildings for a year of operation after having been trained for 60 months prior.

Controller	Norm. Return	RMSE [^oC]	Cost €
Modern House - Copenhagen			
Q	-1.06	1.27	971
C1-15	-1.25	1.31 (3.1%)	1056 (8.0%)
C1-30	-1.19	1.33 (4.5%)	1003 (3.2%)
C1-30-NW	-1.29	1.39 (8.6%)	1018 (4.6%)
Old House - Copenhagen			
Q	-0.96	1.12	1920
C2-15	-1.25	1.11 (-0.9%)	2128 (9.8%)
C2-30	-3.24	1.20 (6.6%)	1985 (3.3%)
C2-30-NW	-4.13	1.26 (11.1%)	2022 (5.0%)
Modern Apartment - Copenhagen			
Q	-0.61	0.96	492
C3-15	-0.72	0.94 (-2.1%)	539 (8.7%)
C3-30	-0.74	0.96 (0.0%)	512 (3.9%)
C3-30-NW	-0.77	1.03 (6.8%)	521 (5.6%)

Table 3: Comparison of Controllers With Setback [Paper B].

There was two takeaways from these initial results of mixing loop control using reinforcement learning. First of all it is possible to improve performance compared to an industrial well tuned controller given a long enough training period. Secondly the training speed has to be improved, with the results showing an infeasible long duration before reaching and overtaking the performance of a well tuned industrial controller.

In the rest of the work three steps are taken to improve training speed of the controller. A multi step method was implemented using a radial basis function approximation, a flow compensated eligibility trace was proposed and partial mutual information was used to choose better state information for the controlling agent.

Function approximation

In the initial work in [Paper B] the value function was represented using a table where the state action space was discretized. Depending on the granularity of the discretization and the numbers of state and actions the dimension quickly increases. To combat this other weight based function approximations are often used where the value function is estimated by a set of weights (w)

$$\hat{Q}(s, a, \mathbf{w}) \approx Q_\pi(s, a). \tag{32}$$

To update the weights the stochastic gradient descent approach is often used [101]

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{1}{2}\alpha\nabla\left[Q_\pi(s, a) - \hat{Q}(s, a, \mathbf{w})\right]^2 \tag{33}$$

$$= \mathbf{w}_t + \alpha\left[Q_\pi(s, a) - \hat{Q}(s, a, \mathbf{w})\right]\nabla\hat{Q}(s, a, \mathbf{w}) \tag{34}$$

where ∇ is a vector of partial derivatives of the weights in dimension d

$$\nabla f(\mathbf{w}) = \left(\frac{\partial f(\mathbf{w})}{\partial w_1}, \frac{\partial f(\mathbf{w})}{\partial w_2}, ..., \frac{\partial f(\mathbf{w})}{\partial w_d}\right) \tag{35}$$

Since $Q_\pi(s, a)$ is unknown an unbiased bootstrapped target is often used U_t as an approximation

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha\left[U - \hat{Q}(s, a, \mathbf{w})\right]\nabla\hat{Q}(s, a, \mathbf{w}). \tag{36}$$

Often bootstrap targets are used that are not unbiased. In (31) the update of Q-learning was shown. If function approximation is used the bootstrap target is $r_{t+1} + \gamma\max_{a'} Q(s', a', \mathbf{w})$. Here it can be seen that the bootstrap target is dependent of the weights making it biased. Methods like this that ignore the influence of the weights on the target are called semi gradient.

Different function approximations have been tried throughout this project, such as the table based in earlier example, but also non-linear deep neural

networks has been tried with different basis functions and forms. The best results has however been using a linear form of radial basis function approximation. This might be due to the smooth nature of radial basis functions fitting the nature of the mixing loop application. The smoothness and differentiability also provide easier solutions for solving the value function with regards to the action providing maximum return.

A linear function approximation is given on the form

$$\hat{Q}(s,a,\mathbf{w}) = \sum_{i=1}^{d} w_i x_i(s,a), \tag{37}$$

where $\mathbf{x}(s,a)$ is a feature vector. In the linear case the stochastic gradient reduces to the simple form

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha \left[U - \hat{Q}(s,a,\mathbf{w})\right] \mathbf{x}(s,a) \tag{38}$$

The radial basis function that is used for the basis of the feature vector is

$$x_i(\mathbf{s},\mathbf{a}) = exp\left(- \sum_{k_s=1}^{n_s} \frac{(s_{k_s} - c_{k_s,i})^2}{2\varsigma_{k_s,i}^2} - \sum_{k_a=n_s+1}^{n_a+n_s} \frac{(a_{k_a} - c_{k_a,i})^2}{2\varsigma_{k_a,i}^2} \right). \tag{39}$$

Here $\mathbf{s}$ is a vector of n_s states and $\mathbf{a}$ is a vector of n_s actions.
$\mathbf{c} = [c_{k_{s,1}}, ..., c_{k_{s,n_s}}, c_{k_{a,1}}, ..., c_{k_{a,n_a}}]$ are the center points and $\boldsymbol{\varsigma}$ the width of the features in the dimension $\mathbb{R}^{n=n_s+n_a}$. In this work the features are placed uniformly over the state action space.

On or off policy

Training on or off policy is a key concept in Reinforcement Learning. Training on policy means approximating the value function for the policy being used for control. Training off policy means finding the value function for a policy other than what is being used for training. Often the greedy policy is desired for control since this maximises the return. However if only actions are taken that by the current knowledge are optimal, no new knowledge of potential better actions will be acquired. So instead a policy that incorporates exploration can be used. An example of such policy could be the ϵ-greedy policy where a random action is chosen with ϵ probability. In this case off policy training could be used where the policy that is being trained is the greedy policy while the controlling policy is ϵ-greedy. In this manner the controlling agent ensures to explore new actions other than the one which by current knowledge is the optimal action. Is it however desired to exploit the current knowledge and control optimal with respect to current knowledge this can be done since off policy training was done towards the greedy policy. The concept of exploration versus exploitation is a common one in self learning optimal control.

The temporal difference error is the error between the bootstrapped target value function and the new estimate, here given for the on policy method State-Action-Reward-State-Action (SARSA)

$$\delta_t^S = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t). \tag{40}$$

The classic Q-learning is an off policy method where the max function is used on the bootstrap target to learn the greedy policy while controlling with another.

$$\delta_t^Q = r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t). \tag{41}$$

In [118] a shift parameter σ was introduced to shift between on and off policy and even used intermediate values which has been shown to improve performance in some cases.

$$\delta_t^\sigma = \sigma_{t+1}\delta_t^S + (1 - \sigma_{t+1})\delta_t^Q. \tag{42}$$

In Chapter 7.3 it is shown how this is used to achieve what is often referred to as apprenticeship training. Here a mentoring controller is used while the apprenticeship is getting some initial training before taking over.

Eligibility trace

In one step methods, which the classic Q-learning is an example of, only a single measured reward is used to generate the bootstrap target. In multi step methods multiple rewards are used to calculate a more accurate bootstrap target. If the full return is used as target then the method becomes a monte carlo method. An n-step method would use the following return in the target

$$G_{t+n} \doteq r_{t+1} + \gamma r_{t+2} + \gamma^{n-1} r_{t+n} + \gamma^n V(S_{t+n}), 0 \leq t \leq T - n \tag{43}$$

Instead of using a single return, n-step returns are often weighed by λ^{n-1}.

$$G_t^\lambda = (1 - \lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} G_{t+n} + \lambda^{T-t-1} G_{t+\infty} \tag{44}$$

With this bootstrap target the semi gradient descent becomes

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha \left[G_t^\lambda - \hat{Q}(s, a, \mathbf{w}) \right] \nabla \hat{Q}(s, a, \mathbf{w}). \tag{45}$$

In this way using λ parametrization all methods in between one step ($\lambda = 0$) and Monte Carlo ($\lambda = 1$) are representable. In these forward looking methods n-steps are taken and rewards sampled before an update can be done. This means that the updates for a Monte Carlo method can only be done after all rewards are sampled and the episode is over taking longs and filling a

lot of memory. To combat this and ease computation eligibility trace is often used. Where the above multi step implementation is called a forward view, eligibilty trace achieves the same updates to the weights using a backward view. The eligibility trace is implemented as a trace vector ($\mathbf{z}$) with the same dimension as the weights, initialised to zero and at each step updated by

$$\mathbf{z_t} \doteq \gamma\lambda\mathbf{z}_{t-1} + \nabla\hat{Q}(s_t, a_t, \mathbf{w}_t). \tag{46}$$

Here the trace for a state is incremented by the gradient and decays by $\gamma\lambda$ as depicted on Fig.

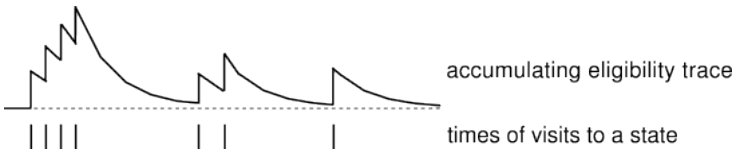

Fig. 19: Eligibility trace of a state as function of times visited [58].

The weights are then adjusted according to

$$\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha\delta_t\mathbf{z}_t. \tag{47}$$

This means that the trace keeps a record of which weights has contributed recently according to the time frame of $\gamma\lambda$. A proof of equivalence of the updates to the weights between the forward view of (45) and the backward view of (47) can be seen in [58]. Now the basic of eligibility trace has been introduced a flow variable eligibility trace is proposed.

Flow variable eligibility trace

The state action value function describes the expected return as a function of the state the system is in and the chosen action. The actions in the mixing loop application are pump speed and supply temperature. To make sure that the impact that the actions instigate on the return is captured in the return horizon a flow compensation is proposed. An example of this is ensuring a high ΔT. Here the return horizon needs to be contained in the return temperature that arises from changing mixing temperature. A trace decay λ based return scheme is used with added flow compensation.

The proposed method lets λ be dependent on the varying transport delay by utilizing a constant parameter ϕ and the scaled flow $q_n(t)$ as in

$$\lambda(q) = \frac{\phi}{q_n(t)} \quad q_n(t) \in [q_{n,min} \leq q_n(t) \leq 1], \tag{48}$$

It is proposed that the ϕ^* giving optimal performance of the controlling agent is a function of the lumped volume v as described in chapter 7.1. The lumped

volume that is used here is the one providing most mutual information between the supply and the return temperature.

$$\phi^* = h(v_n), \tag{49}$$

where v_n is scaled by the maximum flow of the system and the function $h()$ mapping $\phi \in \mathbb{R} : 0 \leq \phi \leq 1$.

In [Paper B] a generic physical model of the mixing loop application was presented to be used for pre-training. The same model was used in [Paper C] to establish the relation between v_n and ϕ^* using an empirical approach. In Fig. 20 two reinforcement learning controllers are compared. One using a constant λ and the other with flow compensated $\lambda(\phi/q)$. The control is run on the model for a year and the sum of the returns over a year is plotted as a function of either λ or ϕ.

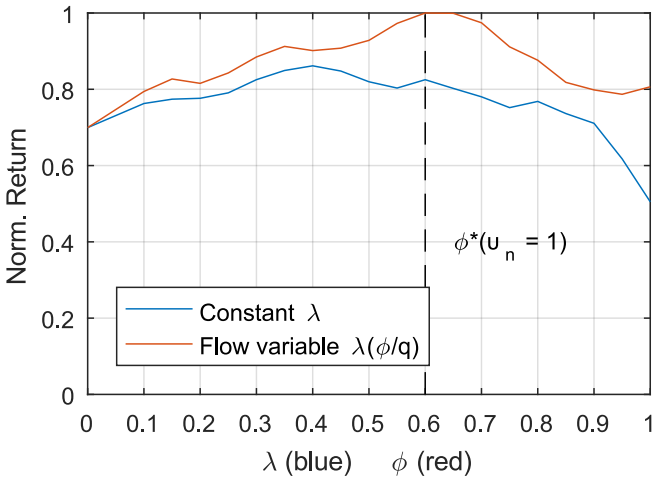

Fig. 20: Norm. yearly return for different values of constant λ and flow variable delay with different α for physical model with $upsilon_n = 20$ [Paper C].

Here the flow compensated λ performs better than the constant. Furthermore in this system with $v_n = 1$ the highest yearly return occurs at $\phi^* = 0.6$. To find the relation $\phi* = h(v_n)$ multiple simulations are run with different v_n. In Fig. 21 the ϕ^* found for different v_n can be seen.

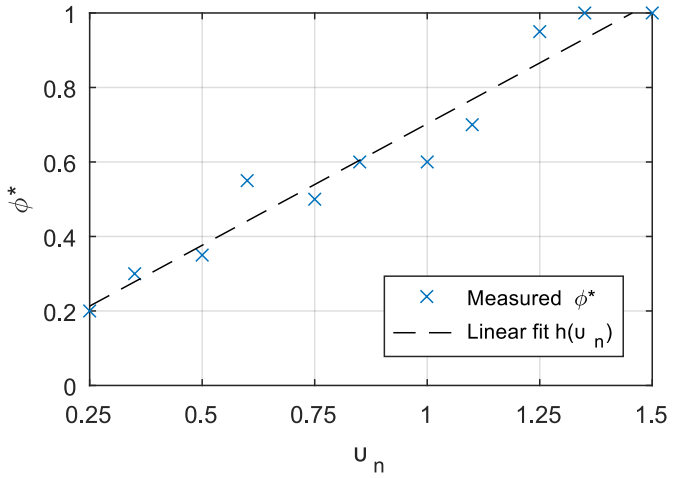

Fig. 21: Data points for relations between ϕ^* and v_n. An affine linear curve is fitted to the data [Paper C].

An affine linear approximation is deemed to fit the relation. For a description of how this is used in the final control scheme see chapter 7.3.

To validate the improvement of using a flow compensation λ the proposed method was compared with other controllers on the experimental setup described in [Appendix F]. Results from the comparison of controllers with lower and higher ϕ compared to ϕ^*, together with one using constant λ and an industrial grade controller can be seen in Table 4.

Controller	Norm. Return	RMSE [K]	Cost €
Q(ϕ*)	-1	0.31	10076
Industrial	-1.44	0.28	12146 (20.5%)
Q($\phi^* - 0.2$)	-1.16	0.25	10519 (4.4%)
Q($\phi^* + 0.2$)	-1.25	0.33	10690 (6.1%)
Q($\lambda = 0.5$)	-1.29	0.29	10751 (6.7%)

Table 4: Comparison of Controllers performance over 6 months after 5 months training [Paper C].

7.3 Plug & play control scheme

A plug and play control scheme is presented here. Plug and play is meant in the sense when the mixing loop is plugged in and connected to the sensor

network then no further tuning or calibration should be needed in regards to control.

Algorithm

To get a better overview of the full plug & play control scheme it is divided into the sequential components illustrated in Fig. 22.

Fig. 22: Overview of components included in the control scheme.

The content of each of the components is described in the following.

Industrial Control

The pseudo code for the industrial control component can be seen in Algorithm 1.

Result: Industrial control with data logging
Initialize : Industrial Controller
Parameters : t_{train}, t_{vali}
begin
  repeat
    Industrial standard mixing loop control
    Logging input variables $\mathbf{x}_s$, actions $\mathbf{a}_s$
    and rewards r_s
  until *Runtime* = t_{train};
  repeat
    Industrial standard mixing loop control
    Logging input variables $\mathbf{x}_v$, actions $\mathbf{a}_v$
    and rewards r_v
  until *Runtime* = t_{vali};
end

Algorithm 1: Component: Industrial Control with data logging [Paper D].

For an initial period an industrial commercial available controller is used. While a, here unnamed, commercial controller is used for this work other controllers are also applicable. For a general understanding on the control in the industrial controller see Chapter 2. While the industrial controller is running data is collected. This data consists of all available measured state

data points, with flow however being necessary for the flow compensations. Furthermore the actions pump speed and supply temperature along with the reward at all time steps are also stored. The reward function giving the rewards to be stored is on the form

$$R(t) = \begin{cases} -(e(t)^2 + \beta(\psi_{heat}(t) + \psi_{pump}(t))) & 5 \leq t \ mod(24h) \leq 21 \\ -\beta(\psi_{heat}(t) + \psi_{pump}(t)) & otherwise \end{cases} \tag{50}$$

Here $e(t)$ is the maximum absolute temperature error for all the measured zones. The parameter β is a weight between comfort and cost due to the multi objective nature of the reward. $\psi_{heat}(t)$ is the heating power cost. By using cost the reinforcement learning will be able to learn the relations that differ for different heat sources. Example of this could be a tariff for high return temperature in district heating or varying electricity cost for self owned heat pump system. The reward function here shown uses a static time setback period. This can be set to match a calendar module.

Determine ϕ

The pseudo code for the determination of ϕ component can be seen in Algorithm 2.

Result: Determine ϕ
Initialize : Load logged data from industrial control
begin
Determine v^* as $\max_v \ I\left(T_{s,t-v/q:T-v/q}; T_{r,t:T}\right)$
Find q_{max} as maximum measured flow in logged data. Normalise
$v_\eta^* = \frac{v^*}{q_{max}}$ Determine the optimal ϕ as $\phi^* = h(v_\eta^*)$
$\lambda(q)$ is computed at all time steps in the logged data as
$\lambda(q) = \frac{\phi^*}{q_\eta(t)}$ Where $q_\eta(t) = \frac{q(t)}{q_{max}}$ Compute $G_{t:T}^{\lambda(q)}$ from logged
data
end

Algorithm 2: Component: Determine ϕ [Paper D]

This component determines ϕ^*. This is used to calculate the flow variable return in the logged return and therefore needs to be the first step. The first task is to find the lumped volume that provides highest mutual information between the supply temperature and the return. From the normalised lumped volume ϕ^* is determined from the linear approximation $h()$. When

ϕ^* has been determined the flow variable return from the initial logged data can be computed and used in the input variable selection. Later ϕ^* is used in the reinforcement learning control to establish the trace length online.

Input Variable Selection

The pseudo code for the input variable selection component can be seen in Algorithm 3.

Result: Input Variable Selection
Initialize : Load training data of all inputs $\mathbf{x}_{t:t+m_s}$, return $G^{\lambda(q)}_{t:t+m_s}$ and flow $q_{t:t+m_s}$. Load validation data of n inputs $\mathbf{x}_{t:t+m_v}$, return $G^{\lambda(q)}_{t:t+m_v}$ and flow $q_{t:t+m_v}$.
Parameters : tol
begin
Remove information given by actions by
$G^{\lambda(q)}_{t:t+m_s} \leftarrow G^{\lambda(q)}_{t:t+m_s} - E[G^{\lambda(q)}_{t:t+m_s}|\mathbf{a}_{t:m_s}]$
$\mathbf{x}_{t:t+m_s} \leftarrow \mathbf{x}_{t:t+m_s} - E[\mathbf{x}_{t:t+m_s}|\mathbf{a}_{t:m_s}]$
add actions **a** to set of selected inputs **z** **repeat**
Find input with highest mutual information as
$z_{s,t:m_s} \leftarrow \max_{j,v} \; I\left(\mathbf{x}^j_{t-v/q:t+m_s-v/q}; G^{\lambda(q)}_{t:t+m_s}\right)$
Generate estimators $E[G^{\lambda(q)}_{t:t+m_s}|z_{s,t:m_s}]$
and $E[\mathbf{x}_{t:t+m_s}|z_{s,t:m_s}]$
Calculate residuals as
$G^{\lambda(q)}_{t:t+m_s} \leftarrow G^{\lambda(q)}_{t:t+m_s} - E[G^{\lambda(q)}_{t:t+m_s}|z_{s,t:m_s}]$
$\mathbf{x}_{t:t+m_s} \leftarrow \mathbf{x}_{t:t+m_s} - E[\mathbf{x}_{t:t+m_s}|z_{s,t:m_s}]$
Add z to set of selected inputs **z**
$RMSE \leftarrow \sqrt{\frac{\sum_{t=1}^{m_v}\left(G^{\lambda(q)}_{t:t+m_v} - E[G^{\lambda(q)}_{t:t+m_v}|z_{v,t:t+m_v}]^2\right)}{m_v}}$
$RMSE_{prev} \leftarrow RMSE$
until $tol > \frac{RMSE_{prev} - RMSE}{RMSE_{prev}}$;
end

Algorithm 3: Component: Input Variable Selection [Paper D]

The objective of the input variable selection is to choose a subset of input variables to represent the value function. This is done by choosing the inputs that holds the most information with respect to predicting the return. The input selection is done from data logged during the initial industrial control phase. The returns calculated in the determine ϕ component is also loaded. The dataset is divided into two parts, one for training and one for cross validation used as stopping criteria. A tolerance is declared. If there is less improvement than the tolerance in the cross validation RMSE by adding an additional input the input selection is stopped. Prediction models based on radial basis networks are used to sort the information in the selection input from the remaining inputs and return to establish the residuals used for selecting the following inputs. Since the actions are used in the state action value function to predict the return these can be considered as input variables that are predetermined for the subset of input variables. Therefore, the information given from the actions needs to be removed before analysing the remaining input variables. After the stopping criteria is reached a subset of input variables with respect to index and lumped volume has been determined and the next component can be initiated.

Pre-training

The pseudo code for the pre-training component can be seen in Algorithm 4.

Result: Pre-training
Initialize : Load logged data for selected input sets $\mathbf{z_{t:T}}$, flow $q_{t:T}$, rewards $r_{t:T}$ and ϕ^*. Set weights $\mathbf{w}$ and trace vector $\mathbf{z}$ to zero.
begin
  Set $\sigma = 0$ to train off policy
  Train reinforcement learning as Algorithm 6 on logged data
  $\mathbf{w_{pt}} \leftarrow \mathbf{w}$
end

Algorithm 4: Component: Pre-training [Paper D]

The objective of the pre-training component is to use the initial gathered data to pre-train the reinforcement learning controller before it is used for online control. For pre-training the dataset from the initial phase of the industrial control is used for the input variables that was selected. The whole time series is used as one and not divided as in the previous component. Since the data was gathered while being controlled by the industrial controller the reinforcement learning has to train off-policy and σ is there set to zero. After the reinforcement learning has trained on the initial data the weights are saved to use in the online reinforcement learning control.

Reinforcement Learning Control

The pseudo code for the reinforcement learning control component can be seen in Algorithm 5.

Result: Online $Q_\phi(\sigma, \lambda)$
Initialize : Load ϕ^*. Weights $\mathbf{w} = \mathbf{w_{pt}}$, trace vector $\mathbf{z}$. Take action $\mathbf{a}'$ according to ϵ-greedy $\pi(.|\mathbf{s}_0)$. Calculate feature state $\mathbf{x} = \mathbf{x}(\mathbf{s_0}, \mathbf{a}')$.
$Q_{old} = 0$
Parameters : ϵ, α, γ, σ
repeat every sample
  Observe r and $\mathbf{s}'$
  Choose $\mathbf{a}'$ according to ϵ-greedy π
  $\mathbf{x}' \leftarrow \mathbf{x}(\mathbf{s}', \mathbf{a}')$
  $Q \leftarrow \mathbf{w}^T\mathbf{x}$
  $Q'_S \leftarrow \mathbf{w}^T\mathbf{x}'$
  $Q'_Q \leftarrow \max_{\mathbf{a}'} (\mathbf{w}^T\mathbf{x}(\mathbf{s}', \mathbf{a}'))$
  $\delta^\sigma \leftarrow \sigma(r + \gamma Q'_s - Q) + (1 - \sigma)(r + \gamma Q'_Q - Q)$
  Observe flow q
  if $q_{max} \leq q$ **then**
    $q_\eta \leftarrow 1$
  else if $q \leq q_{min}$ **then**
    $q_\eta \leftarrow q_{min}/q_{max}$
  else
    $q_\eta \leftarrow q/q_{max}$
  end
  $\lambda \leftarrow \frac{\phi^*}{q_\eta}$
  $\mathbf{z} \leftarrow \gamma\lambda\mathbf{z} + (1 - \alpha\gamma\lambda\mathbf{z}^T\mathbf{x})\mathbf{x}$
  $\mathbf{w} \leftarrow \mathbf{w} + \alpha(\delta^\sigma + Q - Q_{old})\mathbf{z} - \alpha(Q - Q_{old})\mathbf{x}$
  $Q_{old} \leftarrow \sigma Q'_S + (1 - \sigma)Q'_Q$
  $\mathbf{x} \leftarrow \mathbf{x}'$
  Take action $\mathbf{a}'$
until *Mixing Loop Stop;*

Algorithm 5: Component: Reinforcement Learning Control [Paper D]

The objective of this component is to implement reinforcement learning control. To get a better initial performance the pre-training weights are used. The reinforcement learning scheme contains flow variable eligibility trace as introduced. The trace is implemented as a dutch trace as proposed in [45] due to good computational properties.

7.4 Results on performance of control scheme

A summary of the results regarding the plug and play control scheme is here done. The control scheme is tested on a hardware in the loop test setup. The test consists of two parts; a hardware part where a mixing loop system is supplying a heat exchanger. The heat exchanger is cooled by a chiller and that is controlled to match a load model. The load model controls the flows and temperature from the heat exchanger such that the office buildings behaviour is emulated. The load model is generated from data logged in the office building as can be seen in [Appendix E]. Multiple parallel hardware in the loop test are run under same load conditions, but with different controllers for comparison. A more thorough description of the experimental setup is in [Appendix F].

To illustrate the process of the proposed plug and play algorithm results from the intermediate components are shown along with the final results. In Fig. 23 the lumped volume giving highest mutual information between the supply and return temperature can be seen. This is used in the component determine ϕ^* and lead to $\phi^* = 0.8$.

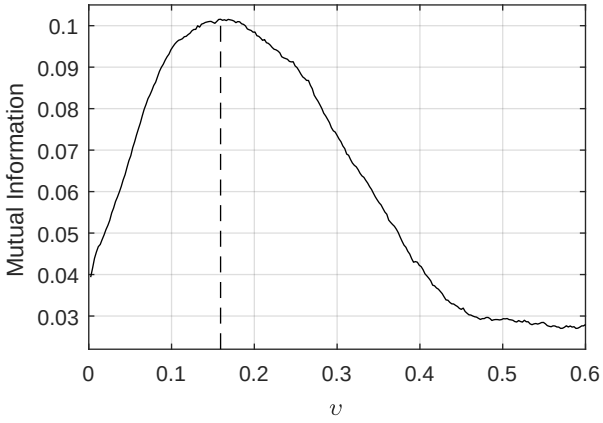

Fig. 23: Mutual information between supply temperature and return temperature as a function of v [Paper D]

From ϕ^* the flow variable return can be computed for the logged data. The computed return is used in the 3. component Input Variable Selection.

In Fig. 24 prediction error on validation data of of the return is shown as a function of input set length. In this way it can be seen how the prediction error improves by adding the next input that is chosen. The stopping criteria of component 3 stopped after the first 8 inputs are selected, with the actions being the first 2. For a better overview the prediction error was calculated for

the first 15 inputs as shown in Fig. 24.

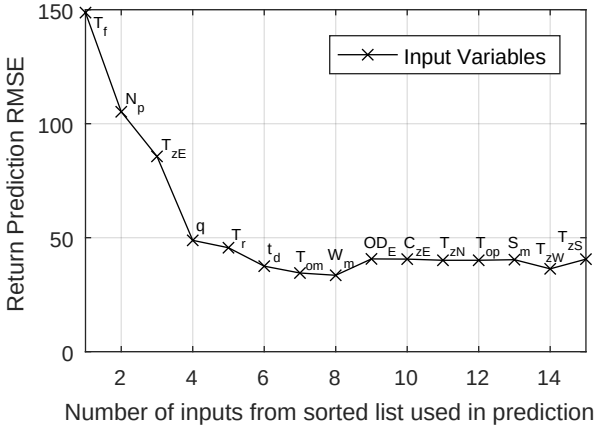

Fig. 24: Return prediction error as function on number of inputs from sorted list. The inputs are shown at the lumped volume which leads to highest mutual information [Paper D]. Inputs: Forward Temperature (T_f), pump Speed (N_p), temperature eastern zone (T_{zE}), flow (q), return temperature (T_r), time of day (t_d), outdoor measured temperature (T_{om}), measured wind speed (W_m), Opening degree radiator valve eastern zone (OD_E), CO_2 level eastern zone (C_{zE}), temperature northern zone (T_{zN}), outdoor predicted temperature (T_{op}), Solar radiation measured (S_m), temperature western zone (T_{zW}), temperature southern zone (T_{zS}).

To test how other subsets of input variables would be perform compared to the chosen subset multiple subsets was run for 200 days. The first plot of Fig. 25 is a representation of the convergence of the weights. Instead of plotting all weights a normalised sum of the weights is shown is for an easier overview. The number of input variables used in the state subset is declared as n_s. The plug and play control scheme stopped after the 6 first input variables containing highest partial mutual information ($n_s = 6$). The results show that subsets with lower amount of states leads to a faster convergence speed. The flow variable return at all time steps is shown in Fig. 25. The plug and play controller ($n_s = 6$) achieves the highest return during this time period. It is to be expected that versions with larger state spaces will in time converge and give same or higher return as the chosen subset.

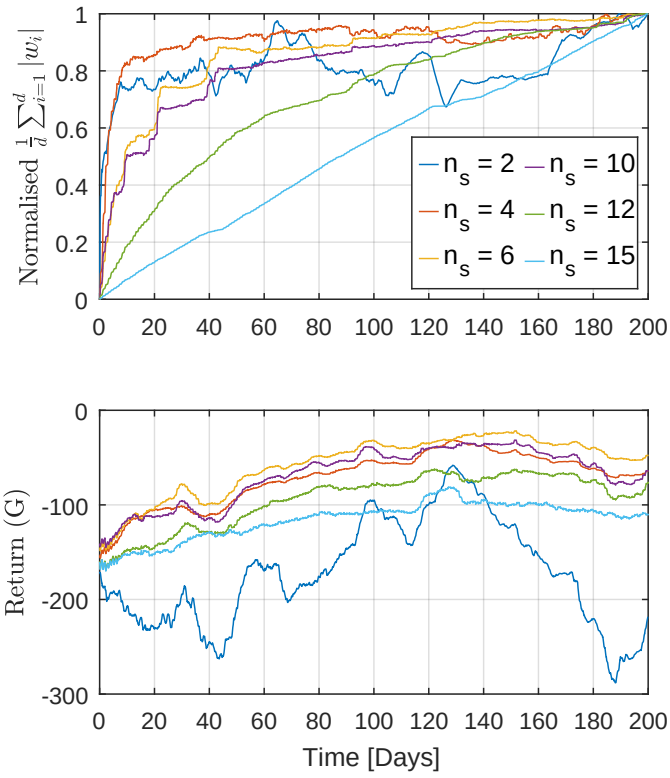

Fig. 25: Comparison of returns and weight convergence for different sets of states [Paper D].

In Fig. 26 both the plug and play control scheme and a industrial controller that is hand tuned are tested for 140 days. The first plot shows the RMSE of the average zone temperature for the whole building. The second plot shows at every sample the summation of operating cost for the last 14 days. The third plot shows the return which the reinforcement learning agent seeks to maximise. The fourth plot shows the sum of absolute values of the weights.

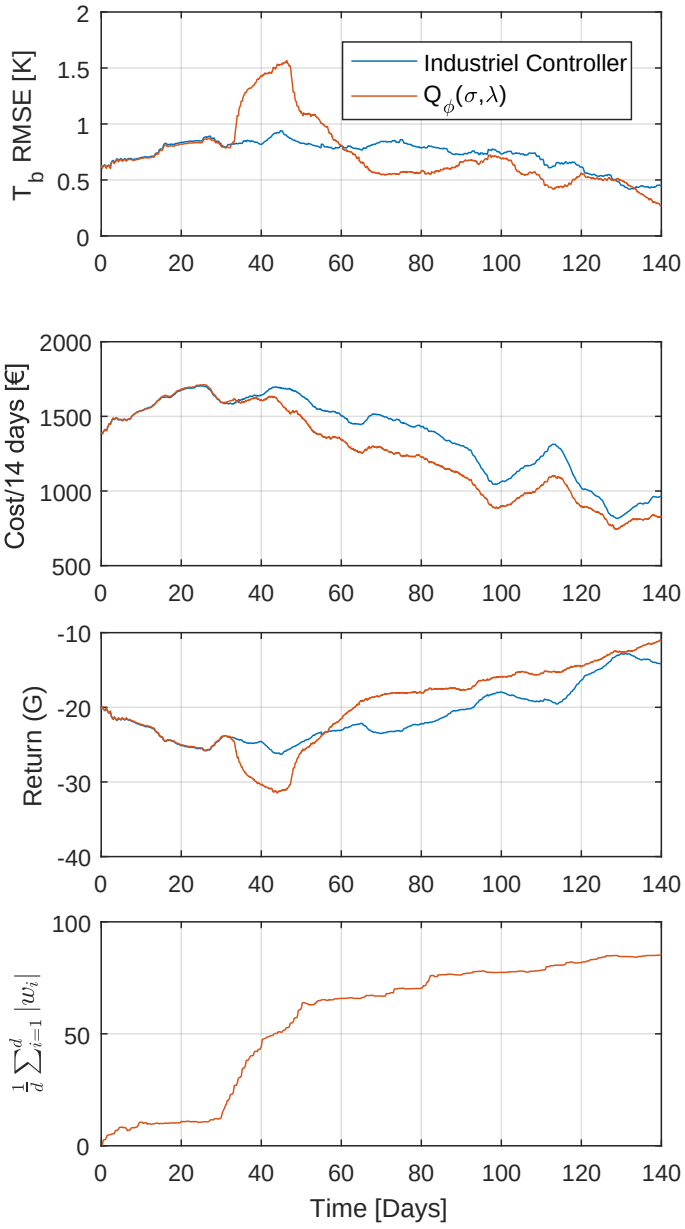

Fig. 26: Comparison of Building Temperature RMSE with a 14 days running window. Operating cost for running 14 days. The return and the summation of absolute weights [Paper D].

For the first 30 days the control is identical due to this being the initial data gathering phase for the plug and play control scheme. In the weights

plot it can be seen that from the data during this initial period pre-training takes the weights only a small way towards the value it later converges towards. This is due to the industrial control only exploring a narrow area of the state-action space. At 30 days when the reinforcement learning takes over control the performance goes down. At this stage the reinforcement learning controller only has the information gained from off policy training on the initial data. After approximate 25 days the performance in the measure of the return equals that of the industrial controller. From this point onwards the performance improves compared to the industrial controller. When measured from day 60 to day 140 the plug and play control scheme improves the comfort by having 19% less temperature error while saving 14% on operational cost.

8 Conclusion and recommendations

The work presented in this thesis is the development of a plug and play control scheme for building heating via mixing loops. The problem originated from a desire to implement optimal control to decrease operational cost of the mixing loop while keeping high comfort without a long manual commissioning phase of the system. The result of this work is presented as a collection of papers enclosed in Part II of this thesis, and has been summarized in the previous chapters. This chapter presents the conclusions drawn on the basis of these results, as well as the author's recommendation for future investigations on the subject of plug and play mixing loop control.

8.1 Conclusion

A plug and play control strategy was employed to deal with the challenges of providing optimal control from a prefabricated mixing loop system to a variety of buildings and HVAC systems without a lengthy commissioning phase. The main contributions in this project are within the four categories: Input variable selection, reinforcement learning, plug and play control scheme and experimental validation.

Input Variable Selection
A flow compensated input variable selection scheme was derived in [Paper A]. The input variable selection was based on partial mutual information to determine a subset of input variables holding largest mutual information with respect to an estimation target. Due to the challenge of flow variable delay a flow compensation was derived based first principle physical domain knowledge.

Reinforcement Learning
A reinforcement learning controller was proposed in [Paper B]. A generic model of the mixing loop system was proposed to pre-train on to improve initial performance of the reinforcement learning agent. To improve training speed a Gaussian kernel backup for tabular methods was proposed. While the control scheme was shown to improve performance over industrial controllers the training speed was low. The period before reaching the performance level of the industrial controller and thereafter improving was longer than deemed feasible for the mixing loop application.

An improved reinforcement learning controller was proposed in [Paper C]. This controller was implemented as a multi step agent using eligibility trace to improve training speed. A flow variable eligibility trace was designed an shown to improve performance on a mixing loop systems. The proposed reinforcement learning scheme improved on training speed and reached the performance level of the industrial controller within 50 days.

Plug and play control scheme
A plug and play control scheme was proposed in [Paper D]. This control scheme utilized the input variable selection of [Paper A] to find a subset of inputs to represent the state space of the reinforcement learning control proposed in [Paper C]. In this control scheme an industrial controller controls for an initial period while the reinforcement learning agents learns off policy. It was shown that from the initial data the plug and play control scheme can determine a suitable state space.

Experimental validation
Different experimental setups was used in this work. To test the input variable selection data was gathered from an office building as described in [Appendix E]. The performance of the reinforcement learning algorithm proposed in [Paper B] was tested via high fidelity simulation as described in [Appendix G]. A hardware in the loop experimental setup [Appendix F] was used to first test the reinforcement learning in [Paper C] and later the plug and play control scheme in [Paper D]. For the final proposed plug and play control scheme the experimental results showed that the after data has been gathered and the reinforcement learning agent takes over control it takes around 25 days for agent to achieve same performance level as the industrial controller. After this initial period the control scheme improves further on performance and measured from day 60 to 140 the temperature error was reduced by 19% while saving 14% on operational cost.

The savings potential of the proposed plug and play control scheme is dependent of the compared baseline control. In this work a commercial available industrial controller is tuned specifically for the tested building to provide a baseline. Such a well tuned controller is a challenging baseline since it

requires that the controller is well commissioned. A less challenging baseline as could be found in a building without proper commissioning would lead to a higher savings potential and a shorter training period to reach same level of performance.

8.2 Recommendations

This section presents some of the author's recommendations for future research directions within the field of plug and play control for mixing loops. It also lists some of the practical challenges that arises with data driven self learning control.

A challenge of reinforcement learning based controllers is the poor performance before a certain amount of knowledge of the system has been gained by the controlling agent. In this work pre-training on a generic model and by data gathered using and industrial controller has been tried to give better initial performance. While the initial performance in the experimental setup is deemed within what may be considered tolerable for the occupants this would still have to be tested further. Field test would have to be done to ensure that the level of the performance in the initial period does not cause so much inconvenience for the occupants that it outweighs the later improved performance.

Further research should also be done into improving initial performance or increasing training speed. In this work a generic model based on first principle physics was proposed to increase initial performance. Another idea could be to create a library of models which is characterised by some hyper parameters that are easily determinable by the installer. This could be area of the building, build year, number of zones, ratio of windows etc. When the installer has entered values for the hyper parameters the matching model is then chosen and performed pre-training on to improve initial performance.

Another way of increasing training speed could be by finding a structure for the value function of lower dimension that fits the mixing loop application. If the structure does not fit variations of systems that the mixing loop is installed into the increased training speed may come at the cost of performance. One initial approach could be to derive the structure from already existing commercial available controllers which would then lead to an self learning optimal scheme for finding the optimal parameters in already proven control structures.

Another approach to increasing training speed could be by adding input variables over time, often in the literature called curriculum learning. By adding a small subset of inputs that holds most information first and let the agent learn to control these first a good initial performance can quickly be obtained. Then when this "curriculum" is learned further input variables are added. In the proposed method a subset of inputs is found and added at

once. It should be studied if breaking the selected inputs into smaller subsets and adding over time could increase initial learning rate.

The self learning nature of reinforcement learning comes at the price of exploration where actions are taken that may be random in nature or simply non optimal by the current knowledge of the controlling agent. In the mixing loop application this may lead to discomfort of the occupants of the building zones. More research should be done on how exploration can be done in a manner which causes a minimum of discomfort for the occupants. This may be by constraining the exploration space of the actions or simply stop all exploration for a while if some constraint on a state has been triggered.

The input variable selection scheme is such that it can choose to add more delay measurements of the same variable. In higher order dynamic systems more samples of an input variable may be needed to represent a state, such as multiple position measurements for a velocity. Given the large dynamic effects due to the thermal capacities in a building it was anticipated that some input variables would hold a large amount of information at multiple delays. This has however not been the case in the experimental setups where only little information is left at other delays when an input variable has been selected. This causes a discrepancy between the physical understanding of the system and the results of the data driven analysis that gives cause for further study. Maybe the delay is of much larger influence than the dynamical effects in this system and therefore overshadows the effects of the dynamics.

In building heating multiple mixing loops are often used to control different zones. These mixing loops will often be codependent. This could be due to being supplied from the same pressure and thermal energy source or the heating zones being adjacent. In such cases providing optimal control across the different mixing loops either by distributed or global optimisation could improve performance.

While a large number of reinforcement learning approaches has been tried during this work the popularity of the research topic means that new variations are developed frequently. Especially a large amount of policy gradient methods are being showcased in the literature with good results on various applications. Continually study of new reinforcement learning methods applicable to the mixing loop application should be done.

References

[1] A. Rupam Mahmood, Huizhen Yu, Martha White, Richard S. Sutton, "Emphatic Temporal-Difference Learning," *arXiv:1507.01569*, 2015.

[2] A. Afram, F. Janabi-Sharifi, A. S. Fung, and K. Raahemifar, "Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC

systems: A state of the art review and case study of a residential HVAC system," *Energy and Buildings*, vol. 141, pp. 96–113, 2017.

[3] T. Ahonen, J. Tamminen, J. Ahola, J. Viholainen, N. Aranto, and J. Kestilä, "Estimation of pump operational state with model-based methods," *Energy Conversion and Management*, 2010.

[4] S. Ali and D. H. Kim, "Energy conservation and comfort management in building environment," *International Journal of Innovative Computing, Information and Control*, 2013.

[5] J. D. Álvarez, J. L. Redondo, E. Camponogara, J. Normey-Rico, M. Berenguel, and P. M. Ortigosa, "Optimizing building comfort temperature regulation via model predictive control," *Energy and Buildings*, 2013.

[6] J. Arifovic and R. Gençay, "Using genetic algorithms to select architecture of a feedforward artificial neural network," *Physica A: Statistical Mechanics and its Applications*, 2001.

[7] Z. Artstein, "Linear Systems with Delayed Controls: A Reduction," *IEEE Transactions on Automatic Control*, 1982.

[8] B. Asare-Bediako, P. F. Ribeiro, and W. L. Kling, "Integrated energy optimization with smart home energy management systems," in *IEEE PES Innovative Smart Grid Technologies Conference Europe*, 2012.

[9] F. Ascione, N. Bianco, C. De Stasio, G. M. Mauro, and G. P. Vanoli, "Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort," *Energy and Buildings*, 2016.

[10] E. Atam and L. Helsen, "Control-Oriented Thermal Modeling of Multizone Buildings: Methods and Issues: Intelligent Control of a Building System," *IEEE Control Systems*, vol. 36, no. 3, pp. 86–111, 2016.

[11] P. Bacher and H. Madsen, "Identifying suitable models for the heat dynamics of buildings," *Energy and Buildings*, vol. 43, no. 7, pp. 1511–1522, 2011.

[12] J. A. Bagnell and J. Schneider, "Covariant policy search," in *IJCAI International Joint Conference on Artificial Intelligence*, 2003.

[13] R. Battiti, "Using Mutual Information for Selecting Features in Supervised Neural-Net Learning," *Ieee Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, 1994.

[14] N. Bekiaris-Liberis and M. Krstic, "Compensation of state-dependent input delay for nonlinear systems," *IEEE Transactions on Automatic Control*, 2013.

[15] G. J. Bowden, G. C. Dandy, and H. R. Maier, "Input determination for neural network models in water resources applications. Part 1 - Background and methodology," *Journal of Hydrology*, 2005.

[16] G. J. Bowden, J. B. Nixon, G. C. Dandy, H. R. Maier, and M. Holmes, "Forecasting chlorine residuals in a water distribution system using a general regression neural network," *Mathematical and Computer Modelling*, 2006.

[17] BrainBoxAI, "BrainBoxAI." [Online]. Available: https://www.brainboxai.com/

[18] BuildingIQ, "BuildingIQ." [Online]. Available: https://buildingiq.com/

[19] Y. Chen, L. K. Norford, H. W. Samuelson, and A. Malkawi, "Optimal control of HVAC and window systems for natural ventilation through reinforcement learning," *Energy and Buildings*, 2018.

[20] J. Cigler and S. Prívara, "Subspace identification and model predictive control for buildings," in *11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010*, 2010, pp. 750–755.

[21] B. J. Claessens, P. Vrancx, and F. Ruelens, "Convolutional Neural Networks for Automatic State-Time Feature Extraction in Reinforcement Learning Applied to Residential Load Control," *IEEE Transactions on Smart Grid*, 2018.

[22] C. Cortes, M. Mohri, and a. Rostamizadeh, "L2 regularization for learning kernels," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009.

[23] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2005.

[24] K. Dalamagkidis, D. Kolokotsa, K. Kalaitzakis, and G. S. Stavrakakis, "Reinforcement learning for energy conservation and comfort in buildings," *Building and Environment*, vol. 42, no. 7, pp. 2686–2698, 2007.

[25] Danfoss, "ECL Controller." [Online]. Available: https://store.danfoss.com/en/Heating-and-District-Energy/Electronic-Controllers-and-Monitoring-solutions/ECL-Comfort-Controllers/c/6286

[26] Danish Energy Agency, "Energy Policy Toolkit for Energy Efficiency in New Buildings."

[27] P. Davidsson and M. Boman, "Distributed monitoring and control of office buildings by embedded agents," *Information Sciences*, 2005.

[28] N. K. Dhar, N. K. Verma, and L. Behera, "Adaptive Critic-Based Event-Triggered Control for HVAC System," *IEEE Transactions on Industrial Informatics*, 2018.

[29] H. Doukas, K. D. Patlitzianas, K. Iatropoulos, and J. Psarras, "Intelligent building energy management system using rule sets," *Building and Environment*, 2007.

[30] A. I. Dounis and C. Caraiscos, "Advanced control systems engineering for energy and comfort management in a building environment-A review," pp. 1246–1261, 2009.

[31] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, 2018.

[32] L. Eller, L. C. Siafara, and T. Sauter, "Adaptive control for building energy management using reinforcement learning," in *Proceedings of the IEEE International Conference on Industrial Technology*, vol. 2018-Febru, 2018, pp. 1562–1567.

[33] A. K. Emrah Biyik, "A predictive control strategy for optimal management of peak load, thermal comfort, energy storage and renewables in multi-zone buildings," *Journal of Building Engineering*, vol. 25, 2019.

[34] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *Bernoulli*, 2009.

[35] P. Fazenda, K. Veeramachaneni, P. Lima, and U. M. O'Reilly, "Using reinforcement learning to optimize occupant comfort and energy usage in HVAC systems," *Journal of Ambient Intelligence and Smart Environments*, vol. 6, no. 6, pp. 675–690, 2014.

[36] T. M. Fernando, H. R. Maier, and G. C. Dandy, "Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach," *Journal of Hydrology*, 2009.

[37] P. M. Ferreira, A. E. Ruano, S. Silva, and E. Z. Conceição, "Neural networks based predictive control for thermal comfort and energy savings in public buildings," *Energy and Buildings*, 2012.

[38] E. Fridman, "Time-Delay Systems," in *Control and Mechatronics*, 2019.

[39] S. Gao, G. V. Steeg, and A. Galstyan, "Estimating Mutual Information by Local Gaussian Approximation," *Proceedings of the 31st Conference on Uncertainties in Artificial Intelligence*, p. 224, 2015. [Online]. Available: http://arxiv.org/abs/1508.00536

[40] F. Gers, "Long short-term memory in recurrent neural networks," *Neural Computation*, 2001.

[41] J. Gorodkin, L. K. Hansen, A. Krogh, C. Svarer, and O. Winther, "A quantitative study of pruning by optimal brain damage," *International Journal of Neural Systems*, 2004.

[42] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research (JMLR)*, 2003.

[43] O. Gym, "OpenAI Gym." [Online]. Available: https://gym.openai.com/docs/{#}review

[44] H. Van Hasselt A. Guez and D. Silver, "Deep Reinforcement Learning with Double Q-Learning," *AAAI*, 2016.

[45] R. S. S. Harm van Seijen, A. Rupam Mahmood, Patrick M. Pilarski, Marlos C. Machado, "True Online Temporal-Difference Learning," *Journal of Machine Learning Research*, vol. 17, pp. 1–40, 2016.

[46] International Energy Agency, "Energy Efficiency 2018," Tech. Rep., 2018.

[47] M. Jankovic, "Cross-Term Forwarding for Systems With Time Delay," *Transactions on Automatic Control*, vol. 54, pp. 498–511, 2009.

[48] S. V. Johansen, J. D. Bendtsen, M. R.-Jensen, and J. Mogensen, "Broiler weight forecasting using dynamic neural network models with input variable selection," *Computers and Electronics in Agriculture*, 2019.

[49] J. H. Kim and F. L. Lewis, "Model-free H infinity control design for unknown linear discrete-time systems via Q-learning with LMI," *Automatica*, 2010.

[50] L. Klein, J. Y. Kwak, G. Kavulya, F. Jazizadeh, B. Becerik-Gerber, P. Varakantham, and M. Tambe, "Coordinating occupant behavior for building energy and comfort management using multi-agent systems," in *Automation in Construction*, 2012.

[51] J. Z. Kolter and A. Y. Ng, "Regularization and feature selection in least-squares temporal difference learning," 2009.

[52] G. Konidaris, "Value Function Approximation in Reinforcement Learning using the Fourier Basis," *Learning*, 2008.

[53] R. M. Kretchmar and C. W. Anderson, "Comparison of CMACs and radial basis functions for local function approximators in reinforcement learning," in *IEEE International Conference on Neural Networks - Conference Proceedings*, 1997.

[54] M. Krstic, "On compensating long actuator delays in nonlinear control," in *Proceedings of the American Control Conference*, 2008.

[55] ——, "Lyapunov stability of linear predictor feedback for time-varying input delay," in *Proceedings of the IEEE Conference on Decision and Control*, 2009.

[56] ——, "Input delay compensation for forward complete and strict-feedforward nonlinear systems," *IEEE Transactions on Automatic Control*, 2010.

[57] W. H. Kwon and A. E. Pearson, "Feedback Stabilization of Linear Systems with Delayed Control," *IEEE Transactions on Automatic Control*, 1980.

[58] M. Lee, "Eligibility Trace," 2005. [Online]. Available: http://incompleteideas.net/book/first/ebook/node75.html

[59] X. Li, H. R. Maier, and A. C. Zecchin, "Improved PMI-based input variable selection approach for artificial neural network and other data driven environmental and water resource models," *Environmental Modelling & Software*, vol. 65, pp. 15–29, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364815214003545

[60] Z. Liao, M. Swainson, and A. L. Dexter, "On the control of heating systems in the UK," *Building and Environment*, 2005.

[61] S. Liu and G. P. Henze, "Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 1. Theoretical foundation," *Energy and Buildings*, vol. 38, no. 2, pp. 142–147, 2006.

[62] M. Jankovic, "Control of nonlinear systems with time delay," in *Decision and Control*, 2003, pp. 4545–4550.

[63] ——, "Control of cascade systems with time delay – the integral cross-term approach," in *Decision and Control*, 2006, pp. 2547–2552.

[64] M. Nihtila, "Adaptive control of a continuous time system with time varying input delay," *Systems & Control Letters*, vol. 12, pp. 357–364, 1989.

[65] H. R. Maei and R. S. Sutton, "GQ(lambda): A general gradient algorithm for temporal-difference prediction learning with eligibility traces," 2010.

[66] A. Z. Manitius and A. W. Olbrot, "Finite Spectrum Assignment Problem for Systems with Delays," *IEEE Transactions on Automatic Control*, 1979.

[67] N. J. I. Mars and G. W. van Arragon, "Time Delay Estiamtion in Non-Linear Systems using Average Amount of Mutual Information Analysis," vol. 4, pp. 139–153, 1981.

[68] R. May, G. Dandy, and H. Maier, "Review of Input Variable Selection Methods for Artificial Neural Networks," *Artificial Neural Networks - Methodological Advances and Biomedical Applications*, no. August 2016, p. 362, 2011.

[69] R. J. May, H. R. Maier, G. C. Dandy, and T. M. K. G. Fernando, "Non-linear variable selection for artificial neural networks using partial mutual information," *Environmental Modelling & Software*, vol. 23, no. 10-11, pp. 1312–1326, 2008.

[70] E. Mills, H. Friedman, T. Powell, N. Bourassa, D. Claridge, T. Haasl, and M. A. Piette, "The cost-effectiveness of commercial-buildings commissioning," *HPAC Engineering*, 2005.

[71] MIT Technology Review, "We analyzed 16,625 papers to figure out where AI is headed next." [Online]. Available: https://www.technologyreview.com/s/612768/we-analyzed-16625-papers-to-figure-out-where-ai-is-headed-next/

[72] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning." *Nature*, 2015.

[73] V. Mnih, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, and D. Silver, "Asynchronous Methods for Deep Reinforcement Learning arXiv : 1602 . 01783v2 [cs . LG] 16 Jun 2016," *CoRR*, 2016.

[74] V. Mnih, D. Silver, and M. Riedmiller, "Playing Atari with Deep Neural Nets," *Advances in Neural Information Processing Systems*, 2013.

[75] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu, and J. G. Slootweg, "On-line Building Energy Optimization using Deep Reinforcement Learning," 2018.

[76] N. Morel, M. Bauer, M. El-Khoury, and J. Krauss, "Neurobat, a Predictive and Adaptive Heating Control System Using Artificial Neural Networks," *International Journal of Solar Energy*, vol. 21, no. 2-3, pp. 161–201, 2001.

[77] R. Munos, T. Stepleton, A. Harutyunyan, and M. Bellemare, "Safe and Efficient Off-Policy Reinforcement Learning," *Advances in Neural Information Processing Systems 29*, 2016.

[78] D. S. Nicolas Heess, Jonathan J Hunt, Timothy P Lillicrap, "Memory-based control with recurrent neural networks," *arXiv:1512.04455*.

[79] M. Nihtila, "Finite pole assignment for systems with time-varying input delays," in *Decision and Control*, 1991, pp. 927–928.

[80] NordIQ, "NordIQ." [Online]. Available: https://nordiq.se/

[81] of the environment and D. energy Australia, *Guide to Best Practice Maintenance and Operation of HVAC Systems for Energy Efficiency*, 2017.

[82] C. Olah, A. Mordvintsev, and L. Schubert, "Feature Visualization," *Distill*, 2017.

[83] A. L. Pisello, M. Bobker, and F. Cotana, "A building energy efficiency optimization method by evaluating the effective thermal zones occupancy," *Energies*, 2012.

[84] A. T. Rezvan, N. S. Gharneh, and G. B. Gharehpetian, "Robust optimization of distributed generation investment in buildings," *Energy*, 2012.

[85] J. P. Richard, "Time-delay systems: An overview of some recent advances and open problems," *Automatica*, 2003.

[86] M. Robillart, P. Schalbart, F. Chaplais, and B. Peuportier, "Model reduction and model predictive control of energy-efficient buildings for electrical heating load shifting," *Journal of Process Control*, 2019.

[87] F. Ruelens, S. Iacovella, B. J. Claessens, and R. Belmans, "Learning agent for a heat-pump thermostat with a set-back strategy using model-free reinforcement learning," *Energies*, vol. 8, no. 8, pp. 8300–8318, 2015.

[88] R. Sangi, F. Bünning, J. Fütterer, and D. Müller, "A Platform for the Agent-based Control of HVAC Systems," in *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*, vol. 132. Linköping University Electronic Press, jul 2017, pp. 799–808.

[89] Sauter, "Flexotron." [Online]. Available: https://www.sauter-controls.com/wp-content/uploads/2019/02/757606.pdf

[90] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "TRPO," *ICML*, 2015.

[91] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization (Wiley Series in Probability and Statistics)*, 1992, vol. 156.

[92] G. Serale, M. Fiorentini, A. Capozzoli, D. Bernardini, and A. Bemporad, "Model Predictive Control (MPC) for enhancing building and HVAC system energy efficiency: Problem formulation, applications and opportunities," *Energies*, 2018.

[93] P. H. Shaikh, N. B. M. Nor, P. Nallagownden, I. Elamvazuthi, and T. Ibrahim, "A review on optimized control systems for building energy and comfort management of smart sustainable buildings," pp. 409–429, 2014.

[94] A. Sharma, "Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 - A strategy for system predictor identification," *Journal of Hydrology*, vol. 239, no. 1-4, pp. 232–239, 2000.

[95] A. A. Sherstov and P. Stone, "Function Approximation via Tile Coding: Automating Parameter Choice," 2005.

[96] ShutterStock, "ShutterStock." [Online]. Available: https://www.shutterstock.com

[97] J. Široký, F. Oldewurtel, J. Cigler, and S. Prívara, "Experimental analysis of model predictive control for an energy efficient building heating system," *Applied Energy*, vol. 88, no. 9, pp. 3079–3087, 2011.

[98] O. J. M. Smith, "A controller to overcome dead time," *ISA Journal*, 1959.

[99] Z. Song, R. Parr, X. Liao, and L. Carin, "Linear Feature Encoding for Reinforcement Learning," *Advances in Neural Information Processing Systems*, 2016.

[100] J. Stoustrup, "Plug & Play Control: Control Technology Towards New Challenges," *European Journal of Control*, vol. 15, no. 3-4, pp. 311–330, jan 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0947358009709914

[101] R. S. Sutton and A. G. Barto, "Reinforcement learning: an introduction 2018 complete draft," *UCL,Computer Science Department, Reinforcement Learning Lectures*, p. 1054, 2017.

[102] R. S. Sutton and C. Szepesv, "A Convergent O(n) Temporal-difference Algorithm for Off-policy Learning with Linear Function Approximation," *Computing*, 2009.

[103] R. Sutton and A. Barto, "Reinforcement Learning: An Introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, 1998. [Online]. Available: http://ieeexplore.ieee.org/document/712192/

[104] R. R. S. Sutton, "Generalization in Reinforcement Learning : Successful Examples Using Sparse Coarse Coding," *Advances in Neural Information Processing Systems*, 1996.

[105] Y. T. T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez and D. Silver, D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv:1509.02971*.

[106] G. Tesauro, "Practical Issues in Temporal Difference Learning," *Machine Learning*, 1992.

[107] B. Ulanicki, J. Kahler, and B. Coulbeck, "Modeling the Efficiency and Power Characteristics of a Pump Group," *Journal of Water Resources Planning and Management*, 2007.

[108] United Nations - Environment Programme, "Sustainable buildings." [Online]. Available: https://www.unenvironment.org/explore-topics/resource-efficiency/what-we-do/cities/sustainable-buildings

[109] D. Urieli and P. Stone, "A Learning Agent for Heat-Pump Thermostat Control," in *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, no. May, 2013, pp. 1093–1100. [Online]. Available: http://dl.acm.org/citation.cfm?id=2484920.2485092

[110] H. van Seijen and R. Sutton, "True Online TD(λ)," *Icml*, 2014.

[111] S. Venkadesh, G. Hoogenboom, W. Potter, and R. McClendon, "A genetic algorithm to refine input data selection for air temperature prediction using artificial neural networks," *Applied Soft Computing Journal*, 2013.

[112] M. P. Wand and M. C. Jones, "Comparison of smoothing parameterizations in bivariate kernel density estimation." *Journal of the American Statistical Association*, vol. 88, no. 422, pp. 520–528, 1993. [Online]. Available: http://www.jstor.org/stable/2290332

[113] C. J. C. H. Watkins, "Learning from Delayed Rewards," *Ph.D. thesis, Cambridge University*, 1989.

[114] T. Wei, Y. Wang, and Q. Zhu, "Deep Reinforcement Learning for Building HVAC Control," in *Proceedings of the 54th Annual Design Automation Conference 2017 on - DAC '17*, 2017, pp. 1–6. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3061639.3062224

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

[116] Z. Y. Wu, M. Tryby, E. Todini, and T. Walski, "Modeling variable-speed pump operations for target hydraulic characteristics," *Journal / American Water Works Association*, 2009.

[117] P. A. Yan Duan, Xi Chen, Rein Houthooft, John Schulman, "Benchmarking Deep Reinforcement Learning for Continuous Control." [Online]. Available: http://proceedings.mlr.press/v48/duan16.pdf

[118] L. Yang, M. Shi, Q. Zheng, W. Meng, and G. Pan, "A unified approach for multi-step temporal-difference learning with eligibility traces in reinforcement learning," in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2018-July, 2018, pp. 2984–2990.

[119] J. Zhang, H. Zhang, B. Wang, and T. Cai, "Nearly data-based optimal control for linear discrete model-free systems with delays via reinforcement learning," *International Journal of Systems Science*, 2016.

Part II

Papers

Paper A

Input Selection for Return Temperature Estimation in Mixing Loops using Partial Mutual Information with Flow Variable Delay

Anders Overgaard
Carsten Skovmose Kallesøe
Jan Dimon Bendtsen
Brian Kongsgaard Nielsen

The paper has been published in the
Proceedings of 2017 IEEE Conference on Control Technology and Applications (CCTA), pp. 1372–1377, 2017.

Abstract

In hydronic heating systems for buildings a mixing loop is often used to control the temperature and pressure. An important task of a mixing loop is to control or constrain the return temperature since this leads to energy savings by reducing heat loss and energy consumed by the pump. With increased access to data, it is desirable to create a data driven model for control. Due to the abundance of data available a method for input variable selection (IVS) is used called partial mutual information (PMI). The paper introduces a method to include flow variable delay into the PMI framework. Data from an office building in Bjerringbro, Denmark is used for the analysis. It is shown that mutual information and performance of a generalized regression neural network (GRNN) is improved by using flow variable delay compared to constant delay.

1 Introduction

High energy savings can be achieved in district heating systems by proper utilization of the heating water at the individual user. Return temperature indicates if the heating water is properply utilized. It has been shown, for a district heating system in Sweden, that given a reduction of 10 K on the return temperature resulted in heat loss reduction of 9.2% and pump energy consumption by 56% [11]. Mixing loops provides a way to ensure proper utilization of the heating water. Classically control or constraint of the return temperature is done with conservative feedback controllers, due to long flow variable delays. With the increase in available building data the option for better models of the return temperature arises, which in turn may be used in a model predictive control scheme.

Work has been done into creating thermal models for buildings that can be used in model predictive schemes, for a review on this topic see [1]. Different approaches to modelling building thermal systems can be applied. One approach is using grey box models with system identification as in [2]. Another approach is machine learning where the models take on a black box formulation as in [10]. Mixing loops are installed in many different types of buildings with different pipe networks and availability of sensor data. This opens up for the interest in a machine learning approach. With a vast amount of data available a question arises of which data points to use. Input variable selection (IVS) covers the area of selecting the inputs from a large set of inputs giving the optimal model and is covered in the review [8]. Partial mutual information (PMI) is an IVS method proposed in [13] and has later been used with success in e.g. [9] and [6].

As mentioned a challenge of mixing loop state estimation and control is flow variable delay. In the framework of PMI it comes naturally to select

the input at the delay that provides the highest mutual information, but this gives a constant delay. In [7] time delay estimation was done for nonlinear system using mutual information. This paper proposes a method for IVS for return temperature estimation using PMI with variable flow delay.

The paper starts by, in Section II, representing the preliminaries, mainly the PMI method. The concept of a mixing loops is provided in Section III. In Section IV the proposed method introducing flow variable delay into the PMI framework is described. Results based on experimental data are in Section V. The paper ends with some concluding remarks in Section VI.

2 Preliminaries

IVS is a group of methods that deals with the problem of finding the optimal set of input variables to give the best prediction.

Given the system

$$Y = h(\mathbf{X}), \tag{A.1}$$

where Y is the output of interest, X is a vector of stochastic variables, the inputs. As stated in [3]; If C is the full set of available input variables, choosing the k input variables from C called X that leads to an optimal model h may be done via IVS methods.

It has been argued [8] that various black-box models such as Artificial Neural Networks (ANN) has the capability of only using inputs that are good predictors and applying low weights to redundant or noisy input variables. So why use IVS? In [8] multiple drawbacks of not using IVS are mentioned. The obvious ones are computational effort, due to a large number of inputs, and curse of dimensionality that increases the model domain exponentially with the number of inputs [4].

Many methods for IVS have been developed. For a good review of these see [8]. In this work the method PMI is used as it is a filter method applicable to nonlinear systems. Filter method means finding the input variables without an exhaustive or heuristic search through training of models.

PMI was proposed in [13]. The method is an iterative method where in each step the input variable with highest mutual information to the output is selected. The mutual information is then removed from the system and the next input variable having highest mutual information with the residual is chosen and so forth.

Mutual information between two continuous random variables is defined as

$$I(x;y) = \int\int p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dxdy, \tag{A.2}$$

where $p(x), p(y)$ is the marginal probability density functions and $p(x, y)$ is the joint probability density function. In the case of independent variables $p(x, y) = p(x)p(y)$, which means that the fraction $\frac{p(x,y)}{p(x)p(y)}$ becomes 1 and the mutual information 0. The mutual information is a measurement of how much uncertainty about y is removed by knowing x. In many applications the underlying probability density functions are not known, and they are instead estimated from samples. The approximation of mutual information used here is [5]:

$$I(x;y) \approx \frac{1}{n}\sum_{i=1}^{n} \log\left(\frac{f(x_i, y_i)}{f(x_i)f(y_i)}\right) \tag{A.3}$$

Here f denotes the estimated probability density from n samples of x and y.

A kernel density estimation is used as in [12]. The Parzen window forms an estimator, here given for the joint density estimation

$$\hat{f}(x, y) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}\left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \end{bmatrix}\right) \tag{A.4}$$

$K_{\mathbf{H}}$ is a kernel function. Here the Gaussian kernel is often used ([9], [6]). In ([17], [12]) it is shown that the bandwidth has the largest impact on accuracy. The Gaussian kernel used in this work is

$$K_{\mathbf{H}}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^m |\mathbf{H}|}} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{H}^{-1}\mathbf{x}\right) \tag{A.5}$$

Here m is the dimension of $\mathbf{X}$ and $\mathbf{H}$ is the bandwidth matrix. The off diagonal terms of the bandwidth matrix adjust the orientation of the joint probability density function while the diagonal terms determines the shape [16]. In the bivariate case the bandwidth matrix that is often used in the case of standardised data is

$$\mathbf{H} = h^2 \begin{bmatrix} S_x^2 & S_{xy} \\ S_{xy} & S_y^2 \end{bmatrix}, \tag{A.6}$$

where S_x^2 and S_y^2 is the sample variance of x and y [16]. S_{xy} is the covariance between x and y. The bandwidth used in this work is the gaussian reference bandwidth [14]

$$h = \left(\frac{1}{m+2}\right)^{\frac{1}{m+4}} \sigma n^{\frac{-1}{m+4}}, \tag{A.7}$$

where σ is the standard deviation of the sample data.

PMI can now be explained as the remaining mutual information between x and y when z is already given. This gives the opportunity to iteratively find the input with highest mutual information, select this input and then remove

the information given by this input and start over until all inputs containing information has been chosen.

To remove the information given by a chosen input from the output and the remaining inputs an estimation is needed. This can be made in many ways. A generalized regression neural network (GRNN) [15] is often used ([9], [6]), in which the function that it is build around is

$$\hat{y}(\mathbf{x}) = \frac{\sum_{i=1}^{n} y_i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{n} \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}, \tag{A.8}$$

where

$$D_i^2 = (\mathbf{x} - \mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i) \tag{A.9}$$

Here y_i and $\mathbf{x}_i$ are the sampled training data for one output and multiple inputs. $\mathbf{x}$ is the input values the estimation of $\hat{y}$ is desired at. σ is called the spread variable and determines the smoothness of the estimated probability densities. This regression method is used to determine estimates of the output and the other inputs given the chosen input. This information is then removed from the data and the mutual information analysis can subsequently be done on the residuals.

$$\begin{aligned} u &= y - E[y|z] \\ \mathbf{v} &= \mathbf{x} - E[\mathbf{x}|z] \end{aligned} \tag{A.10}$$

Now the PMI can be found for the remaining inputs as

$$I(\mathbf{x}; y|z) = I(\mathbf{v}; u) \tag{A.11}$$

Choosing the right input variable is not only a question of which variable, but also the time delay

$$\max_{j,k} \quad I\left(\mathbf{x}_{i-k}^{j}; y_i\right), \tag{A.12}$$

where k is the delay and j is the index of the input in the full set of inputs.

At some point choosing further inputs will not improve the model. Here cross validation is used as stop criteria.

3 Application

Heating water for space heating radiators, floor heating, heating coils etc. are supplied via mixing loops, see Fig. A.1. Hereby the control of the pressure and temperature of the heating water at the consumer is independent of the supply. There are two control variables in the mixing loop. The pump speed that controls the differential pressure (dp) and the opening of the control

valve that controls the supply temperature (T_s). The control valve determines how much of the supply water is allowed to mix with the return water, such that the temperature of the water going forward is controllable. The differential pressure and temperature can control how much heat power that goes into the system. If the heat power is too low, the heating needs of the building will not be met. If the heat power is too high the thermostatic valves will close and excess pump energy will be used. Having a high return temperature is a problem due to pipe loss and unnecessary pump energy usage. In this paper the return temperature is described as

$$T_r(t) = h\left(\mathbf{T}_s, \mathbf{q}, \mathbf{T}_z\right), \tag{A.13}$$

where

$$\begin{aligned}
\mathbf{T}_s &= \left[T_s\left(t - \frac{V_1}{q_1}\right), ..., T_s\left(t - \frac{V_n}{q_n}\right)\right] \\
\mathbf{q} &= [q_1, ..., q_n] \\
\mathbf{T}_z &= \left[T_{z1}\left(t - \frac{V_1}{2q_1}\right), ..., T_{zn}\left(t - \frac{V_n}{2q_n}\right)\right]
\end{aligned}$$

Here n is number of pipes routes the water can take. $\mathbf{T}_s$ is a vector of the supply temperature at different times. $\mathbf{q}$ is a vector of the flow in the different pipe routes. V is a vector of the volumes in the different pipe routes. $\mathbf{T}_z$ is the temperature in the different zones supplied by the pipe routes. Notice that the flow is here considered quasi static meaning that it is constant within the variable time frame V_n/q_n. The zone temperatures can described via the differential equation

$$\dot{T}_{zn}(t) = g\left(T_{zn}(t), T_s\left(t - \frac{V_n}{2q_n}\right), q_n, \Phi(t)\right), \tag{A.14}$$

where Φ is a set of all the disturbances that act on the zone temperature. Some of these are outside temperature T_o, wind speed W_s, solar radiation S_o, heat flow from adjacent zones, people located in the zone and the ventilation system. The return temperature is notoriously hard to control due to the following

- Multiple heat sources connected via multiple pipes acting on the return temperature at different delays.
- Flow variable delay
- Unknown disturbances acting on the zone temperature determining the cooling of the hot water.

The data used for proving the effect of the proposed method is gathered from an office building located in Bjerringbro in Denmark during January

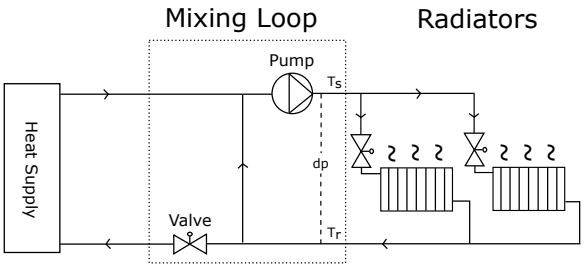

Fig. A.1: Sketch of Mixing Loop

Fig. A.2: Office building used for gathering data. Heating supplied via single mixing loop.

2017, see Fig. A.2. The office building consists of 34 radiator zones divided over 3 floors. There are multiple radiators in a zone, but they are being controlled by a single controller from a single temperature sensor. The building contains multiple sensors and 143 of these are logged and used for this work. The sensors can be seen in table A.1 in the results section.

4 Methodology

In PMI the different inputs are analysed at a set of constant delays. However, in this application the delay depends heavily on the flow, which vary over time. This paper propose a method for incorporating this into the PMI framework.

Given a one pipe system with constant flow and no energy loss the relation between T_s and T_r can be written as

$$T_r(t) = T_s(t - d) \tag{A.15}$$

Using mutual information the delay d may then be found as

$$\max_k \quad I\left(T_s(i-k); T_r(i)\right), \tag{A.16}$$

where the k that gives highest mutual information would be the k that makes δt_k closest to the delay d.

Since the flow varies, using a constant delay will result in an error which increases the further the flow is from the design point. Introduce a flow dependent delay makes it possible to follow the variations in the delay under a quasi static flow assumption. Quasi static meaning that the flow only changes a little during the delay time. Given one pipe the same relationship with flow scaling delay is described as

$$T_r(t) = T_s\left(t - \frac{V}{q}\right), \tag{A.17}$$

where q is the quasi stationary flow and V is the volume of the pipe. Instead of finding a constant delay a volume can now be found giving the highest mutual information

$$\max_V \quad I\left(T_s\left(i - \frac{V}{q}\right); T_r(i)\right) \tag{A.18}$$

For systems with more pipes the relation extends to

$$T_r(t) = h\left(\mathbf{T}_s, \mathbf{q}\right), \tag{A.19}$$

where

$$\mathbf{T}_s = \left[T_s\left(t - \frac{V_1}{q_1}\right), ..., T_s\left(t - \frac{V_n}{q_n}\right)\right]$$
$$\mathbf{q} = [q_1, ..., q_n]$$

Only the total flow q is known and not how it is distributed into specific pipe flows q_1 and q_2. To relate the specific pipe flows to the overall flows a ratio β is introduced

$$\begin{aligned} \beta = \sum_{n=1}^{p} \beta_n = 1 \\ q_n = \beta_n q \end{aligned} \tag{A.20}$$

The following parameter α is now defined as

$$\alpha_n = \frac{V_n}{\beta_n}, \tag{A.21}$$

and inserting this into the vector of supply temperatures gives

$$\mathbf{T}_s = \left[T_s\left(t - \frac{\alpha_1}{q}\right), ..., T_s\left(t - \frac{\alpha_n}{q}\right)\right] \tag{A.22}$$

T_s will not only act through different pipes impacting T_r at different delays, but due to the heat consumption and flow being different in each pipe it

will also have different impact in scale. Using the PMI method the different α_n can be found one at a time from highest impact to lowest.

$$\max_{\alpha_1} \quad I\left(T_s\left(i-\frac{\alpha_1}{q}\right);T_r(i)\right) \tag{A.23}$$

$$\begin{aligned} u &= T_r - E[T_r|T_{s\alpha_1}] \\ z &= T_s - E[T_s|T_{s\alpha_1}], \end{aligned} \tag{A.24}$$

where $T_{s\alpha_1}$ is the time series delayed by the flow variable delay using α_1

$$\max_{\alpha_2} \quad I\left(z\left(i-\frac{\alpha_2}{q}\right);u(i)\right) \tag{A.25}$$

To show that the PMI method with variable flow has the ability to find the α_n a numerical simulation of equation A.19 is done for a system with ($\alpha_1 = 0.02$,$\alpha_2 = 0.04$,$\alpha_3 = 0.12$). The simulation is done for a week where the flow changes in a quasi static manner, see Fig. A.3. The supply temperature follows a pulse signal with a period of 4 hours going from 345 K to 330 K and back up again. In Fig. A.4 one of the edges of this pulse and the response on the return temperature is shown. Fig. A.5 shows that the maximum mutual information for each iteration of the PMI with variable flow method peaks out the correct α values one by one. In theory this continues until α_n is found for all the pipes. In practise the mutual information level becomes too small to be used in the prediction model, meaning only some flow dependent delays are used. The method can also be used for the other inputs that acts upon the system. Take for example a zone temperature, which would affect the system as seen in Equation A.13. In this case there would be no information of which α_n, found from T_s, that matches the specific zone. Instead a new α that maximizes mutual information between zone temperature and return temperature is found using the same method. This also has the added benefit that the pipe length does not have to be the same for the supply and return pipe.

It is important to recall that α contained the ratio of the total flow that runs in the specific pipe. If the ratio changes α also changes which still poses as a source of error. Another small approximation error occurs due to the discretized sampling. This means that not all values for α/q can be chosen. A quantization function is used

$$Q\left(\frac{\alpha}{q}\right) \in \mathbb{Z}_{\geq 0}, \tag{A.26}$$

where $Q : \mathbb{R} \to \mathbb{Z}$

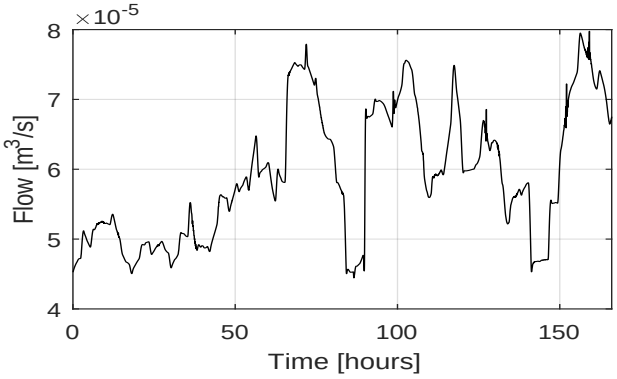

Fig. A.3: Flow input to simulation. Changes over whole simulation period, but is quasi static within the delay times.

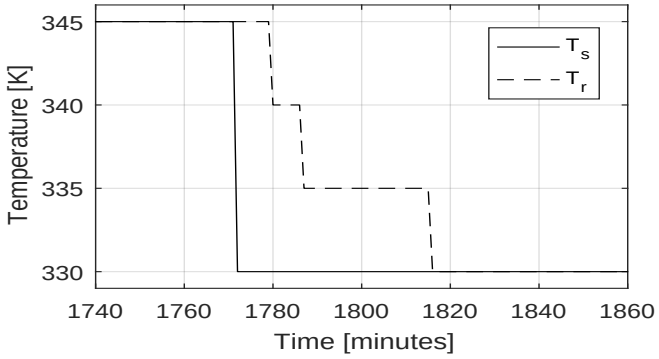

Fig. A.4: One edge of the pulses given in the simulation. The delays are a function of the flow given at that time.

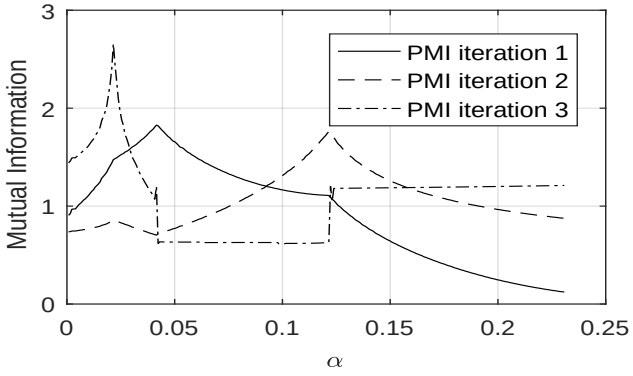

Fig. A.5: Mutual Information between T_s and T_r as a function of α_{T_s}

5 Results & discussion

The data which is used for the PMI with flow variable delay analysis is data from 6 days with sampling interval of 1 minute. Many input variables are searched for mutual information, but to give an example of the sampled data, T_s and T_r during 5 hours is plotted in Fig. A.6.

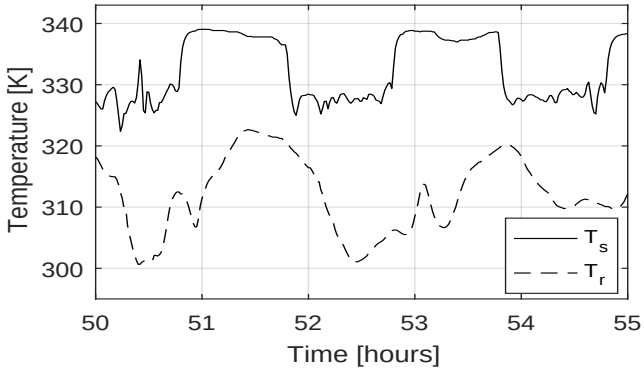

Fig. A.6: T_s and T_r plotted for 5 hours out of the 6 days of data.

To improve persistence of excitation the set point for the control of T_s is set to a pulse with a period of 2 hours and amplitude of 10 K.

To illustrate mutual information of the system all inputs at the delays which represents highest mutual information can be seen in Fig. A.7.

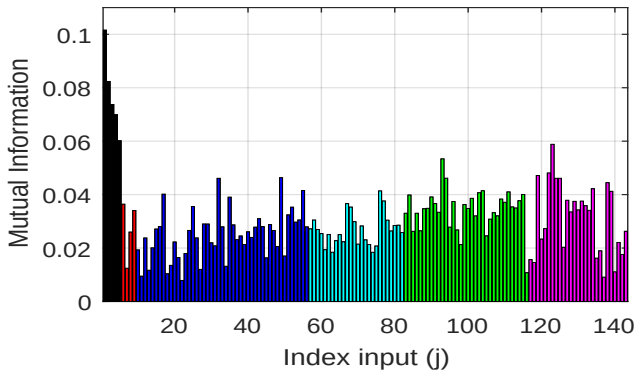

Fig. A.7: Mutual Information as a function of input index. For each input, α is chosen giving maximum mutual information. See Table A.1 for input indexes.

It is only the first iteration of the PMI with flow variable delay that is plotted, so notice that other inputs can contain higher mutual information

with the residuals in later iterations. In Table A.1 the index of the input variables is given.

Index (j)	Input Description	Name
1	Supply Temperature	(T_s)
2	Diff. pressure	(dp)
3	Primary Flow	(q_p)
4	Mixing Valve Opening Degree	(OD_{mv})
5	Heat Power Mixing Loop	(P_m)
6	Outside Temperature	(T_o)
7	Solar Radiation	(S_o)
8	Wind Direction	(Wd)
9	Wind Speed	(Ws)
10-13	Heat Power Ventilation Systems	(H_{v1-4})
14-18	Ventilation Air Temperature	(T_{v1-5})
19-51	Ventilation Ducts Opening Degrees	(OD_{d1-33})
52-56	Ventilation Fan Speeds	(VAV_{1-5})
57-82	CO_2 level in zones	(C_{1-26})
83-116	Zone Temperatures	(T_{s1-34})
117-143	Radiator Valve Opening Degrees	(OD_{r1-27})

Table A.1: Inputs indexes.

Fig. A.7, shows that the mutual information is highest at the control variables T_s and dp. The maximum mutual information in Fig. A.7 for each input is found as shown for three of the inputs in Fig. A.8.

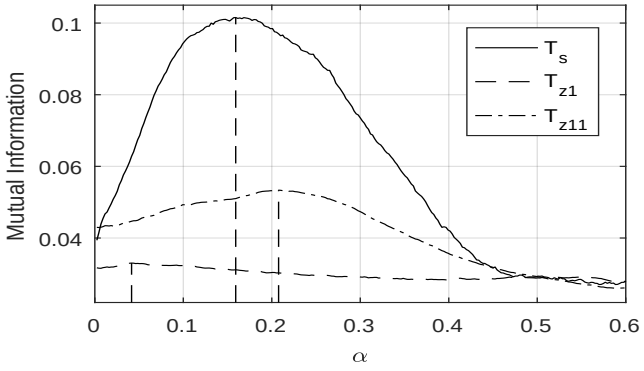

Fig. A.8: Mutual Information between T_r and the three inputs T_s,T_{z1} and T_{z11} as a function of α

Here a search of maximum mutual information as a function of α is done

for T_s, T_{z1} and T_{z11}. The dotted vertical lines indicate the maximum mutual information for each input. Apart from the different value of mutual information it is also interesting to observe the different α where maximum appear at, which illustrates the different pipe lengths and flow ratio that the zones are subject to. In regard to T_s the α giving the maximum mutual information is $\alpha_{T_s} = 0.16$. The method uses flow variable delay which means that each sample a new delay is calculated according to the flow. To give an idea of the delay times the mean delay with $\alpha_{T_s} = 0.16$ is calculated to $\mu(\alpha/q) = 18\ minutes$. In Table A.2 the mutual information is given for constant delay and flow variable delay at the delays that gives highest mutual information. It shows that flow variable delay gives highest mutual information in the case of the three example inputs.

Model/Input	T_s	T_{z1}	T_{z11}]
Constant Delay	0.091	0.032	0.047
Flow Variable Delay	0.102	0.034	0.054
Improvement	12%	6%	15%

Table A.2: Max. Mutual Information using Constant or Variable Delay

To make a comparison between constant and flow variable delay the ability to model T_r for both methods is tested. The PMI method with the suggested flow variable delay was used to pick a set of input variables. GRNN models was made for different dimensions of the input set. Cross validation was done on these models to choose the dimension of the input set use in the final model. In Fig. A.9, the Root Mean Square Error (RMSE) of the cross validation data is plotted as a function of used input variables.

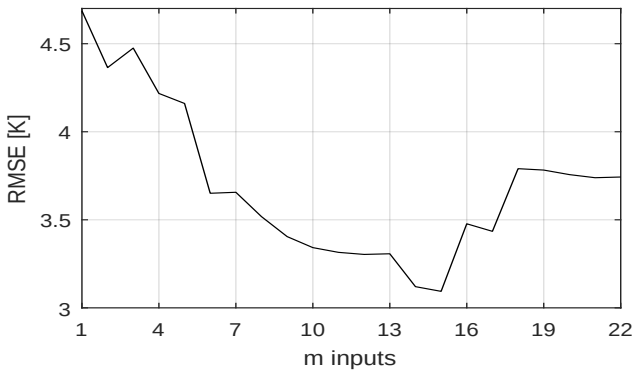

Fig. A.9: Choosing number of inputs (m) by cross validation.

From this, the number of inputs providing the lowest RMSE was chosen, which is the first 15 inputs in the set. The chosen inputs can be seen in table A.3.

Selected	Input	Selected	Input	Selected	Input
1.	T_s	6.	T_{z12}	11.	T_o
2.	dp	7.	OD_{d32}	12.	C_{15}
3.	OD_{r12}	8.	OD_{r11}	13.	Ws
4.	T_{z21}	9.	OD_{mv}	14.	T_{z14}
5.	P_m	10.	q_p	15.	T_{z6}

Table A.3: Chosen Inputs

In Fig. A.10, the estimation on cross validation data using a GRNN with the chosen input dimension can be seen.

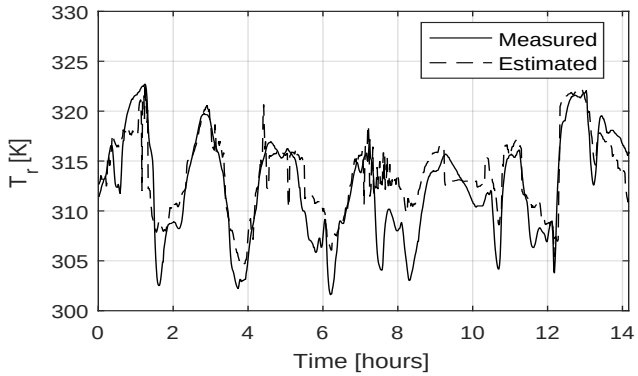

Fig. A.10: Cross Validation.

In table A.4 the RMSE is compared between three models. The flow variable delay model where the prediction horizon changes with flow according to the chosen α_{T_s}. The next model is a constant delay model where the same inputs are used, but at constant delays, giving a constant prediction horizon. The prediction horizon is the delay at where T_s holds most mutual information. This is chosen because it would be a natural control horizon in a model predictive control scheme. The final model (baseline) is a simple first order autoregressive model where the time delay is constant and the prediction horizon is the same as the constant delay model. The baseline is added to relate to the simplest model where the prediction is equal to the present measurement. It is shown that the model containing variable flow delay performs better than using constant delay.

Model	RMSE [K]
Flow Dependent Delay	3.09
Constant Delay	4.01
Baseline	5.69

Table A.4: Comparison of models

6 Conclusion

The motivation for this work is to improve the already existing PMI method to improve IVS and thereby estimation of the return temperature in mixing loops. This was achieved by adding a flow variable delay to the framework. It is shown on measured data that this increases the mutual information between input and output variables compared to using constant delay. Using flow variable delay also leads to an increased performance in terms of RMSE when applied to a GRNN model. Further work needs to be done into analysing when persistence of excitation is reached in a given dataset. The curse of dimensionality is also a concern, where the quantity of the data puts a limit of the dimension of chosen inputs.

References

[1] E. Atam and L. Helsen, "Control-Oriented Thermal Modeling of Multizone Buildings: Methods and Issues: Intelligent Control of a Building System," *IEEE Control Systems*, vol. 36, no. 3, pp. 86–111, 2016.

[2] P. Bacher and H. Madsen, "Identifying suitable models for the heat dynamics of buildings," *Energy and Buildings*, vol. 43, no. 7, pp. 1511–1522, 2011.

[3] R. Battiti, "Using Mutual Information for Selecting Features in Supervised Neural-Net Learning," *Ieee Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, 1994.

[4] R. Bellman, "Adaptive control processes: A guided tour," *Princeton University Press*, vol. 28, pp. 1–19, 1961.

[5] S. Gao, G. V. Steeg, and A. Galstyan, "Estimating Mutual Information by Local Gaussian Approximation," *Proceedings of the 31st Conference on Uncertainties in Artificial Intelligence*, p. 224, 2015.

[6] X. Li, H. R. Maier, and A. C. Zecchin, "Improved PMI-based input variable selection approach for artificial neural network and other data driven environmental and water resource models," *Environmental Modelling & Software*, vol. 65, pp. 15–29, 2015.

[7] N. J. I. Mars and G. W. van Arragon, "Time Delay Estiamtion in Non-Linear Systems using Average Amount of Mutual Information Analysis," vol. 4, pp. 139–153, 1981.

[8] R. May, G. Dandy, and H. Maier, "Review of Input Variable Selection Methods for Artificial Neural Networks," *Artificial Neural Networks - Methodological Advances and Biomedical Applications*, no. August 2016, p. 362, 2011.

[9] R. J. May, H. R. Maier, G. C. Dandy, and T. G. Fernando, "Non-linear variable selection for artificial neural networks using partial mutual information," *Environmental Modelling & Software*, vol. 23, no. 10-11, pp. 1312–1326, oct 2008.

[10] N. Morel, M. Bauer, M. El-Khoury, and J. Krauss, "Neurobat, a Predictive and Adaptive Heating Control System Using Artificial Neural Networks," *International Journal of Solar Energy*, vol. 21, no. 2-3, pp. 161–201, 2001.

[11] R. Sallent Cuadrado, "Return temperature influence of a district heating network on the CHP plant production costs," no. June, p. 60, 2009.

[12] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization (Wiley Series in Probability and Statistics)*, 1992, vol. 156.

[13] A. Sharma, "Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 - A strategy for system predictor identification," *Journal of Hydrology*, vol. 239, no. 1-4, pp. 232–239, 2000.

[14] B. Silverman, "Density estimation for statistics and data analysis," *Chapman and Hall*, no. 1, pp. 1–22.

[15] D. F. Specht, "A general regression neural network," *Neural Networks, IEEE Transactions on*, vol. 2, no. 6, pp. 568–576, 1991.

[16] M. P. Wand and M. C. Jones, "Comparison of smoothing parameterizations in bivariate kernel density estimation." *Journal of the American Statistical Association*, vol. 88, no. 422, pp. 520–528, 1993.

[17] ——, "Kernel Smoothing," *Encyclopedia of Statistics in Behavioral Science*, vol. 60, no. 60, p. 212, 1995.

Paper B

Mixing Loop Control using Reinforcement Learning

Anders Overgaard
Carsten Skovmose Kallesøe
Jan Dimon Bendtsen
Brian Kongsgaard Nielsen

The paper has been published in the
CLIMA 2019 REHVA HVAC World Congress,
E3S Web of Conferences Vol. 111, 2019

The layout has been revised.

Abstract

In hydronic heating systems, a mixing loop is used to control the temperature and pressure. The task of the mixing loop is to provide enough heat power for comfort while minimizing the cost of heating the building. Control strategies for mixing loops are often limited by the fact that they are installed in a wide range of different buildings and locations without being properly tuned. To solve this problem the reinforcement learning method known as Q-learning is investigated. To improve the convergence rate this paper introduces a Gaussian kernel backup method and a generic model for pre-simulation. The method is tested via high-fidelity simulation of different types of residential buildings located in Copenhagen. It is shown that the proposed method performs better than well tuned industrial controllers.

1 Introduction

In Europe buildings account for 40% of the total energy usage. In the residential sector space heating accounts for 66% of the building energy consumption [6]. It is predicted that scheduling and improved control can lead to savings of 11-16% [2]. This huge savings potential is the reason that building control keeps being an active research area, see reviews [14] and [4]. In this work the focus is on building heating via mixing loops. Mixing loops are used to ensure proper comfort, heat power utilization and energy savings in buildings. Low heat power utilization leads to low efficiency in the supply coming from district heating. It has been shown that lowering the return temperature by 10^oC gave a heat loss reduction of 9.2% and pump energy reduction by 56% at the district heating plant [13]. So why is this important for the end user? The district heating plants are starting to enforce proper heat water cooling through added fees on a high return temperature. Ensuring proper heat power utilization in the control of the mixing loop can therefore also help reduce the end costumers cost of heating the building.

A lot of research has been done on optimal building thermal control, often in the form of Model Predictive Control (MPC). Examples of this are [11] and [16]. Here large savings was shown by using an MPC compared to traditional control strategies. The disadvantage of MPC is the reliance on accurate models of the building, especially when the product is installed into many different buildings. Different methods for identifying models of the building using data for MPC has been been explored. In [1] artificial neural networks are used for building the model for MPC, while in [3] subspace methods are used. In this work an alternative approach for learning optimal control through data will be investigated by using reinforcement learning to control a mixing loop. The result in [5] show that reinforcement learning is competitive with an MPC on a power system even when a good model is available.

Even though Reinforcement Learning has been around for a long time, recent results have increased its popularity. This attention is mainly brought on by the Reinforcement Learning algorithm AlphaGo's ability to learn, tabula rasa, how to beat the world champion of the game Go [15]. Reinforcement learning has also been tried on HVAC applications. In [8] reinforcement learning was used to control passive and active thermal storage. Simulated reinforcement learning was used where the controller is getting priori knowledge from simulation. The result in [18] showed savings in heat-pump thermostat control by using reinforcement learning. In [12] a batch reinforcement learning method was used to control a heat-pump.

In reinforcement learning the rate of convergence towards optimal control is an issue, since it often requires a lot of training. In this work a Gaussian kernel backup rule is suggested to improve initial convergence in tabular Q-learning. Kernel based methods have been used in reinforcement learning, but mostly in regards to function approximation methods such as in [9].

The paper starts with an introduction to Reinforcement Learning in Section 2. The concept of building heat supply via a mixing loop is provided in Section 3. In Section 4 the proposed method using Gaussian kernel backup in Q-learning is presented. Section 5 explains the simulation setup. The results are presented and discussed in Section 6. The paper ends with the concluding remarks in Section 7.

2 Preliminaries

In this section reinforcement learning will be introduced. For a more thorough description see [17]. In Fig. B.1 is a general reinforcement learning setup where an agent interacts with an environment.

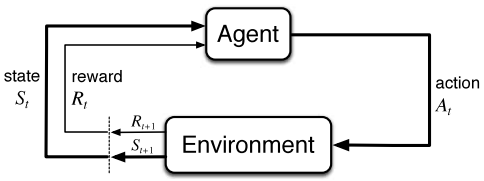

Fig. B.1: Agent-environment interaction [17].

The environment is in a state S_t at time t. "States" is here meant as all the information the agent receives about the environment. The environment also sends out a reward determining the instantaneous value of being in this state. The agent seeks to maximize the cumulative reward called the return [17]

$$G \doteq \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1} \tag{B.1}$$

where γ is the discount factor that lies in the interval $0 \leq \gamma \leq 1$. A higher discount factor will cause the agent to strive for longer term return, but will also increase the convergence rate of the learning agent. T is the final time step. For episodic task this is the end time, but for continuing tasks $T = \infty$. Having both $\gamma = 1$ and $T = \infty$ is not feasible as this would lead to infinite return.

The agent uses a policy, π_t, which goal is to maximize the return. The policy maps the states to an action, hence it is similar to a control law. The mapping can be of stochastic nature or deterministic.

The next element of Reinforcement learning is the value function [17]

$$V_\pi(s) \doteq \mathbb{E}\left[G_t | S_t = s\right] \tag{B.2}$$

The function describes that if starting in state s and continuing to follow policy π, the expected return will be G_t.

By adding onto the value function we get the state-action value function

$$Q_\pi(s, a) \doteq \mathbb{E}\left[G_t | S_t = s, A_t = a\right] \tag{B.3}$$

Which describes the expected return of being in state s, taking action a and afterwards follow policy π.

The goal in reinforcement learning is to find the optimal policy. This is often done through policy iteration by alternating between evaluating V_π using π and improving π using V_π. A greedy policy is a policy that always chooses the action which yields the highest return and is defined as

$$\pi_g(s) \doteq \arg\max_a q\left(s, a\right) \tag{B.4}$$

Such a policy fully exploits the current state-action value function, but the downside is that it does not explore and perhaps updates the state-action value function in such a way that the policy can be improved. This is the recurring problem of exploitation versus exploration. Proofs of convergence towards optimality often relies on exploration for reinforcement learning methods. So both exploitation and exploration needs to be done. A simple way to achieve that is the ϵ-greedy method. Here the greedy action is chosen with probability $1 - \epsilon$ and the rest of the times a random action is taken to explore.

The last element introduced is the learning rate, α, chosen from $0 < \alpha \leq 1$. This determines how much the newly learned information will override older information when updating the value function. In an environment that is fully deterministic the best learning rate is simply 1. Introducing stochastic behaviour such as noise or disturbances not contained in the states changes this towards supporting a lower learning rate.

3 Building Heat Supply via Mixing Loop

The Mixing Loop application is here described in short. This is done to get an understanding of the system, which is necessary for describing a reward function and choosing states and actions for the Q-learning. A simple model of the application is here described by a building with only one zone with one radiator as seen in Fig. B.2.

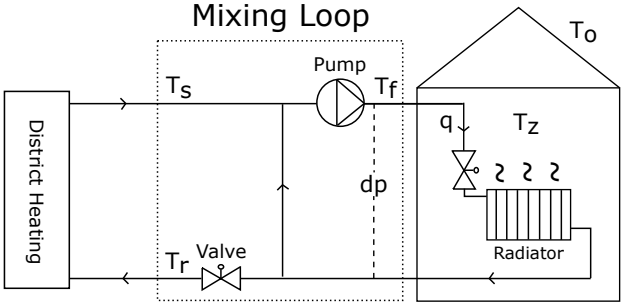

Fig. B.2: Simple schematic of mixing loop application

The zone temperature is controlled by a thermostatic valve. The heat power is supplied via a mixing loop from district heating. The change in zone temperature is here described as the difference between heating, load and disturbance powers

$$C_z \dot{T}_z = \Phi_h + \Phi_L + \Phi_d, \tag{B.5}$$

where C_z is the heat capacity of the zone and T_z is the zone temperature. $\Phi_{h/L/d}$ are the heating, load and disturbance powers. The load power is the cooling acting on the zone from outside the building envelope. The disturbance power is all the remaining power acting on the system, also referred to as free heat. The majority of the disturbance power is created by the occupancy of the house and electric appliances.

The heat power is supplied by a radiator, which is here described as

$$\Phi_h = C_r \left(\frac{T_f + T_r}{2} - T_z \right)^n, \tag{B.6}$$

where C_r is the thermal conductance of the radiator, T_f is the forward water temperature, T_r is the return water temperature and n is a radiator constant. The heat power can also be described via the heating water as

$$\Phi_h = c_w q \left(T_f - T_r \right), \tag{B.7}$$

where c_w is the volumetric heat capacity of the heating water and q is the volume flow rate.

Via these two equations for heat power, the return temperature can be solved for, whereby the dependencies for heat power are

$$\Phi_h = f\left(q, T_f, T_z\right). \tag{B.8}$$

The flow rate

$$q = g\left(u\right)\sqrt{\Delta p}, \tag{B.9}$$

is a function of the thermostatic valve's opening degree u and the differential pressure Δp.

Typically a P controller determines the opening degree of the valve

$$u = K_p\left(T_{ref} - T_z\right), \tag{B.10}$$

Where K_p is the proportional gain and T_{ref} is the reference temperature set by the user.

The mixing loop controls T_f and Δp. By opening the control valve and mixing hot water at temperature T_s with return water having temperature T_r in a ratio that gives the desired T_f, see Fig. B.2. Δp is controlled solely by the pump speed since the mixing loop hydraulically decouples the zone from the supply. The objective is to supply enough heating power for the system to keep the reference temperatures. By looking at (B.8) and (B.9) it can be seen that while the thermostatic valve controls the heat power, T_f and dp influences the gain of the controller. This means that by controlling T_f and dp only the gain of the thermostatic control can be influenced, except for saturation situations which is what is utilized for setback. The objective providing enough heat power has to be kept without increasing the pump pressure too much or increasing the return temperature leading to energy losses in the heat distribution. By (B.5) knowing the load and disturbance power heat power could be controlled as $\Phi_h = \Phi_L + \Phi_d$. The caveat of controlling by balancing the heat load is that if any unaccounted disturbance happens the thermostatic valve will be in saturation and will not be able to reject the disturbance. In mixing loop control it is not desirable to control in ways that eliminates the thermostatic valve's disturbance rejection.

The reward defines the control objective. For heating systems two features are important to optimize: comfort and cost. However, these two features can be described in various ways. For cost it is chosen to include the cost for the pump power, and the cost for the heat power. Other costs that could be included could be the cost of wear and tear of components such as the pump, pipes and valves or commissioning time when installing the HVAC system, but these are not included in this work.

The pump power cost is calculated as

$$\psi_{pump} = \Phi_{pump}\Omega_e, \tag{B.11}$$

where ψ_{pump} is the pump power cost, Φ_{pump} is the pump power consumption and Ω_e is the price of electric power. In this work Ω_e is kept constant at $0.27€/kWh$. If e.g. load shift is desired this should of cause be changed to a time dependant price. In this work the heat source is district heating, where a high return temperature reduces the efficiency, mainly through added heat losses in the distribution network. District heating companies often penalize high return temperatures by increasing the heat power cost as a function of cooling of the heating water. The additional cost is added differently dependent on the district heating company. In this work it is done like the district heating company in Copenhagen, "HOFOR", implements this [7].

$$\psi_{heat} = \Phi_{heat}\Omega_{heat}\eta. \tag{B.12}$$

Here ψ_{heat} is the heat power cost, Φ_{heat} is the heat power used, Ω_{heat} is the base price of the heat power at $88.9€/kWh$, and η is a price correction for cooling of the heating medium calculated as

$$\eta = 1 - \left(\frac{1}{125}(T_s - T_r) + \frac{33}{125}\right) \tag{B.13}$$

This means that the price of the heat power increases 0.8% per ^oC that the cooling of the heating medium is lower than $\Delta 33^oC$.

The heat comfort can be measured in different ways; here, the highest zone temperature error is used

$$e_{max}(t) = \max_{i\in\{1,\cdots,n_z\}} \left|T_{ref} - T_{z,i}(t)\right|, \tag{B.14}$$

where n_z is the number of zones. This ensures the lowest maximum error. Other ways of describing comfort can be the number of times temperatures exceeds a given bound. In this work only night setback is used, but other setback periods can be used via calendar functions or leaning patterns of the inhabitants. The difficult part about night setback is that it is dependent on the specific building. Both how much and for how long the temperature can be changed while ensuring comfort when setback ends varies. Not only from building to building, but also as a function of other states such as outside temperature. When reheating after a setback an optimal reheat "speed" is also important otherwise high return temperature will be imposed due to high flows and forward temperature. High return temperature during reheating is costly since a high amount of heat power is being consumed. Doing this in an optimal fashion should be learned by the reinforcement learning agent.

4 Q-learning with Gaussian Kernel Backup

The reinforcement learning method used here is Q-learning. Q-learning was first described in [19] and is defined by the backup

$$Q\left(S_t, A_t\right) \leftarrow Q\left(S_t, A_t\right) + \alpha\left[R_{t+1} + \gamma \max_a Q\left(S_{t+1}, a\right) - Q\left(S_t, A_t\right)\right] \quad \text{(B.15)}$$

A strength of Q-learning is that it directly finds the value of taking an action in a given state and afterwards following an optimal policy. This makes it model-free as no transition model of the environment is needed. A requirement for convergence towards the optimal policy is that all state-action pairs continue to be visited and updated. The formal proof of convergence can be found in [20]. To ensure convergence the ϵ-greedy method is used with $\epsilon = 0.1$.

Q-learning is here single-step, as seen by the term $[\max_a Q\left(S_{t+1}, a\right)]$, but can be extended to multiple steps. The learning rate α is set to 0.2 and the discount rate to 0.4.

In this work a tabular version of Q-learning is used to ensure convergence. This is feasible when keeping a low dimensionality of the state-action space, **Q**. The state action space used can be seen in Table I.

4.1 Choosing Reward

The reward function of a mixing loop is a multi goal reward system where it seeks to supply the best heat comfort for the building while minimizing cost. When it is deemed that setback can be used, the heat comfort goal vanishes and only the cost remains. The cost that the agent should minimize is the combined cost of the heat and pump power. Due to this multi objective reward a weighting factor, β, is needed, which determines the scaling between improving heat comfort and minimizing cost. In this work $\beta = 0.5$ unless otherwise stated. The reward then becomes

$$R(t) = \begin{cases} -(e_{max}(t)^2 + \beta(\psi_{heat}(t) + \psi_{pump}(t))) & 6 \leq t \ mod(24h) \leq 21 \\ -\beta(\psi_{heat}(t) + \psi_{pump}(t)) & otherwise \end{cases} \quad \text{(B.16)}$$

Here e_{max} is the maximum temperature error out of all the zones, squared to punish larger errors harder. Heat power cost ψ_{heat} and pump power cost ψ_{pump} was described in (B.12) and (B.11). Additionally a soft constraint is added such that low reward is given if any zone temperature goes below 16^oC.

Recall that the reinforcement learning seeks to maximize the cumulative reward. This ensures that an action that decreases power and therefore increases the reward during setback is only good if the building can reach the heat comfort giving high reward when setback is off.

4.2 Choosing States and Actions

As seen in section 3 there are a lot of states that would give added information for the agent. However in this work the focus is on making the minimal state-action space due to working with tabular methods where the state-action space and therefore learning rate suffers greatly from the curse of dimensionality. Another reason for keeping the state space small is the sensors needed for the information. Choosing the states is done by the definition given by [10]:

A state variable is the minimally dimensioned function of history that is necessary and sufficient to compute the decision function, the transition function, and the contribution (here the reward) function.

This selection is here done from the knowledge of the application, but could also have been done via correlation investigation.

States	Size of dimension	Range
Outside Temperature	21	-20 to 20 [^oC]
Time of day	24	1 to 24 [hours]
Actions	**Dimension**	**Range**
Pump Diff. Pressure	5	0 to 0.4 [bar]
Forward Temperature	31	15 to 75 [^oC]

Table B.1: State-Action Space

To ensure the zone temperatures enough heat power should be available for the thermostats. The needed heat power is a product of the load and the free heat, where the load is given by the outside temperature and the free heat given by multiple factors. Due to this T_o was chosen as a state. The free heat is not added explicitly in states in this work to reduce dimensionality, but later work could explore inclusion of indicators such as number of inhabitants present, solar radiation or electric appliances. Time is added as a state as $R(t)$ depends on it. Furthermore time of day can also capture periodic phenomenons, for example if free heat contains daily patterns.

The actions for the mixing loop application are the forward temperature and differential pressure, see section 3. Due to the nature of pumps the pressure is limited at higher flows. In the situation where the set point from the controlling agent is higher than the pump can supply it is set to max. The minimum forward temperature is 15^oC however due to the nature of a mixing the lowest forward temperature that can be supplied is the same as the return temperature at that given time. In the same way the maximum

temperature is only as high as the supply temperature which in this case is controlled to 75^oC. So when choosing a forward temperature the agent can only choose from $T_r(t) <= T_f(t) <= T_s(t)$.

4.3 Gaussian Kernel Backup and pre-simulation

In tabular reinforcement learning using the Q-learning backup rule, the situation can occur where one specific state-action pair has been visited multiple times, but one in vicinity has never been explored. In this case there would be no knowledge of the state in the immediate vicinity since it has never been visited. Due to the priori knowledge of the "smoothness" of the application there must be knowledge to be gained about the optimal action in S_2 from the knowledge about S_1. This comes naturally when using function approximations such as kernel-based methods, but not in the tabular case. To gain increased convergence rate a Gaussian kernel is therefore applied to the backup process. Instead of only doing backup of the one state-action pair, backup is done on all state-action pairs with decreased learning rate the further the state is from the visited state. The learning rates are distributed using a Gaussian kernel. First two indexing vectors are introduced. $\mathbf{x}_t$ is the vector describing the location in the state-action tabular $\mathbf{Q}(S, A)$ at time t. It contains the index for each dimension. $\mathbf{x}$ is the vector describing the location of the state-action pair that is being backlogged to. Both has the dimension $n \times x$, where n is the sum of states and actions, in this case 4.

Now the backup is done to all state-action pairs using the following backup rule

$$Q(\mathbf{x}) \leftarrow Q(\mathbf{x}) + \alpha K_\sigma(\mathbf{x}_t - \mathbf{x}) \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right] \quad \text{(B.17)}$$

Where K_σ is calculated using the Gaussian kernel

$$K_\sigma(\mathbf{x}_d) = \exp\left(-\frac{|\mathbf{x}_d|^2}{2\sigma^2} \right) \quad \text{(B.18)}$$

In this work $\sigma = 1$ and is lowered as time passes. As σ decreases the method will converge to classical Q-learning. In Fig. B.3 an example of a surface between a state and an action in a trained Q state space with and without Gaussian kernel backup can be seen.

Besides adding a Gaussian kernel pre-simulation is done to increase the initial performance of the controller. The pre-simulation is done via the generic model described in Eq. (5) to (10). The reason that a simple generic model is suitable for the initial guess, is that it should work for all the different buildings the product is installed to. The generic model was tried on the different buildings described in the next section and performed satisfactory. An example of this can be seen in the results Fig. B.5.

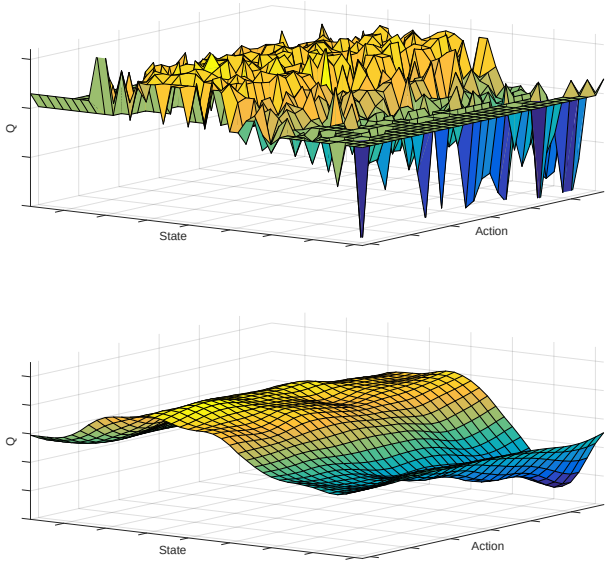

Fig. B.3: Q surface compare without (left) and with (right) Gaussian kernel backup

5 Simulation Setup

The testing of the algorithm is done via simulation on high fidelity building models. The building model is made using the Modelica library "Buildings" [21]. To show the learning ability of the controller, it is used on three different buildings; House from 2015, house from 1960, apartment from 2015 and apartment from 1960. Fig. B.4 shows the two floor plans of the house ($230m^2$) and the one of the apartment ($68m^2$).

Free heat from metabolism, electronics and hot water usage is modelled from typical daily, weekly and monthly patterns of usage. The difference between 2015 and 1960 buildings is the standard building materials of the time and standards for insulation, where Danish buildings from 2015 has a higher degree of insulation. Danish building code is used from each of the periods. The three buildings are situated in Copenhagen Denmark. For comparison some industrial standard controllers are used. There are typically four different tuning parameters to be chosen for the industrial controls. All buildings are supplied by 6 m head pumps. The industrial controllers are running proportional pressure. This means that the pressure rises proportional to the flow. The first parameter is the 3 different levels of proportional control that can be chosen on the selected pump. The next parameter is the outdoor temperature compensation. Here a saturated linear relation between outdoor - and forward temperature is often used. Besides this relation there is often

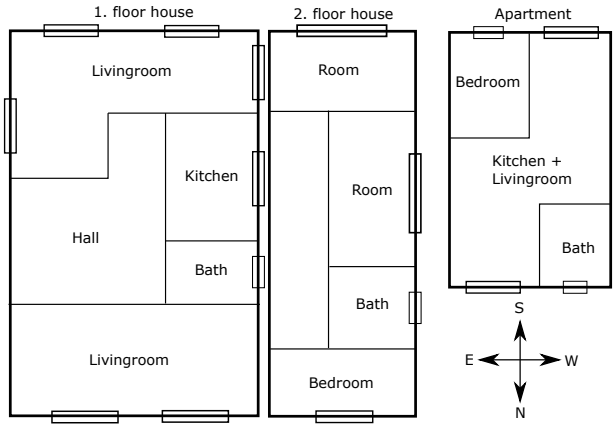

Fig. B.4: Floor plans of house and apartment.

a first order filter applied to the compensation with a time constant, that is the third parameter. The time constant should be matched to compensate the dynamics of the building. If the compared industrial controller is without outdoor compensation then a notation of NW is used. The last parameter is a constant temperature that is to be subtracted from the outdoor compensated forward temperature during setback. Two setback temperatures are used; 15^oC and 30^oC and will be noted as such in the comparison tables. The pump curve, outdoor temperature compensation and filtering is tuned to the specific buildings to give a comparison against well tuned controllers. The tuned controller for modern house is C1, old house C2 and modern apartment C3. For all setback controllers, the setback period is between 9 p.m. and 6 a.m.

When comparing controllers the most important measure of the optimality of the controller is the returns, see (18). Normalized return is used which is the cumulative reward measured every 5 min. over the heating season. Here the heating season is chosen to be the 9 months September-May. For comparison of the controls the discount rate for this return is 1 meaning that all rewards during the heating season counts as equal. The return is normalized by the number of samples for readability. To also be able to compare the controllers directly on the comfort and cost two other measurements are given in the results, the Root Mean Square Error (RMSE) and the cumulative cost of running the system during the heating season.

6 Results & Discussion

In this section results showing the improvement of adding Gaussian kernel backup and pre-simulation will be shown. The results is a comparison with the industrial standard controllers. It is important to emphasize, when evaluating these results, that the industrial benchmark controller such as e.g. $C1-30$ has been carefully tuned for the specific house, which rarely is the case for real world buildings. This means that achieving performance as good as $C1-30$ via a self learning controller results in a much better performance than what is experienced in worse tuned buildings. In Fig. B.5 the convergence of the reinforcement learning controller is shown for standard Q-learning backup, with Guassion Kernel backup, and finally adding pre-simulation. For each training duration, in interval of 1 month, the controller is run for a full heating season and the norm. return for that training duration is calculated. In this way it can be seen how the controller agent improves as a function of training duration. It can be seen that using the Gaussian kernel backup improves the initial performance until approximately the 18th month. Furthermore the Gaussian kernel backup improves the "stability" of the convergence, where the classic Q-learning deteriorates in periods, e.g. from 30-36 months. This graph also show the problem of learning Tabula Rasa. It takes around 30 months before reaching a satisfactory performing agent as the industrial controller C1-30, which is not feasible. Initialization using a priori knowledge by pre-simulating on the generic model provides a better initial controller. More work still needs to be done into increasing convergence rate, since training time still takes too long. The next results are comparisons of performance after 60 months.

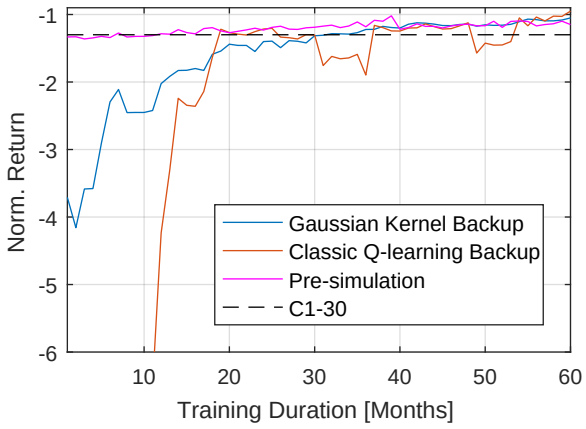

Fig. B.5: Norm. Returns as a function of training duration.

In table II a comparison of the trained Q-learning agent with industrial standard controllers is shown. In parenthesis is the relative improvement the Q-learning provides compares with the industrial controller. The Q-learning agent manages to save energy in all scenarios. Only in two scenarios does the comfort decrease slightly, while gaining large savings. In the modern house the $C1-30$ is the best industrial controller measured in return. Compared to this the improvement in comfort and cost from using the Q-learning agent is 4.5% and 3.2%. Had the industrial controller been tuned worse for the

Controller	Norm. Return	RMSE [^oC]	Cost €
Modern House - Copenhagen			
Q	-1.06	1.27	971
C1-15	-1.25	1.31 (3.1%)	1056 (8.0%)
C1-30	-1.19	1.33 (4.5%)	1003 (3.2%)
C1-30-NW	-1.29	1.39 (8.6%)	1018 (4.6%)
Old House - Copenhagen			
Q	-0.96	1.12	1920
C2-15	-1.25	1.11 (-0.9%)	2128 (9.8%)
C2-30	-3.24	1.20 (6.6%)	1985 (3.3%)
C2-30-NW	-4.13	1.26 (11.1%)	2022 (5.0%)
Modern Apartment - Copenhagen			
Q	-0.61	0.96	492
C3-15	-0.72	0.94 (-2.1%)	539 (8.7%)
C3-30	-0.74	0.96 (0.0%)	512 (3.9%)
C3-30-NW	-0.77	1.03 (6.8%)	521 (5.6%)

Table B.2: Comparison of Controllers With Setback.

modern house by choosing a setback constant of 15 the savings would instead be 8%.

Fig. B.6 shows an example of the time series data of one of the zone temperatures with the Q-learning agent and with the best tuned industrial controller $C1-30$ is shown. The Q-learning agent manages to increase energy savings by increasing the temperature reduction during setback. The Q-learning does this without violating comfort requirements by starting the reheating before leaving setback mode. If an increased comfort is desired the tuning parameter β in the reward function can be adjusted. To see how tuning β affects the performance, see Fig. B.7. Here it is shown that the the agent with lower β starts to lower the temperature later to keep the comfort higher before setback occurs. Likewise it raises the temperature earlier before leaving setback to increase comfort. Recall that the agent is controlling forward

temperatures and pressure, while a thermostat controls the zone temperature. It is by forcing the thermostat into saturation that the lowering of the zone temperature is possible from the mixing loop. Since the thermostat is a p-controller there will be a temperature error which is quite noticeable at around 5 o'clock in Fig. B.7. If setback is disabled the Q-learning agent still

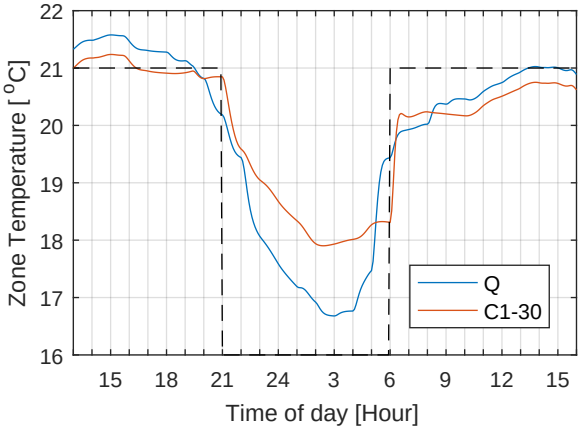

Fig. B.6: Example of zone temperature during setback. Padded line is during setback the constraint and out of setback the set-point.

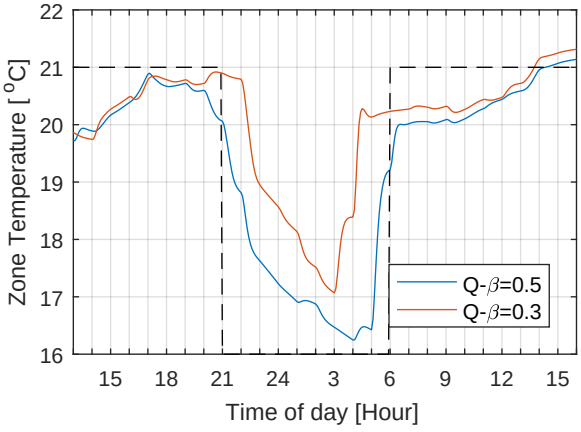

Fig. B.7: Comparison of setback example with different β

manages to save cost while achieving comparable comfort compared to the well tuned controller $C1$ in the modern house, which can be seen in table III. By comparing cost of the Q-learning agent with and without setback it can be

Modern House

Controller	Norm. Return	RMSE [^oC]	Cost €
Q	-2.01	1.23	1029
C1	-2.12	1.21	1076 (4.4%)

Table B.3: Comparison of Controllers Without Setback

seen that a saving of 5.6% is achieved through setback in the modern house. Comparing the industrial controller C1 without setback with the Q-learning agent with setback leads to 9.8% savings.

7 Conclusion

The motivation for this work is to investigate the performance of the reinforcement learning method Q-learning on building heating through mixing loops, while improving on the method by adding a Gaussian kernel backup and pre-simulation on a suggested generic model. In this work it was shown that even with the minimal information via a limited state-action space the reinforcement learning converges to a better performance than industrial standard controllers. Funnelling more information into the agent, such as free heat indicators, should increase the performance even further. However adding more information will decrease the convergence rate. To improve the convergence rate of Q-learning a Gaussian kernel backup method was added. Adding the Gaussian kernel added increased initial convergence rate, but even with the added convergence rate it still took 30 months to reach a satisfactory performance of the agent. By further adding pre-simulation on a generic model the initial controllers performance was greatly enhanced. The convergence rate however is still low, and need further improvement.

References

[1] A. Afram, F. Janabi-Sharifi, A. S. Fung, and K. Raahemifar, "Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system," *Energy and Buildings*, vol. 141, pp. 96–113, 2017.

[2] X. Cao, X. Dai, and J. Liu, "Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade," *Energy and Buildings*, vol. 128, pp. 198–213, 2016.

[3] J. Cigler and S. Prívara, "Subspace identification and model predictive control for buildings," 2010, pp. 750–755.

[4] A. I. Dounis and C. Caraiscos, "Advanced control systems engineering for energy and comfort management in a building environment-A review," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 6-7, pp. 1246–1261, 2009.

[5] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel, "Reinforcement learning versus model predictive control: A comparison on a power system problem," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 2, pp. 517–529, 2009.

[6] L. Gynther, B. Lappillone, and K. Pollier, "Energy efficiency trends and policies in the household and tertiary sectors. An analysis based on the ODYSSEE and MURE databases," no. June, p. 97, 2015. [Online]. Available: http://www.odyssee-mure.eu/publications/br/energy-efficiency-trends-policies-buildings.pdf

[7] HOFOR, "District Heating Prices." [Online]. Available: http://www.hofor.dk/fjernvarme/prisen-paa-fjernvarme-2017/

[8] S. Liu and G. P. Henze, "Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 1. Theoretical foundation," *Energy and Buildings*, vol. 38, no. 2, pp. 142–147, 2006.

[9] D. Ormoneit and Å. Sen, "Kernel-based reinforcement learning," *Machine Learning*, vol. 49, no. 2-3, pp. 161–178, 2002.

[10] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality: Second Edition*, 2011.

[11] S. Prívara, J. Široký, L. Ferkl, and J. Cigler, "Model predictive control of a building heating system: The first experience," *Energy and Buildings*, vol. 43, no. 2-3, pp. 564–572, 2011.

[12] F. Ruelens, S. Iacovella, B. J. Claessens, and R. Belmans, "Learning agent for a heat-pump thermostat with a set-back strategy using model-free reinforcement learning," *Energies*, vol. 8, no. 8, pp. 8300–8318, 2015.

[13] R. Sallent Cuadrado, "Return temperature influence of a district heating network on the CHP plant production costs," Ph.D. dissertation, 2009. [Online]. Available: http://hig.diva-portal.org/smash/get/diva2:228450/FULLTEXT01

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

[15] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. Van Den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[16] J. Široký, F. Oldewurtel, J. Cigler, and S. Prívara, "Experimental analysis of model predictive control for an energy efficient building heating system," *Applied Energy*, vol. 88, no. 9, pp. 3079–3087, 2011.

[17] R. Sutton and A. Barto, "Reinforcement Learning: An Introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, 1998. [Online]. Available: http://ieeexplore.ieee.org/document/712192/

[18] D. Urieli and P. Stone, "A Learning Agent for Heat-Pump Thermostat Control," no. May, 2013, pp. 1093–1100. [Online]. Available: http://dl.acm.org/citation.cfm?id=2484920.2485092

[19] C. J. C. H. Watkins, "Learning From Delayed Rewards," Ph.D. thesis, Cambridge University, 1989.

[20] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992. [Online]. Available: http://link.springer.com/10.1007/BF00992698

[21] M. Wetter, M. Bonvini, T. Nouidui, W. Tian, and W. Zuo, "Modelica buildings library 2.0," *14th International Conference of IBPSA - Building Simulation 2015, BS 2015, Conference Proceedings*, vol. 7, pp. 253–270, 2015.

The paper has been published in the
Proceedings of 2017 IEEE Conference on Control Technology and Applications (CCTA).

Paper C

Reinforcement Learning for Mixing Loop Control with Flow Variable Eligibility Trace

Anders Overgaard
Brian Kongsgaard Nielsen
Carsten Skovmose Kallesøe
Jan Dimon Bendtsen

The paper has been published in the
Proceedings of IEEE Conference on Control Technology and Applications 2019

Abstract

Mixing Loops are often used for proper pressurization and temperature control in building thermal systems. Optimal control of the mixing loop maximizes comfort while minimizing cost. To ensure optimal control for mixing loops in a wide range of different buildings with different load conditions, a self learning controller is here proposed. The controller uses Reinforcement Learning with flow variable eligibility trace. The controller is shown to improve performance of the mixing loop control compared to state of the art reinforcement learning and industrial grade controllers. The controller is tested on a hardware in the loop setup for rapid testing of mixing loop control used in building heating.

1 Introduction

There is a lot of energy to be saved by improving building heating, ventilation and air-conditioning (HVAC). In the United States 40% of energy consumption is in buildings, with 50% of that being from HVAC systems [12]. It is estimated in [2] that 11-16% can be saved by improving control. Due to this huge savings potential multiple control schemes are being researched, where Model Predictive Control [1] and Multi-Agent Systems [14] are two promising areas.

A major problem is improper or lack of commissioning of building. Only around 5% of buildings get commissioned [9], leaving the remaining buildings without well tuned HVAC controls. By introducing self learning controls the need for commissioning can be diminished. Reinforcement Learning as a self learning controller has been studied for building HVAC in [3], [5] and [4]. Reinforcement learning was combined with deep learning function approximation for HVAC control in [13] and building energy optimization in [6].

In this paper the focus is on Mixing Loops which is part of the hydraulic thermal distribution system in buildings. In [7] a method for taking into account the flow variable delay in prediction of the return temperature was shown to improve the prediction. In this work it is shown that adding flow variable delay into a Reinforcement Learning controller improves the performance leading to cost savings.

The paper starts with an introduction to Mixing Loops in Section II. Preliminaries covering concepts of Reinforcement Learning is given in Section III. In Section IV the proposed method using flow variable eligibility trace is presented. Section V explains the hardware in the loop test setup. Section VI explains how the hyper parameter for the controller is determined. The results are presented and discussed in Section VII. The paper ends with the concluding remarks in Section VIII.

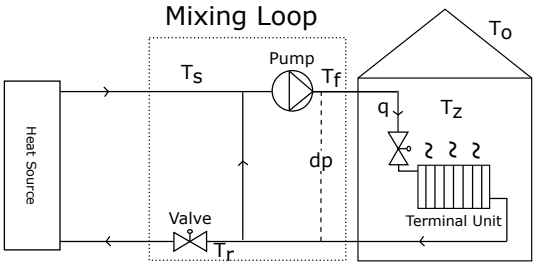

Fig. C.1: Schematic of simple mixing loop application

2 Building Heat Supply via Mixing Loop

Mixing loops are used in building heating and cooling systems to ensure proper pressurization and thermal power utilization. In this work a heating application is examined, but the same methods can be applied to cooling systems. Fig. C.1 shows the a minimal example where a single terminal unit is being supplied by the mixing loop. Here the terminal unit can be any hydronic based heating, be it radiator, floor, or ventilation based, but all fitted with a control valve having a local temperature controller.

By controlling the mixing valve the mixing loop can control the temperature of the water going to the terminal unit. The mixing loop causes a hydraulic decoupling from the heat supply, such that the pressurization is controlled by the mixing loop pump. By changing temperature or pressure the control gain for the terminal unit is changed. Furthermore it is possible to drive the terminal unit into saturation, which can be desirable when e.g. forcing a temperature setback.

The objective is to ensure enough heat power for the following terminal units while minimizing pump and heat power consumption. Additionally temperature setback can be used outside the operating hours of the building.

In this work the focus is on district heating as a heat source. In district heating it is important that the return temperature is as low as possible to increase the efficiency of the district heating system. This is in many places enforced by increasing the heat power cost as a function of low ΔT.

3 Preliminaries

This work makes use of the Reinforcement Learning controller $\mathrm{Q}(\sigma, \lambda)$ introduced in [15] which combines state of the art methods for dealing with temporal difference and eligibility traces in a unified manner. In this section some basic concepts of Reinforcement Learning are briefly summarized. For a deeper look into Reinforcement Learning the reader is referred to [11].

3.1 Basics

The basic idea of Reinforcement Learning is training a controller via reinforcing the desired behaviour by a reward as seen in Fig. C.2. At every time step, t, a reward (R_t) is given. The controller seeks to choose an action that optimizes the following series of rewards called the return (G)

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \tag{C.1}$$

Here $0 \leq \gamma \leq 1$ is a discount rate diminishing future rewards influence on the return. In Reinforcement Learning the control law for the controller is often called a policy π. A value function describes the expected return of being in a state and following a policy

$$v_\pi(s) \doteq \mathbb{E}[G_t | S = s]. \tag{C.2}$$

Another form is the action-value function describing the expected return of being in a state S, taking action A and then follow the policy

$$q_\pi(s, a) \doteq \mathbb{E}[G_t | S = s, A = a]. \tag{C.3}$$

A policy that given state, chooses the action that maximizes the expectation of return is called a greedy policy

$$A_t = \arg\max_a q_\pi(s, a). \tag{C.4}$$

By taking actions and sampling rewards the controller can over time improve the estimate of the value- or action-value function. To ensure exploration, policies such as ϵ-greedy which takes random actions with ϵ probability may be used.

3.2 Temporal Difference

Temporal difference is a central concept of Reinforcement Learning, where ideas from both Monte Carlo and Dynamic Programming are used. Where Monte Carlo waits until the episode is finished to update the estimate of the value function, dynamic programming bootstraps using current estimates to form a new estimate. A simplified representation of this is that Monte Carlo uses an estimation of [11]

$$q_\pi(s, a) \doteq \mathbb{E}_\pi[G_t | S_t = s, A_t = a]. \tag{C.5}$$

Since the expectation is not known a Monte Carlo method uses a sampled return to estimate the value function. Evaluating the same problem using dynamic programming leads to an estimate of

$$q_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]. \tag{C.6}$$

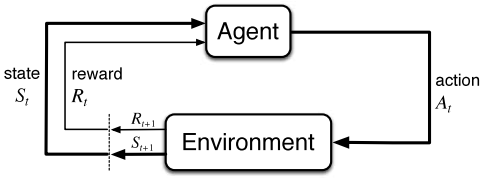

Fig. C.2: Reinforcement Learning [11].

Here the problem is not the estimate which is provided by a model of the system, but that $q_\pi(S_{t+1}, A_{t+1})$ is not known. Instead an estimate from current knowledge is used $Q(S_{t+1}, A_{t+1})$ in bootstrapping.

Temporal difference combines the concepts of Monte Carlo methods and dynamic programming and uses both the sampled values to give an estimate of the expectation while using the current estimate Q of q_π.

The temporal difference error δ is the error between the former estimated value $Q(S_t, A_t)$ and the updated estimate $R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$ used in various forms throughout reinforcement learning.

In [11] the method $\mathsf{Q}(\sigma)$ was first introduced. Here σ is used as a weight between two approaches to temporal difference error

$$\delta_t^\sigma = \sigma_{t+1}\delta_t^S + (1 - \sigma_{t+1})\delta_t^Q. \tag{C.7}$$

σ determines the amount of sampling with the method SARSA $(\sigma = 1)$ being in one end with temporal difference error using full sampling

$$\delta_t^S = R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t). \tag{C.8}$$

And at the other end $(\sigma = 0)$ is Expected SARSA using only expectation where for the special case, the often used Q-learning, the temporal difference error is

$$\delta_t^Q = R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t). \tag{C.9}$$

3.3 Eligibility Traces

Multi Step Reinforcement Learning learns from the return

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n}), 0 \leq t \leq T - n \tag{C.10}$$

In [10] the TD(λ) method was introduced where a trace decay of the returns in implemented

$$G_t^\lambda = (1 - \lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} G_{t:t+n} + \lambda^{T-t-1} G_t \tag{C.11}$$

In one end at $\lambda = 0$ is the one step algorithms and at $\lambda = 1$ Monte Carlo. In this way reinforcement learning can be tuned to work on different time horizons. Eligibility traces is a smart way of implementing these traces where a trace vector is used instead of saving all earlier steps. When using function approximation the value function can be approximated as $\hat{v}(s, \mathbf{w}) = v_\pi(s)$. Eligibility trace is a vector $\mathbf{z}$ that changes when the corresponding $\mathbf{w}$ is changed and afterwards fades, creating a short term memory.

$$\mathbf{z_t} \doteq \gamma\lambda\mathbf{z}_{t-1} + \nabla\hat{v}(S_t, \mathbf{w}_t) \tag{C.12}$$

$\mathbf{z}$ is then used for weighing how much $\mathbf{w}$ is changed under the backup.

$$\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha\delta_t\mathbf{z}_t \tag{C.13}$$

3.4 The method $\mathrm{Q}(\sigma, \lambda)$

By combining the ideas from $\mathrm{Q}(\sigma)$ and TD(λ) [15] developed $\mathrm{Q}(\sigma, \lambda)$, which is the method that this work builds on. $\mathrm{Q}(\sigma, \lambda)$ uses the temporal difference error δ_t^σ in (C.7), the eligibility trace of (C.12) and the backup of C.13. The proposed algorithm for reinforcement learning of mixing loops $\mathrm{Q}_\phi(\sigma, \lambda)$, which builds on a variation of $\mathrm{Q}(\sigma, \lambda)$, is introduced in section 4.

3.5 Radial Basis Function Approximation

A linear function approximation is used where the state action value function is approximated as

$$\hat{Q}(\mathbf{s}, \mathbf{a}, \mathbf{w}) = \mathbf{w}^T\mathbf{x}(\mathbf{s}, \mathbf{a}) = \sum_{i=1}^{d} w_i x_i(\mathbf{s}, \mathbf{a}) \tag{C.14}$$

The state vector, $\mathbf{s}$ has the dimension n_s and the action vector, $\mathbf{a}$ has n_a. The dimension d is the number of feature points and weights.

A radial basis function is used where the feature points $\mathbf{c}$ are in $\mathbb{R}^{n=n_s+n_a}$

$$x_i(\mathbf{s}, \mathbf{a}) = exp\left(-\sum_{k_s=1}^{n_s} \frac{(s_{k_s} - c_{k_s,i})^2}{2\varsigma_{k_s,i}^2} - \sum_{k_a=n_s+1}^{n_a+n_s} \frac{(a_{k_a} - c_{k_a,i})^2}{2\varsigma_{k_a,i}^2}\right) \tag{C.15}$$

4 Proposed Method

Here a Reinforcement Learning method for Mixing Loops taking into account flow variable transport delays is proposed called $\mathrm{Q}_\phi(\sigma, \lambda)$.

4.1 Flow Dependent Eligibility Trace

To ensure a high ΔT the time horizon over which the return (G) is found needs to contain the return temperature that arises from changing mixing temperature. This means determining λ such that the $n - step$ return containing the return temperature is weighted high. In a single pipe system with volume V_{pipe} the transport delay between the mixing temperature and the return temperature is a function of the flow

$$T_r(t) = T_m\left(t - \frac{V_{pipe}}{q(t)}\right). \tag{C.16}$$

In a system containing multiple pipes, the water will flow in different "routes", with different flow and volumes leading to various transport delays. For this work only a single lumped volume V_l is considered. This lumped volume should not be considered as the sum of volumes, but as the volume that gives the most impact on the input output relation of the temperature. The proposed method lets λ be dependent on the varying transport delay as

$$\lambda(t) = \frac{\phi}{q_n(t)} \quad q_n(t) \in [q_{n,min} \leq q_n(t) \leq 1], \tag{C.17}$$

It is here stated that the ϕ^* giving optimal performance can be found as a function

$$\phi^* = h(V_n, t_s). \tag{C.18}$$

A function, h, that gives the optimal ϕ as a function of V_n and q_n that are the lumped volume and flow. These are scaled by the max flow as

$$V_n = \frac{V}{q_{max}}, \qquad q_n(t) = \frac{q(t)}{q_{max}} \tag{C.19}$$

When the flow goes to zero the delay goes to infinity. To handle this a minimum flow $q_{n,min}$ is used. The function $h(V_n, t_s)/q(t)$ maps into a $\lambda_t \in \mathbb{R}$: $0 \leq \lambda \leq 1$. t_s is the sampling time.

4.2 Flow dependent $\mathrm{Q}_\phi(\sigma, \lambda)$

The proposed algorithm for online $\mathrm{Q}_\phi(\sigma, \lambda)$ with flow variable λ can be seen in Algorithm 6. For the operation of finding solutions to problems such as $\max_{\mathbf{a}} Q(\mathbf{w}^T\mathbf{x}(\mathbf{s}, \mathbf{a}))$ different solvers can be used. In this work a search algorithm was made, which utilizes the knowledge of location of feature points in the radial basis network to make multiple local gradient searches for finding a global maximum. Due to scope of this paper, this solver will not be further introduced.

Result: Online $Q_\phi(\sigma, \lambda)$
Initialize : Weights $\mathbf{w}$, trace vector $\mathbf{z}$. Take action $\mathbf{a}'$ according to ϵ-greedy $\pi(.|\mathbf{s}_0)$. Calculate feature state $\mathbf{x} = \mathbf{x}(\mathbf{s_0}, \mathbf{a}')$. $Q_{old} = 0$
Parameters : ϵ, α, γ, ϕ
repeat every sample
  $\mathbf{a} \leftarrow \mathbf{a}'$
  Observe R and $\mathbf{s}'$
  Choose $\mathbf{a}'$ according to ϵ-greedy π
  $\mathbf{x}' \leftarrow \mathbf{x}(\mathbf{s}', \mathbf{a}')$
  $Q \leftarrow \mathbf{w}^T\mathbf{x}$
  $Q'_S \leftarrow \mathbf{w}^T\mathbf{x}'$
  $Q'_Q \leftarrow \max_{\mathbf{a}'} (\mathbf{w}^T\mathbf{x}(\mathbf{s}', \mathbf{a}'))$
  $\delta^\sigma \leftarrow \sigma(R + \gamma Q'_s - Q) + (1 - \sigma)(R + \gamma Q'_q - Q)$
  Observe flow q
  if $q_{max} \leq q$ **then**
    $q_n \leftarrow 1$
  else if $q \leq q_{min}$ **then**
    $q_n \leftarrow q_{min}/q_{max}$
  else
    $q_n \leftarrow q/q_{max}$
  end
  $\lambda \leftarrow \frac{\phi}{q_n}$
  $\mathbf{z} \leftarrow \gamma\lambda\mathbf{z} + (1 - \alpha\gamma\lambda\mathbf{z}^T\mathbf{x})\mathbf{x}$
  $\mathbf{w} \leftarrow \mathbf{w} + \alpha(\delta^\sigma + Q - Q_{old})\mathbf{z} - \alpha(Q - Q_{old})\mathbf{x}$
  $\mathbf{x} = \mathbf{x}'$
  Take action $\mathbf{a}'$
until *Mixing Loop Stop*;

Algorithm 6: Algorithm $Q(\sigma, \phi)$

5 Test

Testing on buildings is not a trivial task. It is very time-consuming due to slow dynamics and there is often a desire to test performance over multiple years. Furthermore benchmarking can be imprecise due to not having an equal comparison due to different load conditions. This can pose a problem for rapid development. A hardware in the loop approach is used here for faster testing.

The test setup consists of two parts; A hydraulic mixing loop system and a building model. The hydraulic dynamics of the mixing loop react much faster than the thermal dynamics of the building. To increase testing speed, the model of the building is run at accelerated speed in the loop together

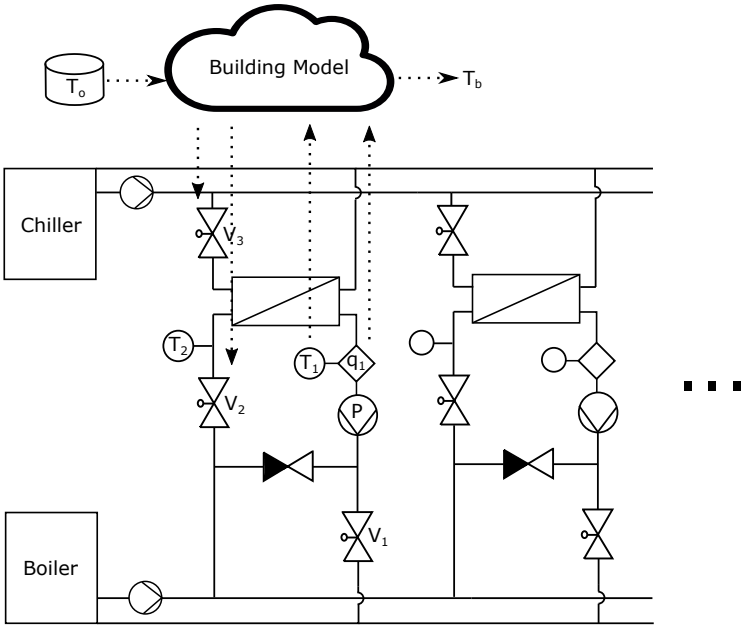

Fig. C.3: Hardware in the loop. Four parallel systems installed.

with the hardware hydraulics. The idea is to run the building model faster, but still slower than the hydraulic dynamics to increase testing speed. This allows for, in the specific test setup, to simulate 12 days in the time of 1 day.

5.1 Hydraulics

Fig. C.3 shows a simplified setup containing mixing loops, a boiler for heat generation and a chiller for generating the chilled water used to simulate the load. In Fig. C.3 V_1 is the mixing valve that controls the mixing temperature T_2. The controller is a local PI controller with gain scheduling on the flow q_1 to compensate for the flow dependent gain. Pump P has local speed controller. The set point for the mixing temperature and the pump speed is controlled by the Reinforcement Learning algorithm.

To simulate the impact of the building on the hydraulics of the mixing loop the valves V_2 and V_3 are temperature controlled via PI controllers. V_2 controls the building temperature (T_b) in the building model to a constant room temperature of 21^oC. In this way the mixing loop will experience a flow dictated by the building model. V_3 controls the return temperature T_1 according to the building model. In Fig. C.4 a picture showing part of the test setup can be seen.

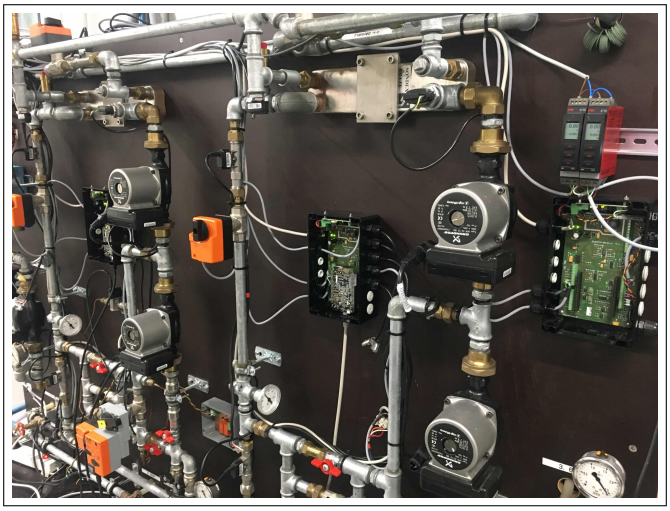

Fig. C.4: Picture overlooking part of test setup.

5.2 Building Model

The structure used for the building model is a Nonlinear Autoregressive External Input Neural Network (NARX net). The building model is trained on data gathered on an office building located in Bjerringbro, Denmark. The building is a 3 floor building with 34 radiator zones supplied by a single mixing loop and controlled to the same set point. In the model the zones are lumped into one by having the average of the 34 zone temperatures being the building temperature T_b. Furthermore the flow data from the building is scaled by a constant C such that $C \cdot q_{max,building} = q_{max,testsetup}$. This is done since the hydraulic network in the test is a smaller version of what the office building contains.

Training was done on 12 months of data from the building and validation on 3 separate months. Step variations was done on set points for mixing temperature and pump speed for a period of the time, while the rest was normal operation with industrial controller to improve model exploration. In Fig. C.5 examples of the fit over 8 days can be seen from the validation data. The Root Mean Square Error (RMSE) for the full validation set on T_b and T_r is 0.28 K and 0.64 K respectively. By visual inspection of the fit and evaluation of the low RMSE on the full validation data the model is deemed a good representation of the office building.

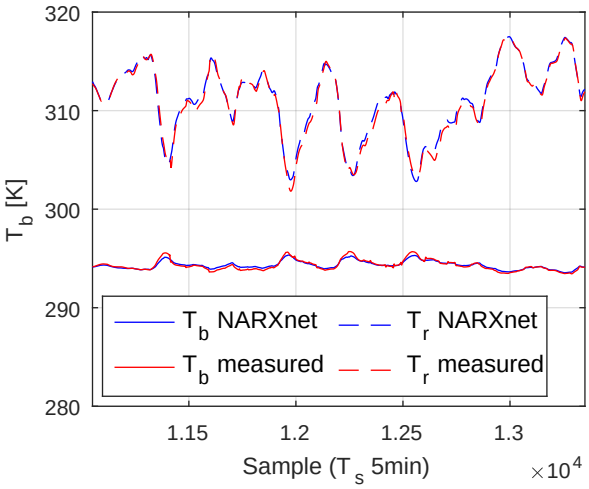

Fig. C.5: Validation of building- (T_b) and return temperature (T_r). Data shown is only 8 days example out the full 3 months validation data.

5.3 Controllers

$Q_\phi(\sigma, \lambda)$ is tested on this hardware in the loop setup with the following parameters: $\alpha = 0.8$, $\gamma = 1$, $\sigma = 0.8$, $\epsilon = 0.1$. The feature points are spread evenly according to dimension over the ranges specified in TABLE C.1 giving $d = 3000$ weights to be trained. The reward function is defined as

$$R(t) = \begin{cases} -(e(t)^2 + \beta(\psi_{heat}(t, \Delta T) + \psi_{pump}(t))) & 5 \leq t \ mod(24h) \leq 21 \\ -\beta(\psi_{heat}(t) + \psi_{pump}(t)) & otherwise \end{cases} \tag{C.20}$$

Here $e(t)$ is the building temperature error that is used when in heating mode, but not in set back mode. $\beta = 0.5$ is a weight between comfort and cost due to the multi objective nature of the reward. $\psi_{heat}(t, \Delta T)$ is the heating power cost which is determined by the district heating company as a function of ΔT. The lower ΔT the higher price. $\psi_{pump}(t)$ is the pump power cost. See [8] for further description of the chosen reward function. To determine the performance of the proposed algorithm it is compared with an industrial standard controller for mixing loops. The controller that is being compared with is the one installed in the office building from which the model was derived. This controller is developed by a major Building Management System (BMS) supplier that is kept anonymous. The industrial controller also performs night setback in the hours 21 to 5.

Table C.1: Radial basis function dimension and range.

State-Action	Dimension	Width	Range
Hour of Day	10	1.5	1-24 [h]
Outdoor Temperature	10	2.5	253-293 [K]
Mixing Temperature	10	3	293-343 [K]
Pump Speed	3	20	0-100 [%]

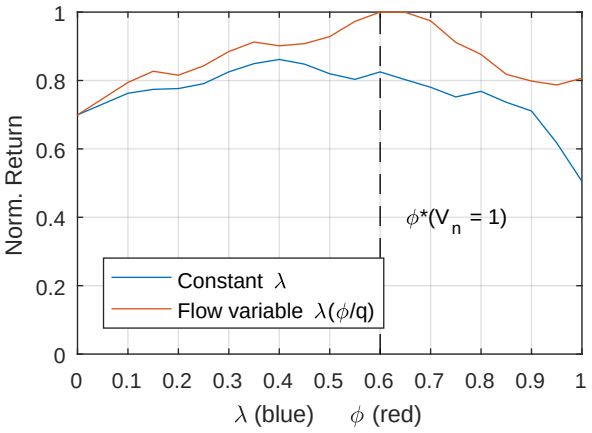

Fig. C.6: Norm. yearly return for different values of constant λ and flow variable delay with different α for physical model with $V_n = 20$.

5.4 Determining ϕ^*

A first principle physical model which was introduced in [8] is first used to compare performance of $Q(\sigma, \lambda)$ with the proposed flow dependent $Q_\phi(\sigma, \lambda)$ for different values of λ and ϕ. The controller was trained for a year and afterwards evaluated running a second year. In Fig. C.6 it can be seen that the highest yearly return occurs at $\phi^* = 0.6$. To determine the relation between the lumped scaled volume and ϕ^* a numerical approximation was done by doing multiple test at different lumped volumes. The sampling time was kept constant $t_s = 300s$ such that an approximation for $\phi^* = h_{t_s=300}(V_n)$ was found as seen in Fig. C.7. From this relation ϕ^* can be determined for the building given V_n. Different ways can be used to get an estimate of the lumped model, such as knowledge of piping. Here it was determined for the office building in Bjerringbro via data analysis. A method for determining the lumped volume through data analysis using Mutual Information was shown in [7]. The lumped scaled volume (V_n) was for the tested building found to be $1.15m^3$ which via the linear approximation $h_{t_s=300}(V_n)$ leads to $\phi^* = 0.8$.

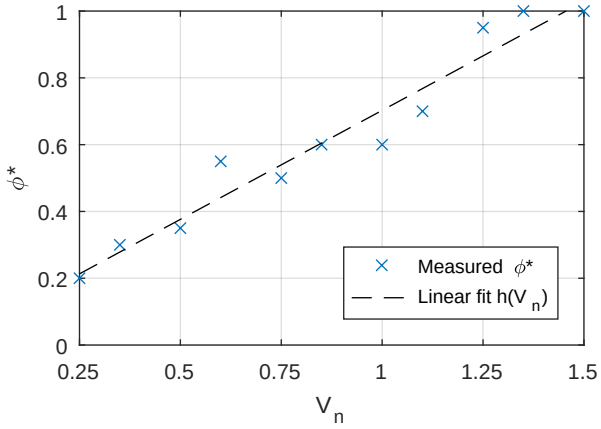

Fig. C.7: Relation between α and V_n. Used to determine $C(V_n)$ from physical model.

6 Results & Discussion

The first results shows how the $Q_\phi(\sigma, \lambda)$ controller performs over the first 5 months compared with the industrial controller. Using the rapid hardware in the loop setup this test took 12.5 days to run. Fig. C.8 the mean absolute value of the weights is shown to as a representation of how the 3000 weights converge. If the system is time invariant and the system is fully explored the weights will over time converge to a final value. There is a steep learning curve the first 45 days with following slower convergence. The development of the weights is an indication of stable convergence.

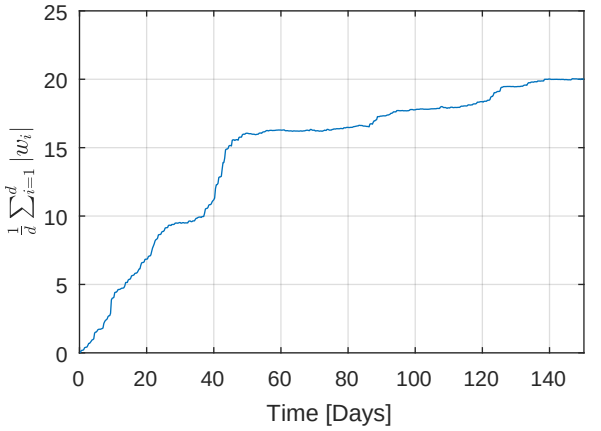

Fig. C.8: mean absolute value of weights converging during training.

Fig. C.9 shows the root mean square error (RMSE) of the building temperature and the heating cost with a moving mean covering the last 14 days. Here it can be seen that the RMSE is at first worse than the industrial controller, but over time $Q_\phi(\sigma, \lambda)$ learns to provide conditions for the local controller to achieve low RMSE. In the cost plot it can be seen that after approximately 55 days $Q_\phi(\sigma, \lambda)$ starts saving compared to the industrial controller. In the last plot the return of the rewards over 14 is shown which is the goal $Q_\phi(\sigma, \lambda)$ optimizes towards. Here 55 days seems to be the point where $Q_\phi(\sigma, \lambda)$ overtakes the industrial controller in performance defined by the reward function. Having worse performance for approximately 2 months than a well tuned industrial controller and afterwards improving seems reasonable, especially if some of the training can be done while commissioning of the building is ongoing.

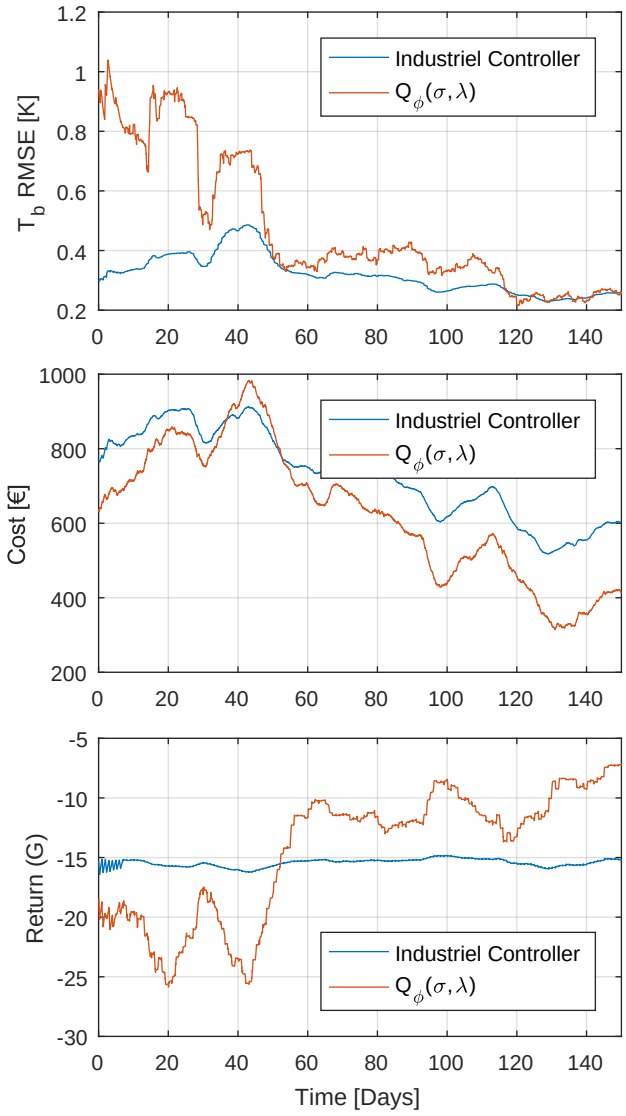

Fig. C.9: Temperature error, cost, and return in a running 14 days window.

For the next results the different controllers are allowed to train for 5 months. Afterwards the controllers are run for 6 months on a different weather data set than was used for training. Summation of normalized return, RMSE and Cost is done for all the controllers and compared in TABLE C.2.

Table C.2: Comparison of Controllers performance over 6 months after 5 months training.

Controller	Norm. Return	RMSE [K]	Cost €
Q(ϕ*)	-1	0.31	10076
Industrial	-1.44	0.28	12146 (20.5%)
Q($\phi^* - 0.2$)	-1.16	0.25	10519 (4.4%)
Q($\phi^* + 0.2$)	-1.25	0.33	10690 (6.1%)
Q($\lambda = 0.5$)	-1.29	0.29	10751 (6.7%)

All the controllers provide conditions for the local temperature controller to achieve similar low RMSE. On return and cost the proposed method with the estimated ϕ^* performs best. Compared to the industrial controller it saves 20.5% in costs over the 6 months period. The controllers with lower and higher ϕ values perform worse, but still better than the industrial controller and when using a constant eligibility trace λ.

7 Conclusion

A method for including flow variant eligibility traces into the state of the art Reinforcement Learning controller $\mathsf{Q}(\sigma, \lambda)$ was introduced. The proposed controller was tested on a hardware in the loop setup where a Mixing Loop system is combined with a building model fitted to data from an office building in Bjerringbro, Denmark. The advantages of this is faster test and easier comparison of multiple controllers since all parallel tests are on the same building model exposed to the same conditions. The disadvantage of this is that only the hydraulic part is real, such that shortcomings from incomplete knowledge in the building model can not be tested for. The proposed method improved performance over an industrial standard controller and $\mathsf{Q}(\sigma, \lambda)$ without flow variable eligibility trace. The proposed controller reached same performance as the tuned industrial controller after 50 days. After 5 months of training the proposed controller operated with same level of comfort, while saving 20.5% on cost.

References

[1] A. Afram and F. Janabi-Sharifi, "Theory and applications of HVAC control systems - A review of model predictive control (MPC)," 2014.

[2] X. Cao, X. Dai, and J. Liu, "Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade," *Energy and Buildings*, vol. 128, pp. 198–213, 2016.

[3] K. Dalamagkidis, D. Kolokotsa, K. Kalaitzakis, and G. S. Stavrakakis, "Reinforcement learning for energy conservation and comfort in buildings," *Building and Environment*, vol. 42, no. 7, pp. 2686–2698, 2007.

[4] L. Eller, L. C. Siafara, and T. Sauter, "Adaptive control for building energy management using reinforcement learning," in *Proceedings of the IEEE International Conference on Industrial Technology*, vol. 2018-February, 2018, pp. 1562–1567.

[5] P. Fazenda, K. Veeramachaneni, P. Lima, and U. M. O'Reilly, "Using reinforcement learning to optimize occupant comfort and energy usage in HVAC systems," *Journal of Ambient Intelligence and Smart Environments*, vol. 6, no. 6, pp. 675–690, 2014.

[6] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu, and J. G. Slootweg, "On-line Building Energy Optimization using Deep Reinforcement Learning," 2018.

[7] A. Overgaard, C. S. Kallesoe, J. D. Bendtsen, and B. K. Nielsen, "Input selection for return temperature estimation in mixing loops using partial mutual information with flow variable delay," in *1st Annual IEEE Conference on Control Technology and Applications, CCTA 2017*, vol. 2017-Janua, 2017, pp. 1372–1377.

[8] A. Overgaard, C. S. Kallesøe, J. D. Bendtsen, and B. K. Nielsen, "Mixing Loop Control using Reinforcement Learning," in *Proceedings of the 13th REHVA World Congress CLIMA 2019*, 2019, pp. Accepted, not yet published.

[9] K. W. Roth, F. Goldstein, and J. Kleinman, "Energy Consumption by Office and Telecommunications Equipment in Commercial Buildings Volume I: Energy Consumption Baseline," *Engineering*, vol. I, p. 201, 2002.

[10] R. S. Sutton, "Learning to Predict by the Methods of Temporal Differences," *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1988.

[11] R. S. Sutton and A. G. Barto, "Reinforcement learning: an introduction 2018 complete draft," *UCL,Computer Science Department, Reinforcement Learning Lectures*, p. 1054, 2017.

[12] U.S. Department of Energy, "2011 Buildings Energy Data Book," *Energy Efficiency & Renewable Energy Department*, p. 286, 2012.

[13] T. Wei, Y. Wang, and Q. Zhu, "Deep Reinforcement Learning for Building HVAC Control," in *Proceedings of the 54th Annual Design Automation Conference 2017 on - DAC '17*, 2017, pp. 1–6.

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

[15] L. Yang, M. Shi, Q. Zheng, W. Meng, and G. Pan, "A unified approach for multi-step temporal-difference learning with eligibility traces in reinforcement learning," in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2018-July, 2018, pp. 2984–2990.

Paper D

Reinforcement Learning for Building Heating via Mixing Loop with Data Driven Input Variable Selection

Anders Overgaard
Carsten Skovmose Kallesøe
Jan Dimon Bendtsen
Brian Kongsgaard Nielsen

The paper has been submitted for publication in the
IEEE Transaction on Neural Networks and Learning Systems

Abstract

A plug and play control scheme for mixing loops in building heating systems is proposed. Plug and play refers to achieving good control performance without any need for tuning the mixing loop controllers to a specific building. The overall control scheme consists of two components; A reinforcement learning controller to provide self learning optimal control and a data driven input variable selection part to achieve a good performance within a shorter training period. The approach is based on partial mutual information such that only a subset of the input sensors providing most relevant information is used. To account for the flow dependent delays present in the mixing loop, flow compensation is used both in the state selection and in determination of the length of the eligibility trace. The proposed control scheme is tested on a hardware-in-the-loop test setup where a mixing loop is supplying a heat exchanger emulating an office building. The results show that the proposed control scheme offers improved performance compared to a commercially available industrial controller and to reinforcement learning controllers using other subsets of inputs.

1 Introduction

Mixing loops are often used in building heating and cooling systems to ensure proper pressurisation and heat power utilization. Proper control of pressure and temperatures can ensure improved comfort, energy and monetary savings. Space heating and cooling account for a large part of the worlds energy consumption. With increased population and economic growth, cooling energy use in buildings has risen from 3.6 EJ to 7 EJ since 2000 [8]. The International Energy Agency predicts that without efficiency improvements this number will double by 2040, but in their increased efficiency scenario this can be kept to an increase of 19% [8].

Classically, mixing loops in industrial applications are controlled by a mixture of feed forward compensation and slow feedback due to long flow depenent delays. Optimal control has been studied in other HVAC applications, often in the form of Model Predictive Control (MPC) as described in [1]. The main drawback of MPC is the need for good models. Mixing loops are installed in a multitude of different buildings under different conditions, which would require expensive commissioning to match the models with the specific buildings in order to employ MPC in practice. In [12] 150 newly constructed buildings was tested with the conclusion that the average energy savings that could be achieved by a proper recommissioning of the existing equipment was 18%. To deal with this challenge a self learning optimal control scheme is sought. Model predictive control using methods such as subspace identification [3] or using grey box models such as ARMAX [26] has been considered. Reinforcement learning has been tried on different HVAC

systems in [4], [6] and [5]. In [24] and [13], deep learning is applied as function approximation for Reinforcement Learning on HVAC systems. In [15] mixing loop control using Reinforcement Learning with flow compensation is introduced.

Where model predictive control needs good models, Reinforcement Learning needs training time. When dealing with many inputs that has information about the system, training time is impacted by the curse of dimensionality. To avoid this and find a set of inputs giving good performance within a feasible time frame Input variable selection (IVS) is used for state selection. See [10] for a review of IVS techniques. Partial Mutual Information (PMI) was proposed in [18] as a method for selecting inputs in nonlinear systems, applying an iterative mutual information approach. This method was later used and improved in [11] and [9]. In [14] PMI was used to estimate return temperature in mixing loops with flow variable compensation.

In this work, the proposed method mainly consists of two components. A reinforcement learning control scheme where a flow variable eligibility scheme is proposed. The other component is an IVS scheme that chooses the inputs that provide the most relevant information for the reinforcement learning agent to increase training speed. This input variable selection is based on partial mutual information with a proposed flow compensation.

The structure of the paper is as follows. A brief introduction into reinforcement learning and mutual information is given in Section 3. In Section 2 building heating via mixing loop is explained. This is followed in Section 4 where the proposed method is introduced. This section is divided into three subjects; Reinforcement Learning Control of Mixing Loop, State Selection and a Method Overview. To test the proposed plug and play control scheme, a hardware-in-the-loop test setup was designed and is described in Section 5. The most important results of the test are presented and discussed in Section 6. The paper is ended by some concluding remarks in Section 7.

The notation used in this work is calligraphic letters for random variables $\mathcal{X}$, standard letters for scalars x, bold letters for vectors $\mathbf{x}$. Subscript $t : T$ means that it is a time series of the variable from time t to terminal time T.

2 Building Heat Supply via Mixing Loop

In hydraulic heating or cooling systems with multiple users or zones, mixing loops can be used to meet the different pressure and thermal power demands among consumers to improve performance and energy efficiency. In modern buildings a multitude of sensors and prediction data are available for the mixing loop control. Fig. D.1 illustrates a simplistic mixing loop system with only one terminal unit and examples of different sensors that can be used in buildings. The main objective of a building heating system is to keep the

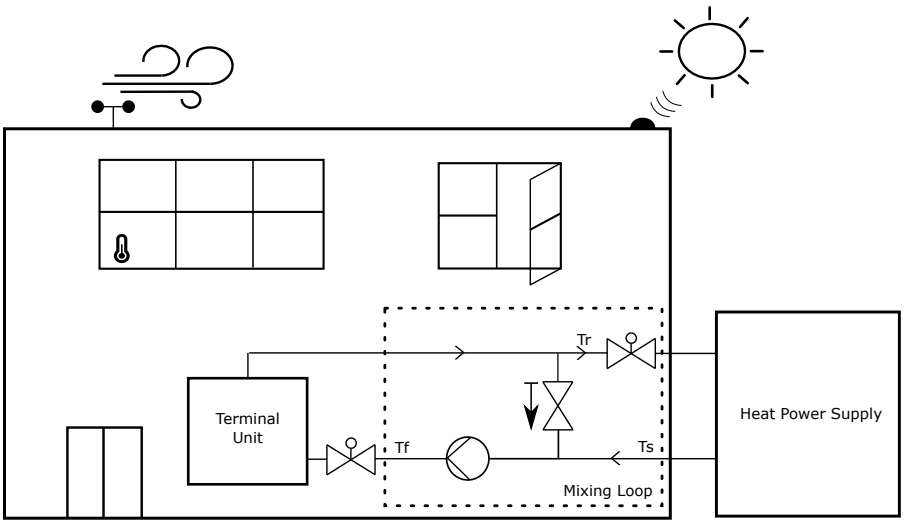

Fig. D.1: Schematic of a mixing loop application

users comfortable through proper zone temperatures. A simplified dynamic model of zone temperature T_z is

$$C_z \dot{T}_z = \Phi_h + \Phi_L + \Phi_d, \tag{D.1}$$

where C_z is the heat capacity of the zone, Φ_h is the heat power supplied from the terminal units, Φ_L is the heat load, which for building heating is negative. This load mainly comes from the outside air cooling the building. Φ_d is the disturbances which for building heating can arise from free heat generated by human metabolism, electric appliances or solar radiation etc.

Heat power is generated at a heat power supply, which can be of various types; District heating, heat pump, gas boiler, electrical boiler etc. With multiple consumers, such as different buildings in district heating or multiple zones in the case of a large building, the optimal pressures and temperatures are different. Optimal here meaning having maximum comfort in the building/zone at lowest cost. Furthermore the optimal temperature and pressure changes with changing load conditions. To handle this a mixing loop can be used to decouple the pressure and temperature from the supply system. This is achieved by having a shunt such that the return water can be mixed with the supply water. A pump controls the pressure for the zone by changing pump speed, ω, and a mixing valve controls the forward temperature T_f in the range between the supply temperature T_s and the return temperature T_r. A one way valve is used to eliminate direct flow from supply to return. From the mixing loop the heating or cooling water runs to a terminal unit: Radiator, air handling unit or floor heating etc. This terminal unit has

a local control loop where a control valve controls the zone temperature to a set point temperature. In a real system the mixing loop will often supply multiple terminal units.

While the main objective of the building heating system is to keep a high comfort for the user, it is also desired to minimise operational cost. The cost of operating the system, $\Psi(t)$, includes the cost of power for pumps and valves, but also the cost of the heat power flowing through the mixing loop. In non-operated hours of the building, e.g. at nights or during vacation time, the temperature can be lowered without causing discomfort. This is called setback and can be forced from the mixing loop control by lowering the pressure and the forward temperature enough such that the control valves of the terminal units are saturated and can therefore not deliver sufficient heat power to keep the temperature at the set points. The cost of heat power from different supplies varies with different parameters. In e.g. district heating low return temperature is important to mitigate heat losses. The heat power cost is often a function of ΔT at the consumer, thus at constant supply temperature, high return temperature increases the heat power cost. For a local electrical boiler with shorter pipes the return temperature matters little on the price of heat power, but the varying electricity prices can perhaps enable savings through load shifting. In load shifting the thermal storage of the building can be utilized to heat at low electricity price ahead of a period with high loads and high electricity prices.

Between the mixing loop system and the terminal units there will often be a large pipe network to carry the heating water. This causes flow dependent transport delays, which are fundamental to the mixing loop application. Temperatures propagating with the flow from the mixing loop throughout the pipe network to the terminal units and back again will be subject to a long transport delay. Flow impacts the velocity of the water, causing the transport delay to be a function of flow. This especially affects the relation between the forward temperature and the return temperature from the mixing loop which can be described as

$$T_r(t) = h\left(\mathbf{T}_f, \mathbf{q}, \mathbf{T}_z\right), \tag{D.2}$$

where

$$\begin{aligned}
\mathbf{T}_f &= \left[T_f\left(t - \frac{V_1}{q_1}\right), ..., T_f\left(t - \frac{V_N}{q_N}\right)\right]^T \\
\mathbf{q} &= \left[q_1, ..., q_N\right]^T \\
\mathbf{T}_z &= \left[T_{z1}\left(t - \frac{V_1}{2q_1}\right), ..., T_{zN}\left(t - \frac{V_N}{2q_N}\right)\right]^T.
\end{aligned}$$

Here, h is a function that describes the power dissipation in the system and N is number of transport routes the water can take through the pipes. $\mathbf{T}_f$ is

a vector of the forward temperature at varying flow dependent delay. $\mathbf{q}$ is a vector of the flow in the different pipe routes. $\mathbf{V}$ is the volumes in the pipe routes. $\mathbf{T}_z$ are the zone temperatures, which affects the power consumption, here located at equal supply and return pipe volume. Notice that the flow is quasi-static, meaning that it is assumed constant within the variable time frame V_N/q_N.

The objective of the mixing loop is to ensure heat power available for the terminal units to achieve high comfort, while minimizing cost. The mixing loop has two control variables to influence this; the forward temperature, T_f, and the pump speed ω. High comfort is here limited to achieving the desired zone temperature, but can be expanded to other comfort parameters, such as humidity. There are a lot of variations of heating and cooling systems where mixing loops can be used. It is important that the control can handle many different scenarios, be it cooling via ventilation, heating via radiator etc. If the control problem is defined as achieving the highest comfort at the lowest cost, the problem is however the same for all variations of the system. The control problem then becomes

$$\underset{T_f,\omega}{\text{minimize}} \quad \int_0^\infty L(t)||T_z(t) - T_{set}||_2 + W\Psi(t)\, dt \tag{D.3}$$

where $L(t)$ is a setback parameter, $T_z(t)$ is the zone temperature, T_{set} is a set point temperature for the zone temperature, W is a weight between cost and comfort, $\Psi(t)$ is the cost of operating the system and ω is the speed of the pump. In (D.1) a description of the zone temperature was given as a function of Φ_h, Φ_L and Φ_d. The heat power Φ_h can be influenced by the actions of the mixing loop. In this work the control problem is solved by a self learning optimal control scheme for mixing loops via reinforcement learning.

3 Preliminaries

Reinforcement Learning consists of a controlling agent that acts upon an environment. When acting upon the environment, a reward is given depending on the achieved state. This is often modelled by a Markov decision process where the probability of changing states from s to s' while taking the action a is

$$\mathcal{P}^a_{ss'} = p(s_{t+1} = s'|s_t = s, a_t = a). \tag{D.4}$$

Taking action a and bringing the environment into state s' yields the reward r_{t+1}. The expected reward of being in a state s and taking the action a can then be described as

$$R^a_{ss'} = \mathbb{E}[r_{t+1}|s_t = s, s_{t+1} = s', a_t = a]. \tag{D.5}$$

The agent seeks to act on the environment such that a cumulative n-step reward is maximized. This sum is referred to as the return and is often used on the form

$$G_{t+n} = \sum_{k=0}^{n-1} \gamma^k r_{t+k+1}, \tag{D.6}$$

where the added discount rate $0 \leq \gamma \leq 1$ ensures that the return is well defined going to infinite time, while ensuring a higher importance of rewards happening sooner. If a return is used that stretches to infinity $G_{t+\infty}$. The agent relies on a policy π to determine what action to take. To describe the expected return, when being in a state, taking an action and afterwards following a given policy the state action value function is used

$$Q_\pi(s,a) = \mathbb{E}[G_{t+\infty}|s_t = s, a_t = a] = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s, a_t = a]. \tag{D.7}$$

From (D.4), (D.5) and (D.7) the Bellman Equation for the state action value function can be derived

$$Q_\pi(s,a) = \sum_{s'} \mathcal{P}^a_{ss'} \left[R^a_{ss'} + \gamma \sum_{a'} \pi(s',a') Q_\pi(s',a') \right] \tag{D.8}$$

For the optimal policy that maximises the return this becomes

$$Q_{\pi^*}(s,a) = \sum_{s'} \mathcal{P}^a_{ss'} \left[R^a_{ss'} + \gamma \underset{a'}{max} Q_{\pi^*}(s',a') \right] \tag{D.9}$$

This is what is approximated in Q-learning [23]. Here the optimal action value function, no matter which policy is followed is found, with the condition that all state-action pairs needs to be continuously visited. The iterative backup for Q-learning is

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r_{t+1} + \gamma \underset{a'}{max} Q(s',a') - Q(s,a) \right]. \tag{D.10}$$

Here $\alpha \in [0,1]$ is the learning rate. While one goal for the agent is to approximate the value function another is for the agent to find the optimal policy π^* that maximises the return for every state. Since the value function and the optimal policy are dependent upon each other they are often optimized in an iterative fashion called the value-policy iteration.

A greedy policy is a policy that chooses the action that maximises the return

$$a_t = \arg\max_a Q_\pi(s_t, a). \tag{D.11}$$

The problem of the greedy policy is that no new exploration into a potentially more rewarding action is done. The trade-off between exploration and

exploitation of current knowledge is central to reinforcement learning since both optimality and adaptiveness is desired. Therefore, stochastic policies where an amount of exploration can be achieved are often used. For a deeper look into Reinforcement Learning the reader is referred to [21].

Temporal difference learning is a key concept in Reinforcement Learning. It is often described as a mixture of Monte Carlo and Dynamic Programming. In Monte Carlo the full episode of actions, state transitions and returns are measured and then the estimate of the state-action value function is computed purely from measurements. In dynamic programming a model of the Markov Decision Process is already known, so an estimate from this knowledge can be used for bootstrapping. This combines into temporal difference where the bootstrap target is calculated, both from the sampled reward and the system knowledge already acquired as seen in (D.10). The temporal difference error, δ is the error between the current estimation of the state-action value function and the new estimate. In the on policy method SARSA the temporal difference error is

$$\delta_t^S = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t). \tag{D.12}$$

On policy means that the agent is learning the state-action value function according to the same policy that is being followed. In off policy methods the behaviour policy being used by the agent is different from the target policy being learned. Q-learning in (D.10) is off policy since the target policy is the optimal policy as seen by the bootstrapping using the maximising action

$$\delta_t^Q = r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t). \tag{D.13}$$

Both temporal difference errors δ^S and δ^Q is here shown on one step form since only one measurement of the reward was used in the bootstrap target. Multi Step methods often perform better than single step by using more samples. In TD(λ) [20] this was parametrised using a trace decay λ of the returns such that it can span from $\lambda = 0$ being the one step methods and up to $\lambda = 1$ where it becomes a Monte Carlo method

$$G_t^\lambda = (1-\lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} G_{t+n} + \lambda^{T-t-1} G_{t+\infty}. \tag{D.14}$$

Multi step methods are almost always implemented as eligibility traces due to the computational advantages. An eligibility trace utilizes a trace vector $\mathbf{z}$ that changes according to the partial derivatives of the weights with respect to the estimated value function and decays by $\gamma\lambda$.

$$\mathbf{z_t} \doteq \gamma\lambda \mathbf{z}_{t-1} + \nabla_{\mathbf{w}} \hat{Q}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{w}_t). \tag{D.15}$$

The weights are then adjusted according to

$$\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha \delta_t \mathbf{z}_t. \tag{D.16}$$

The trace $\mathbf{z}$ is often implemented as an accumulating trace or a Dutch trace as proposed in [7].

As described by the value function (D.7) the state space used along with the chosen actions has to provide information about the estimation of the return. Input Variable Selection is a group of methods that deals with finding the set of input variables that gives the best prediction. In [10] several different approaches to input variable selection is reviewed. Based on the ability to handle nonlinear relations and fast computation the method Partial Mutual Information [18] was chosen. This method is based on mutual information. Mutual information between two continuous random variables $\mathcal{X}$ and $\mathcal{Y}$ is defined as [17]

$$I(\mathcal{X};\mathcal{Y}) = \int\int p(\mathcal{X},\mathcal{Y}) \log\left(\frac{p(\mathcal{X},\mathcal{Y})}{p(\mathcal{X})p(\mathcal{Y})}\right) d\mathcal{X}d\mathcal{Y}, \tag{D.17}$$

where $p(\mathcal{X}), p(\mathcal{Y})$ are the marginal probability density functions and $p(\mathcal{X},\mathcal{Y})$ is the joint probability density function. If the two variables are independent then $p(\mathcal{X},\mathcal{Y}) = p(\mathcal{X})p(\mathcal{Y})$ and the fraction $\frac{p(\mathcal{X},\mathcal{Y})}{p(\mathcal{X})p(\mathcal{Y})}$ equals 1, meaning no mutual information. Partial Mutual Information works in an iterative manner by choosing the input variable with most information, then removing that information from the prediction target leaving a new residual prediction target. Then the input giving most information about the residual prediction target is found and so forth until stopped or all input variables are sorted. A data driven method for choosing a subset of available states to represent the state space of the value function is proposed in Section 4.

4 Method

A plug and play control scheme is proposed in this section. This scheme builds on two main components; Reinforcement Learning control of mixing loops and data driven state selection.

4.1 Reinforcement Learning Control of Mixing Loop

Two major aspects of the plug and play control is that it can control the temperature and pressure in an optimal sense, and that it can adapt to various systems. Reinforcement Learning control fits these requirements by adapting towards a control policy that is optimal in the sense of maximizing a return.

In the proposed method a linear function approximation is used to approximate the state-action value function. This is due to both guarantees of convergence and ease of solving for the optimal action. This approximation

has a weight for every feature point

$$\hat{Q}(\mathbf{s}, \mathbf{a}, \mathbf{w}) = \sum_{i=1}^{d} w_i x_i(\mathbf{s}, \mathbf{a}). \tag{D.18}$$

Here the dimension d is the number of feature points and weights. The state vector, $\mathbf{s}$ has the dimension n_s and the action vector, $\mathbf{a}$ has n_a.

Radial basis functions are used for their smoothness and differentiability. The basis functions are centred in the feature points located in $\mathbf{c} = [c_{k_s,1}, ..., c_{k_s,n_s}, c_{k_a,1}, ..., c_{k_a,n_a}]$ with the
feature width $\boldsymbol{\varsigma} = [\varsigma_{k_s,1}, ..., \varsigma_{k_s,n_s}, \varsigma_{k_a,1}, ..., \varsigma_{k_a,n_a}]$

$$x_i(\mathbf{s}, \mathbf{a}) = \exp\left(- \sum_{k_s=1}^{n_s} \frac{(s_{k_s} - c_{k_s,i})^2}{2\varsigma_{k_s,i}^2} - \sum_{k_a=n_s+1}^{n_a+n_s} \frac{(a_{k_a} - c_{k_a,i})^2}{2\varsigma_{k_a,i}^2} \right). \tag{D.19}$$

The value function contains multiple states. Each state is divided into 10 points between minimum and maximum measured values s_{min} and s_{max}. Where the points intersect a feature point is located. The leads to $10^{n_a+n_s}$ feature points with $\mathbf{c} \in \mathbb{R}^{n_a+n_s}$. All weights in the value function are initiated to zero. Apprenticeship learning is used where a commercially available industrial controller is used to control the mixing loop for an initial period to gain some initial training of the reinforcement learning controller before taking over. To achieve knowledge sharing a temporal difference error used is based on a σ parametrisation, presented in [25], which provides a way of shifting between on policy and off policy.

$$\delta_t^{\sigma} = \sigma_t \delta_t^{S} + (1 - \sigma_t)\delta_t^{Q}. \tag{D.20}$$

By setting $\sigma = 0$ and letting the reinforcement learning algorithm train on data logged in the initial period a transfer of knowledge can be done.

To implement the reinforcement learning as a multistep method a dutch trace as proposed in [7] is used as can be seen in Algorithm 7. As described in Section 2 the flow dependent delay changes the horizon of when actions impact the reward, e.g. the return temperature response to a change in forward temperature. The idea is to have the trace decay be proportional to the flow. In (D.2) a multiple pipe system is described. Here a lumped pipe volume approximation is used. This means that only the volume where the impact on the input-output delay is highest is used. The trace decay is at every sample computed as

$$\lambda(q_\eta) = \frac{\phi}{q_\eta(t)}, \tag{D.21}$$

Where $\phi \in [0, 1]$ is a constant that is empirically determined as a function of lumped volume in the relation between forward temperature and return

temperature. $q_\eta(t) \in [q_{\eta,min}, 1]$ is the flow normalized by the maximum flow.

$$\phi = h(v_\eta). \tag{D.22}$$

A normalisation with respect to the maximum flow of the system is done on the flow and the lumped volume

$$v_\eta = \frac{v}{q_{max}}, \qquad q_\eta(t) = \frac{q(t)}{q_{max}} \tag{D.23}$$

A description of the lumped volume v is given here. Take an example of a system with no terminal units and only pipe connections between supply and return of the mixing loop. Here the return temperature is a function of the forward temperature acting at different delays due to different pipe routes

$$T_r(t) = h\left(\mathbf{T}_f, \mathbf{q}\right) \tag{D.24}$$

where

$$\begin{aligned} \mathbf{T}_s &= \left[T_s\left(t - \frac{V_1}{q_1}\right), ..., T_s\left(t - \frac{V_N}{q_N}\right)\right]^T \\ \mathbf{q} &= [q_1, ..., q_N]^T. \end{aligned} \tag{D.25}$$

The individual flows in the different pipe routes are not known in this application; only the total flow leaving the mixing loop is known. Therefore a flow ratio β is introduced where the sum flow ratios for p pipe routes is

$$\begin{aligned} \sum_{N=1}^{p} \beta_N &= 1 \\ q_N(t) &= \beta_N q(t) \end{aligned} \tag{D.26}$$

The ratio of flow β is assumed constant, which is a necessary assumption due to only the main flow being known. The terminal units are controlled by regulating valves which can change how the flow ratios are distributed. Changes to outside temperature might change little in ratios due to affecting all zones, were solar radiation only hitting one side of a building might change the ratios more and make the approximation of the assumption less accurate depending on the specific building. Now v_N is defined as

$$v_N = \frac{V_N}{\beta_N}, \tag{D.27}$$

Applying this to the example in (D.25) gives

$$\mathbf{T}_s = \left[T_s\left(t - \frac{v_1}{q}\right), ..., T_s\left(t - \frac{v_N}{q}\right)\right]^T \tag{D.28}$$

The delay goes towards infinity as the flow goes to zero. Therefore a minimum flow threshold is used. How the flow compensation variables v and ϕ are determined can be seen in the full overview of how the different methods tie together in the proposed algorithm.

The reinforcement learning algorithm with added flow variable λ can be seen in Algorithm 7.

Result: Online $Q_\phi(\sigma, \lambda)$
Initialize : Weights $\mathbf{w}$, trace vector $\mathbf{z}$. Take action $\mathbf{a}'$ according to ϵ-greedy $\pi(.|\mathbf{s}_0)$. Calculate feature state $\mathbf{x} = \mathbf{x}(\mathbf{s_0}, \mathbf{a}')$. $Q_{old} = 0$
Parameters : ϵ, α, γ, ϕ, σ
repeat every sample
  Observe r and $\mathbf{s}'$
  Choose $\mathbf{a}'$ according to ϵ-greedy π
  $\mathbf{x}' \leftarrow \mathbf{x}(\mathbf{s}', \mathbf{a}')$
  $Q \leftarrow \mathbf{w}^T\mathbf{x}$
  $Q'_S \leftarrow \mathbf{w}^T\mathbf{x}'$
  $Q'_Q \leftarrow \max_{\mathbf{a}'} (\mathbf{w}^T\mathbf{x}(\mathbf{s}', \mathbf{a}'))$
  $\delta^\sigma \leftarrow \sigma(r + \gamma Q'_s - Q) + (1 - \sigma)(r + \gamma Q'_Q - Q)$
  Observe flow q
  if $q_{max} \leq q$ **then**
    $q_n \leftarrow 1$
  else if $q \leq q_{min}$ **then**
    $q_n \leftarrow q_{min}/q_{max}$
  else
    $q_n \leftarrow q/q_{max}$
  end
  $\lambda \leftarrow \frac{\phi}{q_n}$
  $\mathbf{z} \leftarrow \gamma\lambda\mathbf{z} + (1 - \alpha\gamma\lambda\mathbf{z}^T\mathbf{x})\mathbf{x}$
  $\mathbf{w} \leftarrow \mathbf{w} + \alpha(\delta^\sigma + Q - Q_{old})\mathbf{z} - \alpha(Q - Q_{old})\mathbf{x}$
  $Q_{old} \leftarrow \sigma Q'_S + (1 - \sigma)Q'_Q$
  $\mathbf{x} \leftarrow \mathbf{x}'$
  Take action $\mathbf{a}'$
until *Mixing Loop Stop*;

Algorithm 7: Algorithm $Q_\phi(\sigma, \phi)$

For the operation of finding solutions to problems such as $\max_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a}, \mathbf{w})$ different solvers can be used. Here a function approximation which is linear in the weights, differential and smooth is used. In this work a search algorithm was designed, which utilizes the knowledge of location of feature points in the radial basis network to make multiple local gradient searches for finding a global maximum. Due to scope of this paper, this solver will

not be further introduced.

4.2 State Selection

For the reinforcement agent to be able to learn it needs to be able to predict the future return from the states and actions. This means that the states should hold enough information to be able to make a reasonable prediction of the return (D.14). For building heating and cooling via mixing loop this is dependent on the specific building in which the mixing loop is installed. One building may have large windows where an observation of solar radiation gives information of free heat as described in (D.1). Another building might be poorly insulated and leaky, where observations of wind speeds has more information. It can be argued that if all available inputs are fed into the Reinforcement Learning agent it would still converge if the needed information is available. However, using input variables that holds no information or even redundant information would decrease the learning rate of the algorithm due to the curse of dimensionality; where-as the dimension of the input set rises linearly, the total volume of the model domain increases exponentially [2]. A data driven state selection is proposed such that they are chosen according to the specific building. The problem of automatic state selection is here handled as a prediction problem where a variable set is sought that can predict the return.

The reinforcement learning method that is applied in this work uses the action value function where a prediction of the expected return is done as a function of the state that the system is in and the action that is taken. The actions space is for the mixing loop the pump speed and the forward temperature. Since these give information towards the prediction of the return, this information is removed from the prediction target before choosing the input variables.

Mutual information is here used to value if an input variable has information about the future return.

A discrete approximation of mutual information is used with estimations of the probability density functions f using m samples of the input variable $\mathcal{X}$ and the return $\mathcal{G}$

$$I(\mathcal{X};\mathcal{G}) \approx \frac{1}{m}\sum_{i=1}^{m} \log\left(\frac{f(x_i, G_{t,i}^{\lambda(q_i)})}{f(x_i) f(G_{t,i}^{\lambda(q_i)})}\right). \tag{D.29}$$

Using a kernel density estimator with the Parzen window [16] to estimate the probability density functions

$$\hat{f}(\mathcal{X},\mathcal{G}) = \frac{1}{m}\sum_{i=1}^{m} K_{\mathbf{H}}\left(\begin{bmatrix} x_f \\ G_f \end{bmatrix} - \begin{bmatrix} x_i \\ G_{t,i}^{\lambda(q_i)} \end{bmatrix}\right). \tag{D.30}$$

Where x_f and G_f are the center points.

The bivariate Gaussian kernel is used as the kernel function [11]

$$K_{\mathbf{H}}(\mathbf{k}) = \frac{1}{\sqrt{(2\pi)^2 |\mathbf{H}|}} \exp\left(-\frac{1}{2}\mathbf{k}^T \mathbf{H}^{-1} \mathbf{k}\right). \tag{D.31}$$

Where $\mathbf{k}$ contains the distances of the samples from the center points. In this work an often used bandwidth matrix $\mathbf{H}$ for bivariate data is used

$$\mathbf{H} = h^2 \begin{bmatrix} S_x^2 & S_{xG} \\ S_{xG} & S_G^2 \end{bmatrix}. \tag{D.32}$$

Here S_{xG} holds the covariance between the input data and the return. S_x^2 and S_G^2 are the variances of the samples [22]. A gaussian reference bandwidth [19] is used that in the bivariate case is

$$h = \left(\frac{1}{4}\right)^{\frac{1}{6}} \sigma m^{\frac{-1}{6}}, \tag{D.33}$$

where σ is the standard deviation of the sample data.

After having found the first input variable containing highest mutual information of the return a second input variable is sought that gives highest mutual information after the first is already given. The information already given by the first chosen input z is removed by making an estimation of the return using only the chosen input and then subtracting that from the return and the remaining inputs, leaving a set of residuals. Here an example of input variable $\mathcal{X}_1$ being chosen first and then the input variable $\mathcal{X}_2$ is examined for partial mutual information.

$$\begin{aligned} u_{t:T} &= G_{t:T}^{\lambda(q_t)} - \mathbb{E}[G_{t:T}^{\lambda(q_t)} | x_{1,t:T}] \\ v_{t:T} &= x_{2,t:T} - \mathbb{E}[x_{2,t:T} | x_{1,t:T}] \end{aligned} \tag{D.34}$$

Where $u_{t:T}$ and $v_{t:T}$ are residuals of respectively the return and the second explored input. While $G_{t:T}^{\lambda(q_t)} = [G_t^{\lambda(q_t)}, G_{t+1}^{\lambda(q_{t+1})}, ..., G_{t+T}^{\lambda(q_{t+T})}]$ is a time series of $\lambda(q_t)$ averaged returns, $x_{1,t:T}$ is a time series of the chosen input containing highest mutual information and $x_{2,t:T}$ is a time series for the second explored input. The estimators chosen as radial basis neural networks to match the function approximation of the Reinforcement Learning agent. Finding the second input variable with the now highest partial mutual information with prior knowledge of the variable containing highest mutual information can then be done as

$$I(\mathcal{X}_2; \mathcal{G} | \mathcal{X}_1) \approx I(v_{t:T}; u_{t:T}) \tag{D.35}$$

Due to the long delays of the mixing loop application, choosing the input variable giving the highest mutual information is not only a question of

which variable, but also what time delay holds most information.

$$\max_{j,d} \quad I\left(\mathbf{x}^{j}_{t-d:T-d}; G^{\lambda(q_t)}_{t:T}\right), \tag{D.36}$$

where j is the index of the input in the full set of inputs $\mathbf{X}$ and d is a delay that the times series is shifted by. Due to the flow dependent delay the time delays at which input variables hold information change with the flow. By applying this flow dependent delay the highest mutual information is then searched for at different v.

$$\max_{j,v} \quad I\left(\mathbf{x}^{j}_{t-v/q_t:T-v/q_t}; G^{\lambda(q_t)}_{t:T}\right), \tag{D.37}$$

To avoid the delay going towards infinity as the flow goes to zero a minimum flow is implemented.

After having gone through iterations of choosing inputs via flow dependent partial mutual information at one point further inputs will not add any new knowledge so the algorithm can be stopped. The stopping criteria used for this work is based on cross validation of the estimation model. In Algorithm 8 the state selection is done based on two sets of data, training data for the input selection and validation data for stopping criteria.

4.3 Method overview

In Fig. D.2 an overview of the proposed plug and play control scheme divided into submodules can be seen. The algorithm is represented on pseudo code form in Algorithm 9.

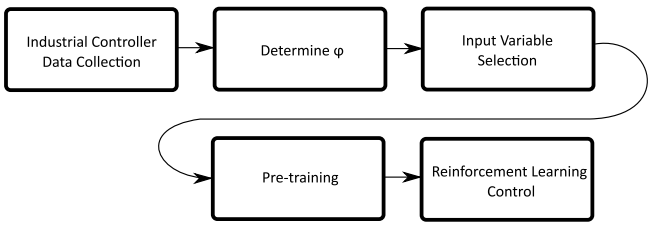

Fig. D.2: Plug and play control scheme

First data is gathered for state selection as well as validation data for the stopping criteria. For the flow compensation used in the reinforcement learning agent the parameter ϕ is needed which is described as a function of v from the forward-return temperature relation. The v that contains the highest mutual information between forward temperature and return temperature, here called v^*, is used to determine ϕ. In [15] a linear relation between $v*$ and the ϕ giving the best performance was found empirically based on a

Result: State Selection
Initialize : Load training data of all inputs $\mathbf{x}_{t:t+m_s}$, return $G^{\lambda(q)}_{t:t+m_s}$ and flow $q_{t:t+m_s}$. Load validation data of n inputs $\mathbf{x}_{t:t+m_v}$, return $G^{\lambda(q)}_{t:t+m_v}$ and flow $q_{t:t+m_v}$.
Parameters : tol
repeat

Find input with highest mutual information as

$$z_{s,t:m_s} \leftarrow \max_{j,v} \; I\left(\mathbf{x}^j_{t-v/q:t+m_s-v/q}; G^{\lambda(q)}_{t:t+m_s}\right)$$

Generate estimators $\mathbb{E}[G^{\lambda(q)}_{t:t+m_s}|z_{s,t:m_s}]$

and $\mathbb{E}[\mathbf{x}_{t:t+m_s}|z_{s,t:m_s}]$

Calculate residuals as

$$G^{\lambda(q)}_{t:t+m_s} \leftarrow G^{\lambda(q)}_{t:t+m_s} - \mathbb{E}[G^{\lambda(q)}_{t:t+m_s}|z_{s,t:m_s}]$$

$$\mathbf{x}_{t:t+m_s} \leftarrow \mathbf{x}_{t:t+m_s} - \mathbb{E}[\mathbf{x}_{t:t+m_s}|z_{s,t:m_s}]$$

Add z to set of selected inputs **z**

$$RMSE \leftarrow \sqrt{\frac{\sum_{t=1}^{m_v}\left(G^{\lambda(q)}_{t:t+m_v} - \mathbb{E}[G^{\lambda(q)}_{t:t+m_v}|\mathbf{z}_{v,t:t+m_v}]\right)^2}{m_v}}$$

$$RMSE_{prev} \leftarrow RMSE$$

until $tol > \dfrac{RMSE_{prev} - RMSE}{RMSE_{prev}}$;

Algorithm 8: Algorithm for state selection

generic mixing loop system model. When ϕ has been determined, this can be used to compute the returns with variable λ from the inputs logged during the first month. During the state selection, all inputs are tested at different flow dependent delays v to determine the highest mutual information with respect to the return $G^{\lambda(q)}_t$. The validation data is used to test if more inputs should be added. Besides being used for the state selection, the data logged during the first month is also used for pre-training of the Reinforcement Learning agent. Since the data is gathered off policy $\sigma = 0$ during this pre-training. Both due to the data being off policy and the industrial controller not utilising any exploration this training is less effective than the later online training. The final step is to let the reinforcement learning controller with flow variable eligibility trace take over control of the mixing loop.

Result: Plug and Play Control Scheme
Initialize : Industrial Controller
Parameters : t_{train}, t_{vali}
repeat
  Industrial standard mixing loop control
  Log input variables $\mathbf{x}_s$, actions $\mathbf{a}_s$
  and rewards r_s
until *Runtime* = t_{train};
repeat
  Industrial standard mixing loop control
  Log input variables $\mathbf{x}_v$, actions $\mathbf{a}_v$
  and rewards r_v
until *Runtime* = t_{vali};

Determine v^* as $\max_v \; I\left(T_{s,t-v/q:T-v/q}; T_{r,t:T}\right)$

Determine ϕ from v^*

Use ϕ to compute $G_{t:T}^{\lambda(q)}$ from logged data

Do state selection via Algorithm 2

Pretrain Reinforcement Learning agent with selected states off-policy $\mathrm{Q}_\phi(0,\lambda)$ using data sets $[\mathbf{x}_{t:t+m_s+m_v}, \mathbf{a}_{t:t+m_s+m_v}, G_{t:t+m_s+m_v}^{\lambda(q)}]$

repeat
  Reinforcement Learning $\mathrm{Q}_\phi(\sigma,\lambda)$ mixing loop control as in Algorithm 1
until *Runtime* = ∞;

Algorithm 9: Reinforcement Learning with data driven state selection

5 Test Setup

Testing building HVAC systems is time consuming due to slow dynamics and the need to test during multiple conditions such as changing weather or usage patterns. When comparing two controllers' performance on the same building, the conditions should be the same for a good benchmark. This can be nearly impossible to obtain due to many sources of disturbances. To be able to test multiple controller settings in the same load scenario within a feasible test time, a hardware-in-the-loop setup is designed. The setup consists of a hardware part where a mixing loop system is supplying a heat exchanger being cooled by a chiller. By controlling flows and temperature, a software-based load model can emulate a building's behaviour. The load model is based on data logged from an actual office building as seen in Fig. D.3. The playback speed of the load model is then increased such that the

Fig. D.3: Office building used for gathering data. Heating supplied via single mixing loop.

dynamics of the building react faster, but still slow enough that the hydraulics of the hardware part are accounted for. To further increase the speed of testing and ease of comparison multiple parallel hardware in the loop test are run under same load conditions, but with different controllers.

5.1 Building Model

The building model is made from data logged in the office building pictured in Fig. D.3 situated in Bjerringbro in Denmark. The data points logged can be seen in Table D.1. A nonlinear autoregressive external input neural network with sigmoid basis functions was trained on one year of data and then validated on 3 months of data. The model outputs zone temperatures, return temperature and flow. The model flow was scaled to match the test setup by the constant $C = q_{max,testsetup}/q_{max,building}$.

5.2 Hydraulics

In Fig. D.4 the schematic and a picture of the test setup can be seen. The load emulation is achieved by controlling valve V_2 such that the flow q_1 matches the flow of the building model. Valve V_3 is controlled such that temperature T_1 matches the return temperature of the building model. A boiler supplies the hot water for heating via the mixing loop and a chiller supplies the cold water that is used to emulate the load. All valves and pumps are fitted with local set point control. Set points for pump P speed and valve $V1$ forward temperature are supplied by either the reinforcement learning agent or the industrial controller.

5.3 Controllers

The proposed algorithm is implemented along with the building model on a microprocessor that resembles what is used in mixing loop control to ensure

Data points	Description
t_d	Time of day
T_{zN}, T_{zE}, T_{zS}, T_{zW}	Zone temp.
T_f	Forward temp.
T_r	Return temp.
N_p	Pump Speed
q	Flow
T_{om}	Outside temp. measured
T_{op}	Outside temp. predicted
S_m	Solar radiation measured
S_p	Solar radiation predicted
W_m	Wind speed measured
W_p	Wind speed predicted
C_{zN}, C_{zE}, C_{zS}, C_{zW}	CO_2 levels in zones
OW_N, OW_E, OW_S, OW_W	Windows open or closed
OD_N, OD_E, OD_S, OD_W	Radiator Valves Opening Degree
F_N, F_E, F_S, F_W	Air fan speeds
T_{aN}, T_{aE}, T_{aS}, T_{aW}	Air supply temp.

Table D.1: Data points

the feasibility of implementation of the algorithms with regards to factors such as memory and processor speed.

The algorithm is compared with an industrial grade controller for mixing loops. The industrial controller used is the one installed in the Building Management System (BMS) of the building, which is used in the model. The control was made by a BMS manufacturing company that will be kept anonymous. To ensure a fair comparison, the parameters in the industrial controller were tuned to achieve higher performance.

6 Results

The results of the test will be presented and discussed in the following. First results from finding the flow compensation parameters are shown. In Fig. D.5 the mutual information between the forward temperature and the return temperature can be seen at different values of v with the highest value being at $v^* = 0.16$. When compensated by the maximum flow of the system this leads to $\phi = 0.8$ by using the empirecally found linear relation in [15]. By using the determined ϕ, the return from the logged data can be calculated. This return is used for the state selection. In Fig. D.6 the return prediction error of the validation data is shown as a function of chosen inputs at the lumped volume giving the highest mutual information. The prediction error

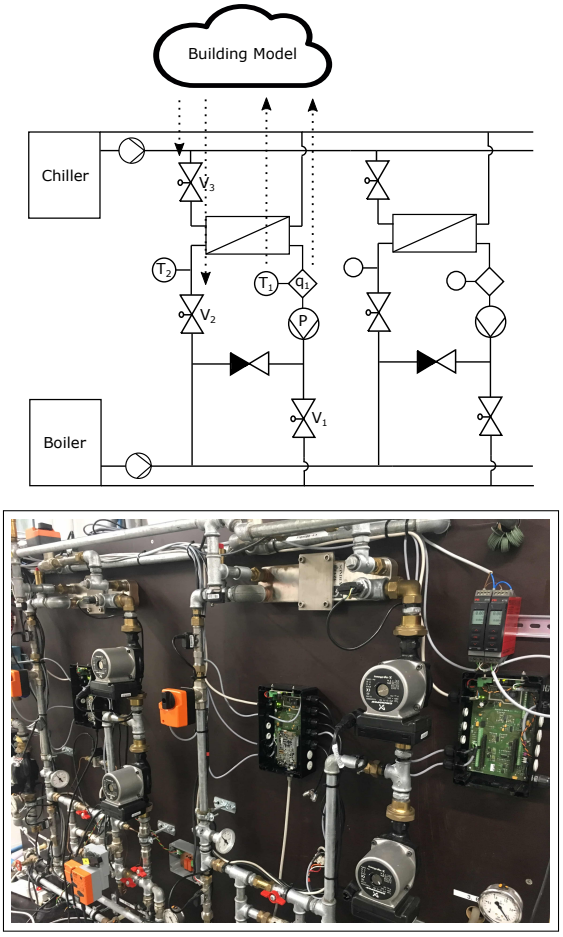

Fig. D.4: Schematic and picture of 2 out of 4 mixing loop configurations.

for the first 15 inputs (including the two actions) are shown. The stopping criteria, stopped the input selection such that the first 8 inputs are used in the Reinforcement Learning agent. The first two inputs are the actions forward temperature and pump speed. The next is zone temperature of the eastern zone, then flow, return temperature, time of day, outdoor measured temperature and the measured wind speed.

To see how the chosen state space performs compared to using more or less states the test setup was run using different amounts of states. In the first plot of Fig. D.7 the normalised sum of weights is shown, where n_s is the number of used states. The plot shows, that the lower the number of states the faster the weights converge. In the second plot of Fig. D.7 the return of using different state spaces can be seen for the same training period. The state

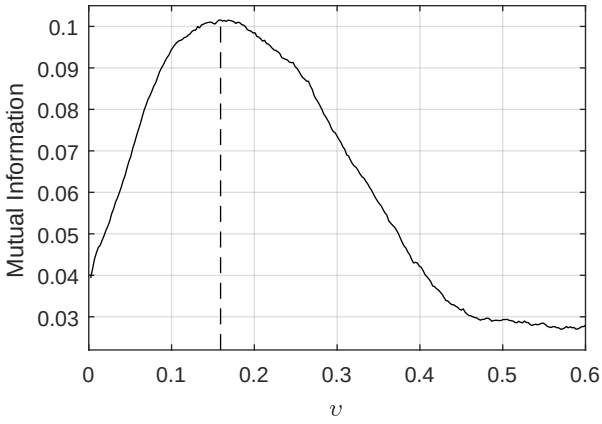

Fig. D.5: Mutual information between forward temperature and return temperature as a function of υ

space chosen by the plug and play control scheme ($n_s = 6$) has the highest return during this time period. It is expected that versions with larger state spaces will in time converge and give same or higher return as the chosen variation. However it is deemed that for an application such as a mixing loop a convergence time that takes much longer than the 40 days is not practically usable.

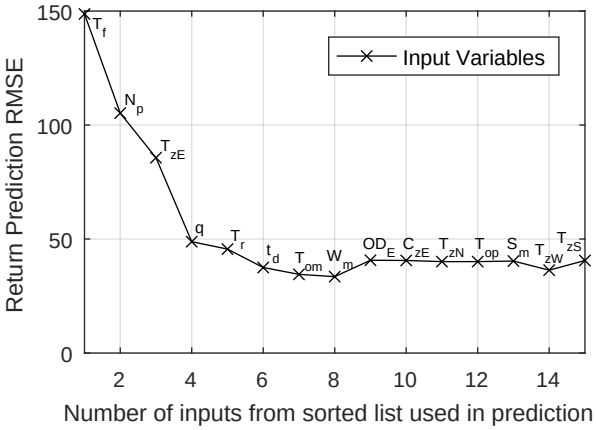

Fig. D.6: Return prediction error as function on number of inputs from sorted list. The inputs are shown at the lumped volume which leads to highest mutual information.

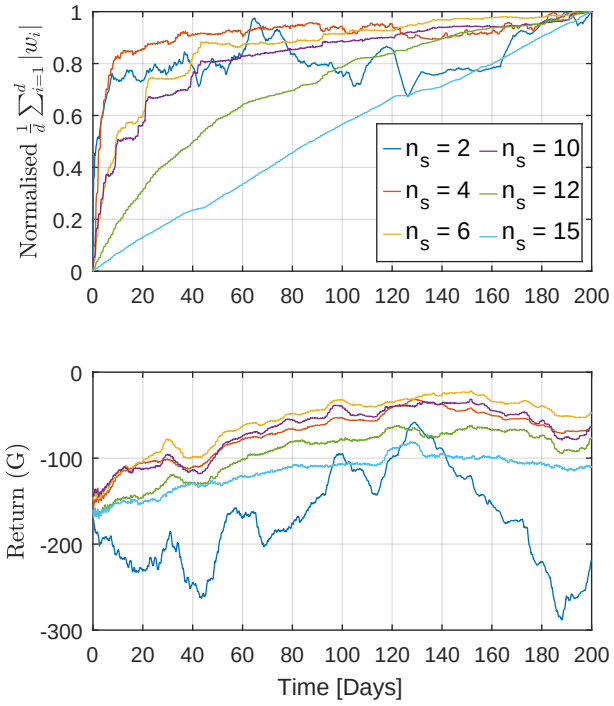

Fig. D.7: Comparison of returns and weight convergence for different sets of states.

In Fig. D.8 the proposed method is compared to the industrial controller.

In the first plot the root mean square temperature error of all the building zones combined are shown. After the Reinforcement Learning agent takes over the control after the initial 30 days the temperature error increases. This time instant is when the reinforcement learning control has taken over from the industrial controller and is beginning to learn the system. After 30 days of training the temperature error improves over the industrial controller.

The second plot shows the running cost of 14 days of operation. Here savings are starting from the initialisation of the reinforcement learning agent, but at the cost of comfort as can be seen in the first plot. The third plot shows the return, which the reinforcement learning agent tries to maximise. Here it can be seen that around day 55 the reinforcement learning agent overtakes the industrial controller in regards to defined return. The last plot shows the summation of the absolute values of the weights. Here it can be noticed that the pre training done using the logged data from the first 30 days only brings this summation of the absolute values of the weights a small step towards the value it converges towards later. This makes sense since the industrial controller only explores a small area of the state-action space.

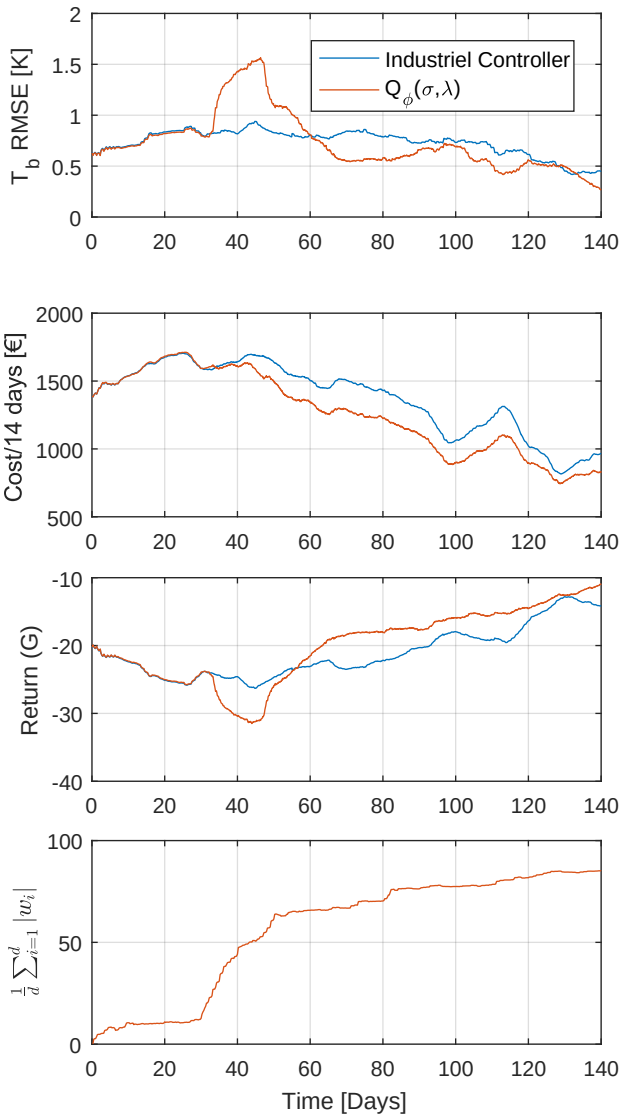

Fig. D.8: Comparison of Building Temperature RMSE with a 14 days running window. Operating cost for running 14 days. The return and the summation of absolute weights.

At day 60 (30 days data logging for state selection + 30 days of training) the reinforcement learning agent has reached a satisfactory performance. From day 60 to day 140 the reinforcement learning agent saves 14% on cost, while having 19% less temperature error.

7 Conclusion

A plug and play control scheme was introduced for mixing loop control in building heating. The two main components of the control scheme are state selection and Reinforcement Learning.

Due to a large set of sensor inputs being available a state variable selection method is employed to improve training speed of the self learning control. The state selection is build using partial mutual information with flow compensation due to the flow varying delays of the mixing loop application.

To achieve self learning optimal control a Reinforcement Learning agent with flow dependent eligibility trace was proposed. By using the subset of input variables as state information in the Reinforcement Learning agent training a shorter training time is achieved.

A hardware-in-the-loop setup was used to test the proposed controller against controllers utilizing another set of states and an industrial controller.

The results showed that the proposed method has the ability to chose a set of states that has a good performance along a training speed which is satisfactory for the mixing loop application.

The performance during the first 30 days of the algorithm is determined by the industrial controller. After data has been gathered and the reinforcement learning agent takes over control it takes around 25 days for agent to achieve same performance level as the industrial controller. After this initial period the reinforcement learning agent improves further on performance and from day 60 to 140 the temperature error is reduced by 19% while saving 14% on operational cost.

References

[1] A. Afram and F. Janabi-Sharifi, "Theory and applications of HVAC control systems - A review of model predictive control (MPC)," 2014.

[2] R. Bellman, "Adaptive control processes: A guided tour," *Princeton University Press*, vol. 28, pp. 1–19, 1961. [Online]. Available: http://arxiv.org/abs/1302.6677

[3] J. Cigler and S. Prívara, "Subspace identification and model predictive control for buildings," in *11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010*, 2010, pp. 750–755.

[4] K. Dalamagkidis, D. Kolokotsa, K. Kalaitzakis, and G. S. Stavrakakis, "Reinforcement learning for energy conservation and comfort in buildings," *Building and Environment*, vol. 42, no. 7, pp. 2686–2698, 2007.

[5] L. Eller, L. C. Siafara, and T. Sauter, "Adaptive control for building energy management using reinforcement learning," in *Proceedings of the IEEE International Conference on Industrial Technology*, vol. 2018-Febru, 2018, pp. 1562–1567.

[6] P. Fazenda, K. Veeramachaneni, P. Lima, and U. M. O'Reilly, "Using reinforcement learning to optimize occupant comfort and energy usage in HVAC systems," *Journal of Ambient Intelligence and Smart Environments*, vol. 6, no. 6, pp. 675–690, 2014.

[7] R. S. S. Harm van Seijen, A. Rupam Mahmood, Patrick M. Pilarski, Marlos C. Machado, "True Online Temporal-Difference Learning," *Journal of Machine Learning Research*, vol. 17, pp. 1–40, 2016.

[8] International Energy Agency, *The Future of Cooling*, 2018.

[9] X. Li, H. R. Maier, and A. C. Zecchin, "Improved PMI-based input variable selection approach for artificial neural network and other data driven environmental and water resource models," *Environmental Modelling & Software*, vol. 65, pp. 15–29, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364815214003545

[10] R. May, G. Dandy, and H. Maier, "Review of Input Variable Selection Methods for Artificial Neural Networks," *Artificial Neural Networks - Methodological Advances and Biomedical Applications*, no. August 2016, p. 362, 2011.

[11] R. J. May, H. R. Maier, G. C. Dandy, and T. M. K. G. Fernando, "Non-linear variable selection for artificial neural networks using partial mutual information," *Environmental Modelling & Software*, vol. 23, no. 10-11, pp. 1312–1326, 2008.

[12] E. Mills, H. Friedman, T. Powell, N. Bourassa, D. Claridge, T. Haasl, and M. A. Piette, "The cost-effectiveness of commercial-buildings commissioning," *HPAC Engineering*, 2005.

[13] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu, and J. G. Slootweg, "On-line Building Energy Optimization using Deep Reinforcement Learning," 2018.

[14] A. Overgaard, C. S. Kallesoe, J. D. Bendtsen, and B. K. Nielsen, "Input selection for return temperature estimation in mixing loops using partial mutual information with flow variable delay," in *1st Annual IEEE Conference on Control Technology and Applications, CCTA 2017*, vol. 2017-Janua, 2017, pp. 1372–1377.

[15] A. Overgaard, C. S. Kallesøe, J. D. Bendtsen, and B. K. Nielsen, "Mixing Loop Control using Reinforcement Learning," in *Proceedings of the 13th REHVA World Congress CLIMA 2019*, 2019, pp. Accepted, not yet published.

[16] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization (Wiley Series in Probability and Statistics)*, 1992, vol. 156.

[17] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. July 1928, pp. 379–423, 1948. [Online]. Available: http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

[18] A. Sharma, "Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 - A strategy for system predictor identification," *Journal of Hydrology*, vol. 239, no. 1-4, pp. 232–239, 2000.

[19] B. Silverman, "Density estimation for statistics and data analysis," *Chapman and Hall*, vol. 37, no. 1, pp. 1–22, 1986.

[20] R. S. Sutton, "Learning to Predict by the Methods of Temporal Differences," *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1988.

[21] R. S. Sutton and A. G. Barto, "Reinforcement learning: an introduction 2018 complete draft," *UCL,Computer Science Department, Reinforcement Learning Lectures*, p. 1054, 2017.

[22] M. P. Wand and M. C. Jones, "Comparison of smoothing parameterizations in bivariate kernel density estimation." *Journal of the American Statistical Association*, vol. 88, no. 422, pp. 520–528, 1993. [Online]. Available: http://www.jstor.org/stable/2290332

[23] C. J. C. H. Watkins, "Learning from Delayed Rewards," *Ph.D. thesis, Cambridge University*, 1989.

[24] T. Wei, Y. Wang, and Q. Zhu, "Deep Reinforcement Learning for Building HVAC Control," in *Proceedings of the 54th Annual Design Automation Conference 2017 on - DAC '17*, 2017, pp. 1–6. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3061639.3062224

[25] L. Yang, M. Shi, Q. Zheng, W. Meng, and G. Pan, "A unified approach for multi-step temporal-difference learning with eligibility traces in reinforcement learning," in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2018-July, 2018, pp. 2984–2990.

[26] J. C. M. Yiu and S. Wang, "Multiple ARMAX modeling scheme for forecasting air conditioning system performance," *Energy Conversion and Management*, 2007.

Paper E

Technical Report on instrumentation of office building for data collection

Anders Overgaard

This paper has not been published
2017

Abstract

This technical report outlines some of the considerations and highlights relating to the data collection in an office building which is used throughout this PhD project.

1 Introduction

To get HVAC data from a real office building a data logger was installed in the office building "Nord 2". A building management system (BMS) is already installed in Nord 2 with access to multiple HVAC relevant sensors around the building. By connecting to the already existing BUS network, installing further sensors and getting external weather forecast from the internet a large data foundation for analysing HVAC was gathered. This data has been used for validating the input variable selection method in Paper A. The data has been further used to create a load model for the hardware-in-the-loop test setup described in [Appendix F].

2 Office Building

To gather data from an actual building the office building "Nord 2" located at Grundfos in Bjerringbro Denmark is used, see Fig. E.1.

Fig. E.1: Picture from the front side of office building Nord 2.

The office building has approximately $8000m^2$ floorspace divided over 3 floors with windows mainly facing south. The floorspace is used for multiple purposes with the main use being office space. Other uses in the building is meeting rooms, canteen, showroom, tea kitchens, toilets and changing rooms. The HVAC system consists mainly of radiator heating with a ventilation system. The building HVAC is controlled by a Schneider Building Management System (BMS). The building is divided into 33 different zones with local control for the comfort parameters temperature and CO_2 level. Fig. E.2 shows the control block visualisation in the BMS.

Fig. E.2: Local control for zone temperature via radiator valves and CO_2 level via air duct dampers.

In the case of heating the local temperature control in the zones are done by adjusting the electric radiator valves while the CO_2 level is controlled by air duct dampers in the ventilation system. In case of cooling, this is done by the ventilation system. Five air handling units are used from where the air is fed into the different zones via air ducts. Fig. E.3 shows an example a control block for an air handling unit in the BMS.

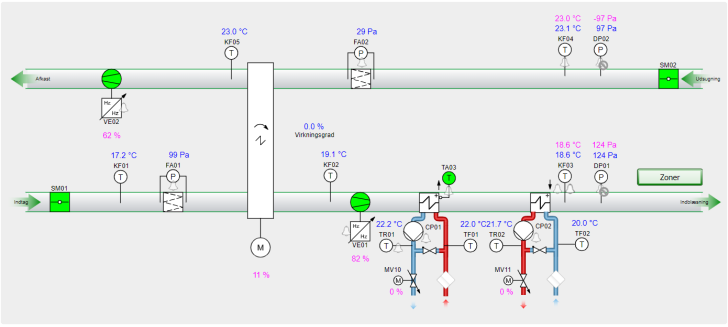

Fig. E.3: Air handling unit control block in BMS

Two thermal power coils are present in the air handling unit. In the case of heating, heat power is added to the inlet air to ensure a proper temperature between 18^0C and 25^0C. If cooling is needed the cooling coil is used to cool the inlet air. In case of cooling the air duct dampers take over temperature control of the zones. Three mixing loops are used in this system. One for the radiators, one for the heating coil in the air handling unit and one for the cooling coil. In Fig. E.4 a control block for a mixing loop in the BMS can be seen.

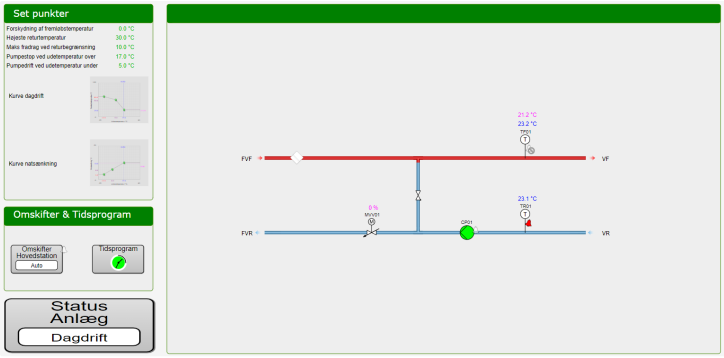

Fig. E.4: Mixing Loop Control block in BMS

3 Data logger

The office building is already fitted with a Schneider BMS that has access to various sensors around the building. To get access and log this data a data logger was made which connects to the BMS via MODBUS and sends the data to the Grundfos cloud solution SYSMON.

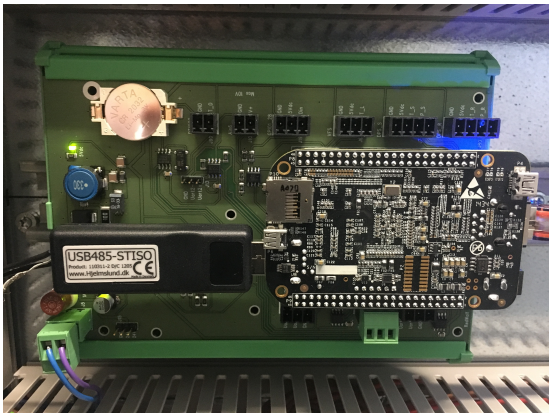

Fig. E.5: Electronics that collects the data and communicates with the cloud. Example of components are microprocessor, MODBUS and Wifi modules.

[H] The data logger, see Fig. E.5, mainly consists of a BeagleBone microprossor, circuit protection, short term power, MODBUS and wifi modules. The data is send to the SYSMON cloud where it can be accessed. Online monitoring can also be done via SYSMON web interface as seen in Fig. E.6. While the main functionality has been to log data the setup is also able to send set point values for control of the different HVAC elements to the BMS

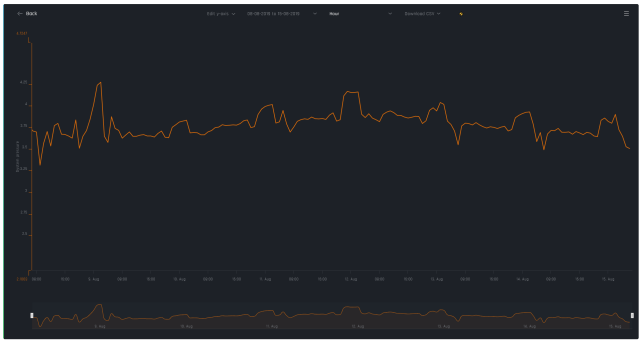

Fig. E.6: Example of web interface for monitoring of the collected data.

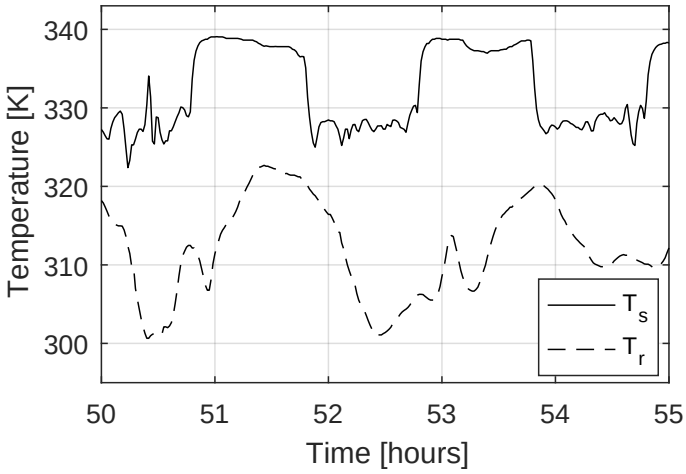

Fig. E.7: Example of fluctuation of input.

which then effectuates it to the local controllers. This is used to improve persistence of excitation in the gathered data. Fig. E.7 shows an example of this where the supply temperature from the radiator mixing loop was fluctuated to see the delayed reaction on the return temperature.

4 Signals

The signals that the data logger gathers consists of the existing signals from the BMS system, added sensors and data from the internet. Weather prediction looking 3 hours ahead are gathered from the internet every hour. The rest of the data is gathered with a sample rate of 10 s at 16 bit. The full list of gathered data is seen in Table E.1.

Logged Identifier	Description
Zone Measurements	
RF01 - RF33	Zone temperatures
CO01 - CO33	Zone CO_2 levels
ZMV01 - ZMV33	Radiator valve opening degree
VAV01 - VAV33	Air duct dampener opening degree
PIR01 - PIR33	Zone PIR sensor
VK01 - VK33	Window open/closed
Mixing Loops Measurements	
TF01 - TF03	Forward temperatures
TR01 - TR03	Return temperatures
MV01 - MV03	Mixing valve opening degree
P1 - P3	Differential pressure over the pump
CP01 - CP03	Pump on/off
AF01 - AF03	Pump flow
AE01 - AE03	Thermal power out of mixing loop
TFP01 - TFP03	Supply temperature primary side
TRP01 - TRP03	Return temperature primary side
Air Handling Units Measurements	
KF01 - KF05	Inlet air temperature
VE01 - VE05	Fan speed
Weather sensors	
SOL	Solar radiation
WS	Wind Speed
WD	Wind direction
TO	Outside dry bulb temperature
Data gathered from internet	
DA	Date
DOW	Day of week
TOD	Time of day
WSP	Predicted wind speed
WDP	Predicted wind direction
SOLR	Predicted solar radiation
TOP	Predicted outside dry bulb temperature
Total data points	246

Table E.1: Data points

5 Conclusion

A data collecting system was installed in the office building "Nord 2" located in Bjerringbro, Denmark. Data is collected from HVAC equipment, building sensors and resources from the internet and stored in a cloud solution. In total 246 data points are logged and used as the data foundation for various analysis and model building.

Paper F

Technical Report for experimental setup for mixing loop control

Anders Overgaard

This paper has not been published
2018

Abstract

This technical report outlines some of the considerations and highlights relating to the hardware-in-the-loop experimental setup for test of mixing loop control. The setup is used throughout this PhD project.

1 Introduction

A hardware-in-the-loop experimental test setup has been developed and used to test and compare control methods. This approach has been used to increase the test speed and comparison between control methods compared to real building HVAC testing.

2 Hardware-in-the-loop

The objective of this setup is to run test of different mixing loop control methods. A hardware-in-the-loop approach is used which is illustrated in Fig. F.1.

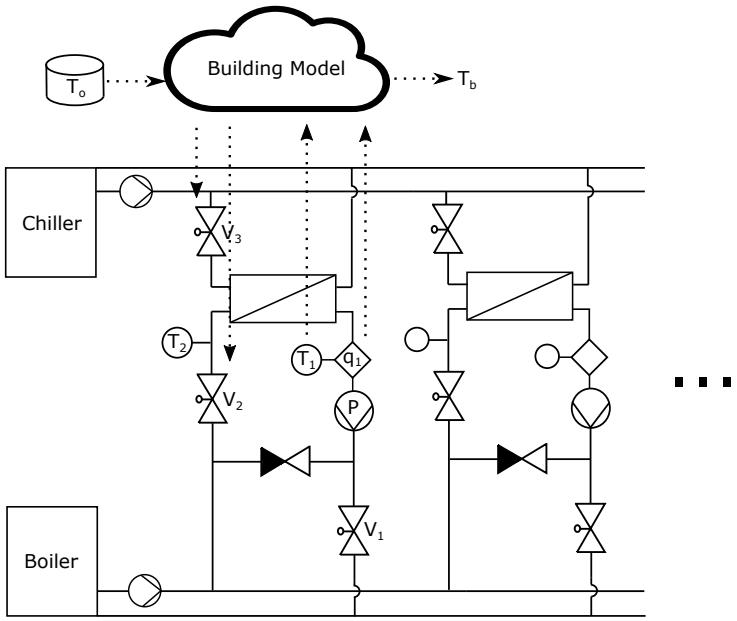

Fig. F.1: Illustration of hardware-in-the-loop test setup.

The setup consists of four mixing loops with a heat supply. The mixing loops each supply a heat exchanger with hot water. Cold water is produced

by a heat pump feeding the load side of the heat exchangers. The flow of the cold water is controlled to emulate a load alike a building heating scenario. The main advantages of this setup compared to testing on a real building is an increased test speed. In mixing loop systems there is a large difference between the hydraulic dynamics and the building thermal dynamics. In this setup the buildings thermal dynamics being emulated is sped up by 12 times compared to real time. At this speed the hydraulic dynamics still runs unaffected due to being faster. When testing building HVAC control another challenge is often creating proper benchmarks since load conditions such as weather and usage of the building change over time. In this test setup four parallel systems are build each having the same load conditions for benchmarking. Another advantage of the setup is that the load model can be change quickly. In this way the load model can contain a different building or the mixing loop feeding into a ventilation system instead of radiators. While speed and easy of benchmarking is obtained by using the proposed test setup it comes at the cost of accuracy and test of unmeasured disturbances. By using a load model to emulate the load the mixing loop system is only tested against what is captured by the specific model. In this way the hardware-in-the-loop setup is considered a good test of methods before choosing one for time consuming field tests.

3 Hardware

In this section the hardware parts are described. In Fig. F.2 the hydraulic schematic of the experimental setup can be seen.

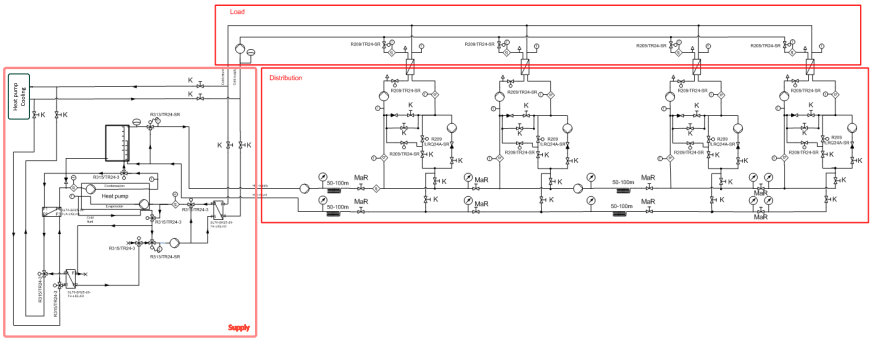

Fig. F.2: Experimental setup.

The hydraulics can be divided into 3 parts: load, distribution and supply. The load system consists of a booster pump and four load sides of heat exchangers. For each load side an electronic valve is placed along with sensors

for flow, supply temperature and return temperature. The distribution system consists of 4 mixing loops which can be changed between using valves or pumps for mixing control. Flow sensors for primary and secondary side flows are installed along with supply, mixing and return temperature sensors. An electronic valve is inserted to control the flow of the heat side of the heat exchanger, which is also used to emulate the load of the system. A booster pump is installed to supply the four mixing loops. Pipe lengths are installed in the system to emulate a building where the four mixing loops serve different zones for test of multiple mixing loop control. In the supply part the heat can be either generated from a heat pump of from an electrical boiler with an accumulating tank. Cold water is generated from a heat pump. In Fig. F.3 a picture overlooking part of the hardware can be seen.

Fig. F.3: Picture overlooking part of the test setup.

To control the different elements in the systems a controller is used for each mixing loops, one for the load system and one for the supply system all connected by CANbus. This controller is a beaglebone microprocessor with various electronic interfaces for pump and valve control, sensor reading and BUS communication as seen in Fig. F.4.

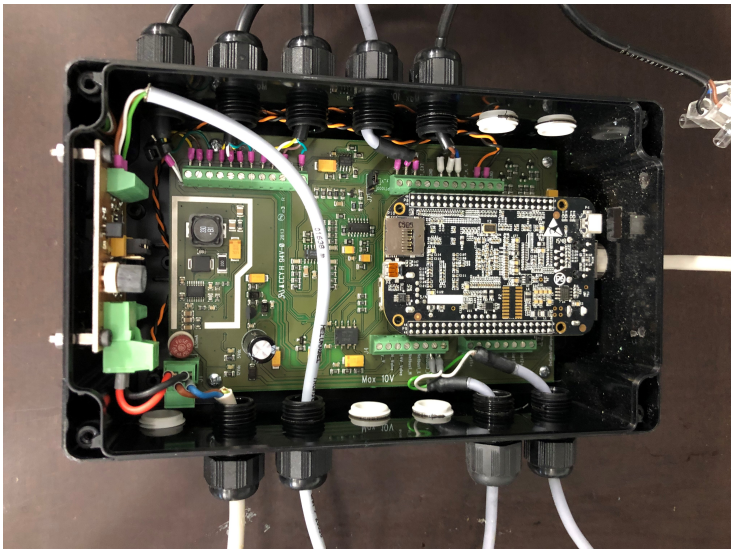

Fig. F.4: Mixing loop control with beagle bone setup.

4 Building Model

Data for generating the load model is gathered from the office building as described in [Appendix E]. The model is a non linear autoregressive neural network structure. Depending on the desired extent of the model different input sets can be used for training. In Fig. F.5 an example of the model with the minimum amount of inputs can be seen.

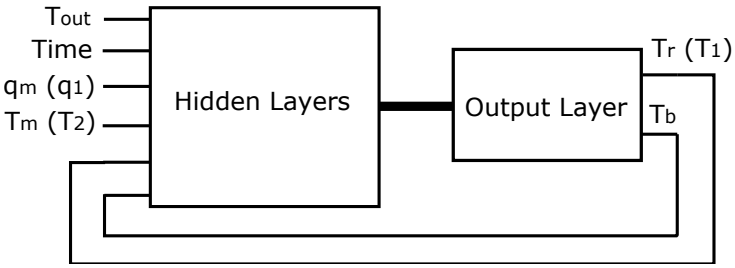

Fig. F.5: Example of neural network model for emulating load.

To validate the models the used data is divided into a training set and a validation set. In Fig. F.6 an example of comparison between some of the validation data and the models building - and return temperature can be seen.

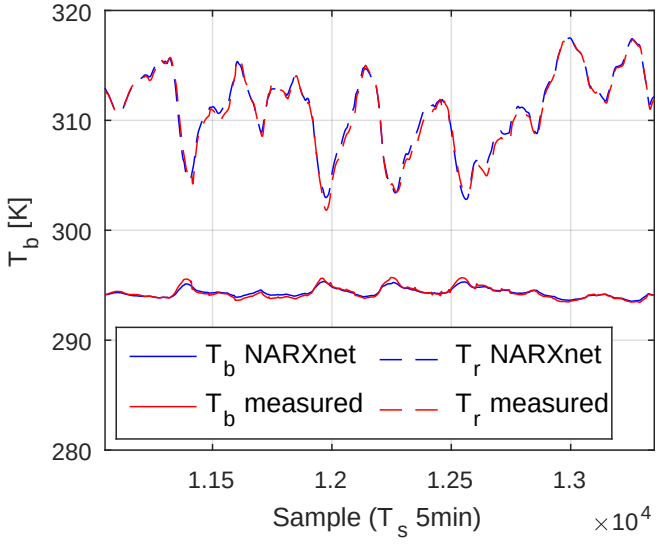

Fig. F.6: Validation of building- and return temperature in the building model.

To emulate the load two different setups has been used. In both setups a PI controller regulates the valve V_3 in Fig. F.1 such that the return temperature T1 reacts according to the building model.

In the first setup valve V_2 controls the building temperature in the building model which is the average of the 33 different zone temperatures. The control of valve V_2 tries to reach the setpoint of 21^{o}C for the building temperature in the building model. By utilizing a PI controller as the radiator controllers in the real building the flow that the mixing loop experiences emulates that of the building. In this setup the measured flow q_1 and the mixing temperature T_2 is part of the input set for the load model that determines the return temperature and the building temperature.

In the other setup the valve V_2 controls the flow to equal that of the building model. In this setup the building model is trained to outbut zone temperatures, flow and return temperature.

Since the load emulation is done by letting measured variables be controlled to the values of the load model there will be a control error. The return temperature controller acting on valve V_3 proved to be the most challenging local controller due to flow dependence of the gain, slow control valve and fast dynamics. In Fig. F.7 an example of the measured return temperature T_1 is compared with the return temperature from the load model.

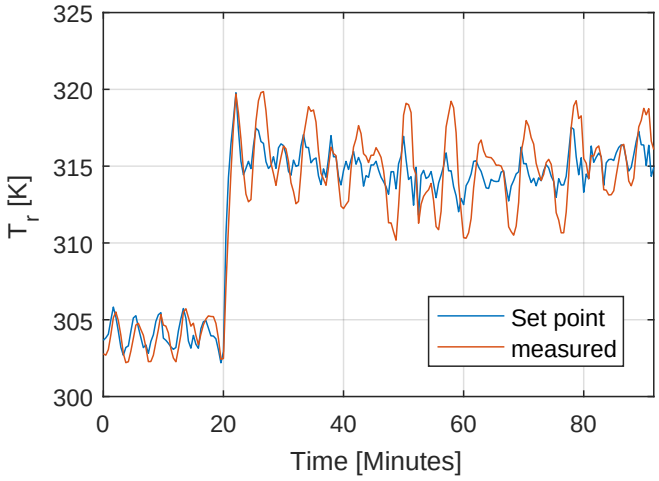

Fig. F.7: Control of the return temperature to match the output of the building model.

To emulate the cost of heat power, the cost structure of a typical district heating supply of Copenhagen "HOFOR" is used. Here the base cost of heat power is 72.28€/MWH and for each degree that ΔT is lower than 32^oC and added cost of 0.8% is added.

5 Conclusion

A hardware-in-the-loop experimental test setup was here introduced. The setup allows for faster testing and easier benchmarking. It is however important to remember that since a building model that is build on data from Nord 2 is used only characteristics captured in this model is tested. This setup has been used to benchmark different control methods. In the further development a natural next step would be testing a chosen solution on a real building HVAC system.

Paper G

Technical Report for simulation driven test

Anders Overgaard

This paper has not been published
2017

Abstract

This technical report outlines some of the aspects relating to the simulations of building HVAC which is used throughout this PhD project.

1 Introduction

Throughout the project simulation driven development of methods has been used. This means that methods has been tested in simulation with various levels of fidelity before being applied to the other test setups. Different forms of models has been used, first smaller principle physical models and later large, but also physical based numerical models. In this technical paper the simulation setup with the highest fidelity will be described.

2 Dymola

Dymola is a modelling tool that can be used for simulation in multiple domains. It is commercial, but uses the open platform Modelica modelling language, such that many open source libraries are available. The simulation tool uses both a visual interface and a code shell, such that a lot of the implementation of a model can be done visually, while the finer details can be coded directly in the modelica language. Behind the blocks the code is based on object oriented programming, making it hierarchal. To solve the differential equation numerically there are numerous solvers that can be used in Dymola. One of the solvers is DOPRI5, which is based on the explicit Runge-Kutta method where the problem [2].

$$\dot{y} = f(t, y), \quad y(t_0) = y_0. \tag{G.1}$$

is approximated by

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i k_i \quad t_{n+1} = t_n + h \tag{G.2}$$

where h is the step size and

$$\begin{aligned}
k_1 &= f(t_n, y_n), \\
k_2 &= f(t_n + c_2 h, y_n + h(a_{21} k_1)), \\
k_3 &= f(t_n + c_3 h, y_n + h(a_{31} k_1 + a_{32} k_2)), \\
&\vdots \\
k_s &= f(t_n + c_s h, y_n + h(a_{s1} k_1 + a_{s2} k_2 + \cdots + a_{s,s-1} k_{s-1})).
\end{aligned}$$

The coefficients follow the Butcher tableau

$$
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{ss-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}
$$

DOPRI5 uses variable step size, where the local error, previous step size and the user-defined tolerance are used to calculate the step size. The output interval can then be defined. This setup can handle very large models where the amounts of differential equations to be solved can be in the hundred of thousands.

3 Buildings Library

Models were created using components that was either developed during the project or from the "Buildings" library. The buildings library is open source and developed by Lawrence Berkeley National Labroatory [3]. It contains models to simulate buildings, such as rooms, HVAC, controls, weather, fluids and airflow. In Fig. G.1 an example of a model for a building zone build in dymola can be seen.

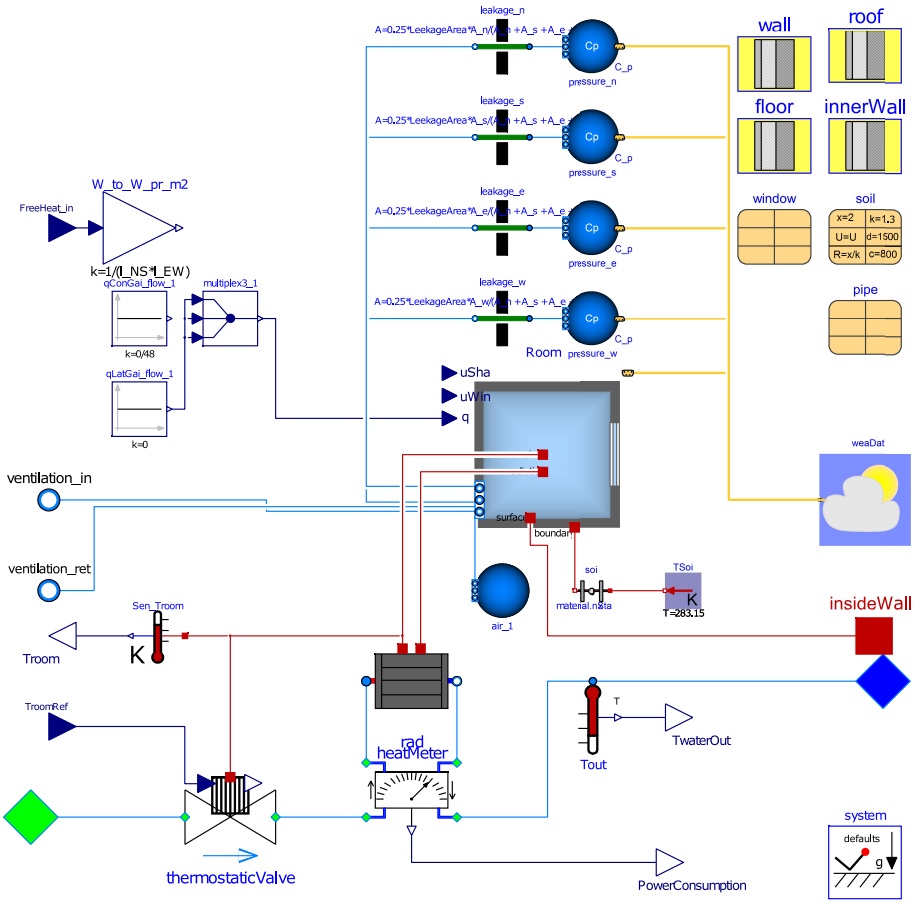

Fig. G.1: Dymola model example of a building zone.

In this example the zone is heated by a radiator, controlled by a thermostat. The building block in the middle holds the description of the structure such as dimensions, materials and windows etc. Many different facets of building models can be implemented in this setup. As an example windows, with different size, glazing and shade etc. Another example is leakage, which is implemented as depending on wind and pressure as can be seen in the four circles in the top. This zone can be connected to other zones such that heat will be conducted between zones. This zone component is a high level component build from multiple smaller components. The fundamental low level components that are often used is in the HeatTransfer package which contains models for conduction, convection and radiosity. An example of a low level component is the "SingleLayer" model where heat conduction for a single layer homogeneous material is computed. In between the ports of the

"SingleLayer" the material is divided into a user defined number of states. In the case of no phase change the SingleLayer numerically computes the heat equation

$$\rho c \frac{\partial T(x,t)}{\partial t} = \lambda \frac{\partial^2 T(x,t)}{\partial x^2} \tag{G.3}$$

where ρ is the mass density of the material, c is the specific heat capacity, T is the temperature, x is the distance into the material, t is time and λ is the heat conductivity.

4 Models

To test the control scheme in various systems multiple models was build. Two of the types apartment, house and mansion was build with the one following the Danish building standards from 1960 and the other following the standards of 2015. Fig. G.2 shows examples of floor plans from the 2015 house ($230m^2$) and apartment ($68m^2$).

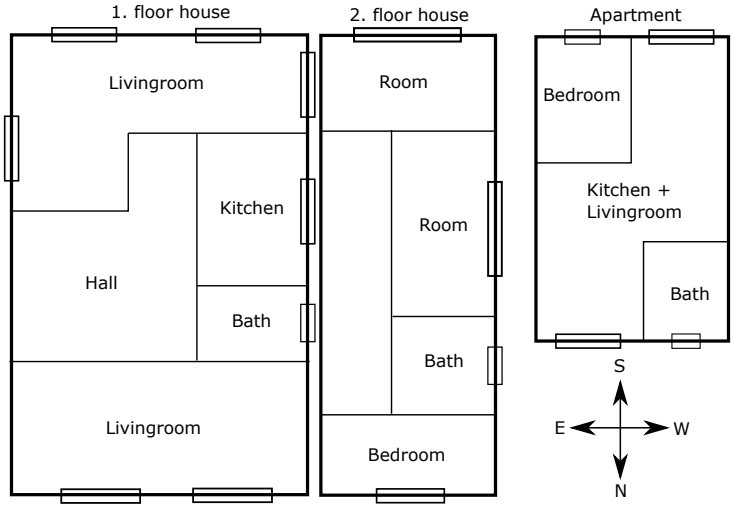

Fig. G.2: Floor plans house and apartment

The difference between 2015 and 1960 buildings is the standard building materials of the time and standards for insulation, where Danish buildings from 2015 has a higher degree of insulation. Danish building code is used from each of the periods.

For a simulation of a larger commercial building a model was build to resemble the School of Art, Design and Media at Nanyang Technological University Singapore. This building was used since good data from the HVAC

system and the building structure was available from earlier work and could be used in an early stage of the project.
The main components contained in the models are:

- Free heat from metabolism, electronics and hot water usage is modelled from typical daily, weekly and monthly patterns of usage.
- Weather data such as dry bulb temperature, wet bulb temperature, wind direction, wind speed, solar radiation, barometric pressure and cloud cover.
- HVAC system components for heating and cooling such as heat sources, distribution and terminal units.
- The building structure containing among other parts walls, floors, roof, furniture and the opening of doors and windows.

5 Controls

Control schemes for the models have been done in three ways during this work.

First method is by coding the control scheme in Modelica and used directly within the Dymola environment.

The second approach is using the interface between Dymola and Simulink. In this way a block as seen in Fig. G.3 containing the Dymola model can be used with input-output ports connecting the two environments. The solvers in Simulink is then used to compute the numerical solution for both the Dymola block and the rest of the Simulink code allowing for variable step solvers.

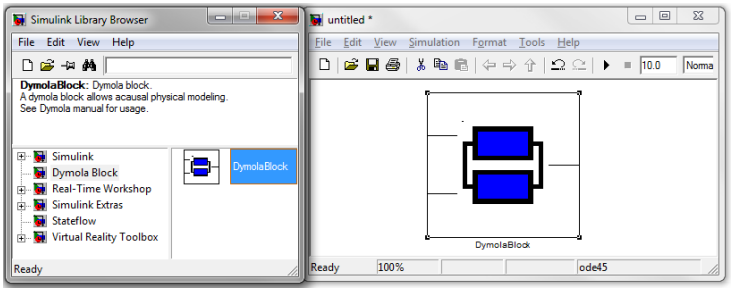

Fig. G.3: Dymola-Simulink interface [1]

The third approach is to generate a functional mock-up unit (FMU) from Dymola that follows the open standard functional mock-up interface (FMI). Since FMU is a standardised model structure many tools interfaces with

these. In this project the Python library PyFMI has been used to interface to python coded control schemes. When using the FMU approach two approaches can be used; model exchange or co-simulation. With model exchange the FMU is solved by an external solver (such as in the Dymola-Simulink interface). In co-simulation the solver is built into the FMU and separate solvers can be used by the two interfacing environments. Since the communication is done in discretized steps only constant step solvers can be used in co-simulation.

6 Conclusion

A simulation setup using Dymola and building models in the modelica language is used for multiple test throughout the project.

References

[1] Claytex, "Dymola." [Online]. Available: https://www.claytex.com/blog/dymola-simulink-interface/

[2] W. H. Press, S. a. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 2007.

[3] M. Wetter, W. Zuo, T. S. Nouidui, and X. Pang, "Modelica Buildings library," *Journal of Building Performance Simulation*, 2014.

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-544-4

AALBORG UNIVERSITY PRESS