Reinforcement Learning Based Control for Heating Ventilation and Air-conditioning

Blad, Christian

DOI (link to publication from Publisher):
10.54337/aau495047991

Publication date:
2022

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

Link to publication from Aalborg University

Citation for published version (APA):
Blad, C. (2022). *Reinforcement Learning Based Control for Heating Ventilation and Air-conditioning*. Aalborg Universitetsforlag. https://doi.org/10.54337/aau495047991

REINFORCEMENT LEARNING BASED CONTROL FOR HEATING VENTILATION AND AIR-CONDITIONING

BY
CHRISTIAN BLAD

DISSERTATION SUBMITTED 2022

AALBORG UNIVERSITY
DENMARK

Reinforcement Learning Based Control for Heating Ventilation and Air-conditioning

Ph.D. Dissertation

Christian Blad

Aalborg University
Department of of Materials and Production
Fibigerstræde 16
DK-9220 Aalborg

Dissertation submitted:	June 2022
PhD supervisor:	Associate Professor Simon Bøgh Aalborg University, Denmark
PhD committee:	Associate Professor Peter Nielsen (chair) Aalborg University, Denmark
	Associate Professor Roel Pieters Tampere University, Finland
	Lars Finn Sloth Larsen, Senior Manager R&D Danfoss A/S, Denmark
PhD Series:	Faculty of Engineering and Science, Aalborg University
Department:	Department of Materials and Production

ISSN (online): 2446-1636
ISBN (online): 978-87-7573-898-4

Published by:
Aalborg University Press
Kroghstræde 3
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Stibo Complete, 2022

This PhD thesis is typeset using LaTeX, and the AAU thesis preamble.

Abstract

A lack of resources, the increased focus on CO_2 emissions, and the resulting climate changes have resulted in an increase in the demand for energy saving technologies. This demand has driven this study of Reinforcement Learning (RL) based control for Heating Ventilation and Air-Conditioning (HVAC) systems.

Studies of the world energy consumption show that 40% of the world's energy consumption goes into HVAC systems. Studies also show that cost can be reduced by up to 20% of the current industrial practice by using optimal control on HVAC systems. However, to do optimal control on HVAC systems is difficult. Model Predictive Controllers (MPC) requires a unique model for a given building, which may or may not be economically feasible, hence the reason why it is not normal practice. On the other hand, there are model-free methods that typically are data-driven. Data-driven methods have previously been shown possible but data expensive.

This dissertation focused on data-driven methods for HVAC, and how to make these methods more robust and data efficient. Five papers are presented that all contribute to data efficiency and robustness. Of the five papers new methods are presented in the first four papers and the fifth paper is a field study verifying the methods in a real-world test. The five papers are outlined below:

- **Paper A**: This is a study of RL for Underfloor heating systems. In this paper Eligibility Traces is proven useful, also the issues with efficiency and robustness is identified and discussed.

- **Paper B**: In this paper a method for robust RL is developed. The method uses a traditional control policy on top of the RL controller, then soft constraints are implemented it such a manner that the algorithms learn whenever the traditional control policy takes over.

- **Paper C**: This study uses Multi Agent RL (MARL) to improve data efficiency. A 70% reduction in the convergence time is achieved when comparing Single Agent RL. Additionally, is it shown that a 17% price reduction can be archived with MARL when compared to industrial State-of-the-art controllers.

- **Paper D**: This Paper present a novel framework for Offline RL for HVAC system. This paper demonstrate that a near optimal policy can be found by only training offline.
- **Paper E**: This paper presents a field study of the algorithm developed in paper D. In this study the results of Paper A-D are verified.

The methods described above all support each other. This means that Paper D also includes the methods described in Paper A-C. The simulation studies presented in Paper D show that a near optimal policy can be found with 60 days of data, and the cost reduction is 19% when compared to a traditional control policy. Paper E is a field study, where a lab environment consists of two separate buildings of $9m^2$ and $16m^2$. These buildings are subjective to the outside weather but are not occupied by humans. During the winter 2020/2021 a benchmark test with a traditional control policy was done and during the winter 2021/2022 the Offline RL algorithm was deployed.

Dansk Resumé

Mangel på ressourcer, øget fokus på CO2-udledning og de deraf følgende klimaændringer har resulteret i en stigning i efterspørgslen på energibesparende teknologier. Denne efterspørgsel har drevet denne undersøgelse af Reinforcement Learning (RL) baseret kontrol til Heating Ventilation and Air-Conditioning (HVAC). Undersøgelser af verdens energiforbrug viser, at 40% af verdens energiforbrug bruges i HVAC-systemer. Undersøgelser viser også, at omkostningerne kan reduceres med op til 20% ved at bruge optimal kontrol hvis der sammenlignes med nutidens industrielle praksisser. Det er dog svært at lave optimal kontrol på HVAC-systemer. Model Predictive Control (MPC) kræver en unik model for en given bygning, at lave sådan en model er typisk ikke økonomisk rentabelt, hvorfor det ikke er normal praksis. I modsætning til MPC findes der også er der model frie metoder, disse er typisk datadrevne. Datadrevne metoder har tidligere vist sig muligt, men de kræver meget data hvilket betyder der går lang tid før disse virker. Denne afhandling fokuserede på datadrevne metoder til HVAC, og hvordan man laver disse metoder mere robuste og dataeffektive. Der præsenteres fem artikler i denne afhandling som alle bidrager til dataeffektivitet og robusthed. Af de fem artikler præsenteres nye metoder i de første fire artikler og den femte er et feltstudie, der verificerer metoderne i et virkeligt system. Formålet med de fem artikler er opsummeret nedenfor:

- **Artikel A**: Dette er en undersøgelse af RL for gulvvarmesystemer. I denne artikel er "Eligibility Traces" undersøgt og påvist nyttige i gulvvarme systemer, også problemerne med data effektivitet og robusthed er påvist og diskuteret.

- **Artikel B**: I dette papir er der udviklet en metode til robust RL. Metoden bruger en traditionel kontrol politik som en sekundær kontroller til RL kontrolleren, for at optimere læringen implementeres "soft constraints" på en sådan måde, at algoritmerne lærer, når den traditionelle kontrolpolitik tager over.

- **Artikel C**: Denne undersøgelse bruger Multi Agent RL (MARL) til at forbedre dataeffektiviteten. En 70% reduktion i konvergenstiden opnås ved sammenligning af Single Agent RL. Det er desuden vist, at en prisreduktion på 17% kan opnås med MARL sammenlignet med industrielle standard kontroller.

- **Artikel D**: Denne artikel præsenterer en ny metode for Offline RL til HVAC-system. Denne artikel viser, at en næsten optimal kontrol politik kan findes kun ved brug af offline træning.
- **Artikel E**: Denne artikel præsenterer et feltstudie af algoritmen udviklet i Artikel D. I denne undersøgelse er resultaterne af Paper A-D verificeret.

De ovenfor beskrevne metoder understøtter hinanden. Det betyder, at artikel D også omfatter metoderne beskrevet i artiklerne A-C. Simuleringsundersøgelserne præsenteret i artikel D viser, at en næsten optimal kontrol politik kan findes med 60 dages data, og omkostningsreduktionen er 19% sammenlignet med en traditionel kontrol politik. Artikel E er et feltstudie, hvor et laboratoriemiljø bestående af to separate bygninger på $9m^2$ og $16m^2$. Disse bygninger er udsat for vejret udenfor, men er ikke af menneskelig adfærd. I løbet af vinteren 2020/2021 blev der udført en benchmark test med en traditionel kontrolpolitik, og i vinteren 2021/2022 blev Offline RL algoritmen implementeret og testet.

Contents

Thesis Details

Thesis Title:	A simple AAU Template for a Collection of Papers Ph.D. Thesis
Ph.D. Student:	Christian Blad
Supervisors:	Prof. Simon Bøgh, Aalborg University Assoc. Prof. Carsten Skovmose Kallesøe, Aalborg University/Grundfos A/S

The main body of this thesis consist of the following papers.

- **C. Blad**, S. Koch, S. Ganeswarathas, C.S. Kallesøe, S. Bøgh, "Control of HVAC-systems with Slow Thermodynamic Using Reinforcement Learning," *Procedia Manufacturing,*, vol. 38, pp. 1308–1315, 2019.

- **C. Blad**, C. S. Kallesøe, S. Bøgh, "Control of HVAC-Systems Using Reinforcement Learning With Hysteresis and Tolerance Control," *IEEE/SICE International Symposium on System Integration*, pp. 938–942, 2020.

- **C. Blad**, C. S. Kallesøe, S. Bøgh. A Multi-Agent Reinforcement Learning Approach to Price and Comfort Optimization in HVAC-Systems. Energies. 2021; 14(22):7491. https://doi.org/10.3390/en14227491

- **C. Blad**, C. S. Kallesøe, S. Bøgh, Data-Driven Offline Reinforcement Learning for HVAC-Systems. Currently under second round of review in Energy 2022

- **C. Blad**, C. S. Kallesøe, S. Bøgh, A Field Study of Offline Multi Agent Reinforcement Learning. To be submitted to Energy and Buildings, Elsevier June 2022.

This thesis has been submitted for assessment in partial fulfillment of the PhD degree. The thesis is based on the submitted or published scientific papers which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

Preface

This PhD thesis is written as a collection of papers. The work has been supported by the Innovation Fund Denmark and Grundfos A/S. Specifically, has the work been done in collaboration with the robotic and automation group at Aalborg University and the Technology and Innovation Control Department at Grundfos. The work has been conducted over 3 years and 3 months, from 1st January 2019 to 1st of April 2022.

Acknowledgements

First and foremost, I would like to acknowledge my two supervisors Assoc. Prof. Simon Bøgh and Prof. Carsten Skovmose Kallesøe. It has been a true pleasure to have worked with you, and the environment you have created for me to explore new ideas in, have been amazing. I would also like to thank Grundfos and Hakon Bøsrting for giving me this chance to study RL based control for HVAC, it has been a fantastic opportunity.

A big thank you to my close collage's both at Aalborg University and Grundfos for always entertaining my ideas, maybe also when they were a bit far stretched, and of cause for making these 3 years more fun. A special thank you to Erik B. Sørensen, beside from being good company, the support you have given in modelica/dymola has been vital to this project.

Finally, a big thank you to my friends and family for supporting me throughout this project. I have heard it can be lonesome to do a PhD study, but with family and friends like mine, I haven't experienced it.

Reader's Guide

This PhD thesis is written as a collection of papers. It contains two parts: Part I) are chapters 1-5 where the work done during this thesis is summarized and reflected on and Part II) is the collection of papers. References follow the numeric format, like [x]. A given reference can be found in the list of references on pages 61-65.

The document is written in LaTeXusing Arial and compiled using Overleaf. The source file is available at [1] this template complies with AAU guidelines for an Article Based PhD Thesis. All figures are made in Inkscape or directly exported from python using matplotlib. The papers included as Paper A-E are imported in their original format, hence not cropped or otherwise changed to fit this thesis.

Christian Blad
Aalborg University, June 20, 2022

[1] https://github.com/jkjaer/aauLatexTemplates

Part I

Chapter 1

Introduction

1 Introduction and Motivation

Over the past decade, energy efficiency has become even more important due to the growing concern of climate change, exploitation of natural resources, and a growing world population. This is also clear in the 17 UN world goals, where three speaks directly into this [37]. This Ph.D. project has explored how to utilize state-of-the-art methods in Reinforcement Learning to do model-free optimal control of Heating Ventilation and Air-conditioning (HVAC) systems. The type of HVAC system the algorithms have been tested on, is Underfloor Heating (UFH), in a experimental setup that resembles a domestic house.

The building sectors in the US and China are consuming 40% and 43% of the annual energy production respectively [22, 58]. For comparison the transport sector consumes 28% of the annual energy production, and within the transportation sector is aviation consuming about 11%. Hence even incremental improvements to the HVAC sector, will result in large reductions in the energy consumption. Figure 1.1 shows the energy consumption by sector and their subcategories. Roughly 50% of the energy consumed in the commercial and residential building sector is HVAC energy.

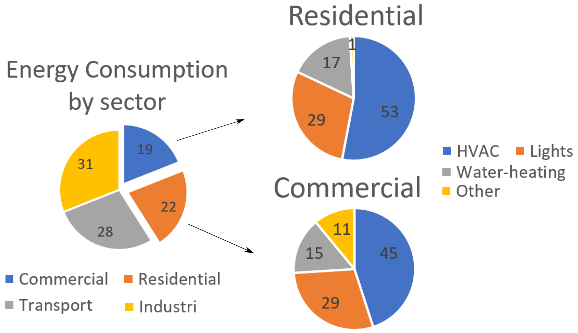


Fig. 1.1: Energy consumption across sectors in the US is seen to the left. The relevant sectors "Commercial" and "Residential" are seen divided into sub categories to the right [46].

Studies show that there is a large potential to reduce the consumption of HVAC energy. This can be done not only by building new buildings or re-insulating existing buildings but also by recommissioning the HVAC system of the existing building mass. In [35] 150 exciting commercial buildings are investigated. It is found that recommissioning the HVAC system can save 15% of the energy consumption on average and that the average return on investment is 0.7 years. This translates to US $18 billion in annual savings in the commercial building sector in the US alone.

Another method to additional reduce the energy consumption of buildings is predictive control. A common method to do predictive control of buildings are model predictive control (MPC) [5, 25, 42]. In these papers is it found that MPC controllers could reduce the energy consumption by 17-24%, whether or not these 17-24% can be directly added to the savings of 15% found in [35] cannot be concluded, but it can be assumed that savings achieved with MPC controllers are additional savings since it was tested on already commissioned buildings. Other smart controllers that, similar to MPC, do predictive control are [49, 54]. These controllers use energy price forecast and a thermodynamic model to minimize energy consumption. Similar savings were achieved with these controllers. All these controllers do however have one large drawback, an extensive thermodynamic model is required, and this gives cause for concern. An extensive model results in even greater commission cost, and continuous upkeep of the model.

An alternatively to model-based predictive control is model-free optimal control methods. Model-free methods naturally require some kind of learning, this Ph.D. project has solely focused on RL, however one could explore other types of learning controllers such as Iterative Learning Control (ILC) that before has been used in control of HVAC systems [53]. As the name suggests these methods are model-free and hence do not need a commission of a model [48]. This means that the issue of badly commissioned HVAC systems as described in [35] will to some extent be solved because the HVAC system will automatically and continuously be commissioned over the entire life span. Furthermore, an RL controller will also perform optimal control like an MPC but without the commission and maintenance of a model.

Model-free RL can overall be done in three ways 1) online, 2) online off-policy, and 3) offline. An illustration of Online RL and Offline RL can be seen in Figure 1.2.

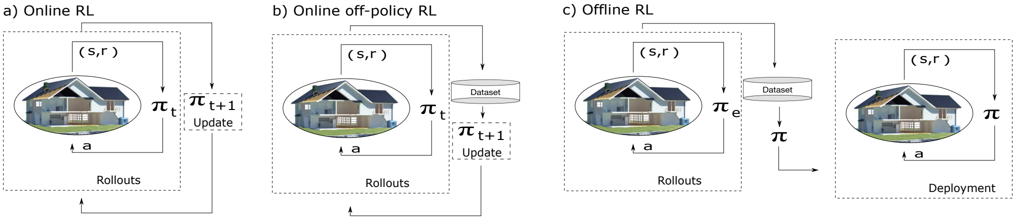


Fig. 1.2: In (a) is a online RL algorithm shown. The policy is updated for every iteration with the latest experience(s,a,s',r). In (b) is the online off-policy shown, here the policy is updated not only with the latest experience but with all experiences. In (c) offline RL is visualized. In Offline RL data is gathered with policy π^e a policy π is then derived from the data and this policy is then deployed.

Numerous papers have been written on online RL and online off-policy RL for HVAC systems. In [11, 40] online RL is used to control the thermostat in a thermal zone and the mixing loop in a commercial building respectively. It is shown that RL can be used with success to control parts of a Heating system. In [51] online off-policy RL used to control the thermostats in multiple zones. This Ph.D. project has only explored off-policy RL in both online and offline settings. To the author's knowledge is there no papers on RL for Underfloor heating as heating installation. Therefor is this the first topic this project is concerned with. Common for all RL algorithms is that one cannot ensure robust/safe behavior because of the black-box nature of an RL algorithm is [21]. This is especially true, during early training and exploration of the action-state space. Because this project aims to develop a commercially viable algorithm, is robustness and safe learning one of the main topics of this study and a solution to safe RL for HVAC has been proposed.

RL suffers from what Richard Bellman referred to as "the curse of dimensionality" [12]. This means that as the number of actions and states grow so does the complexity of the action-state space and the value function that a value-based RL algorithm is associated with. To combat the growing complexity, artificial neural networks as function approximators have shown great promises [36]. Additionally, to using schemes to generalize over the action-value function the environment can also be re-formulated as a Markov Game (MG). By formulating the environment as a MG instead of a Markov Decision Process (MDP), Multi-Agent RL(MARL) can be used. MARL can with some assumptions reduce the complexity of the action-state space. This is especially interesting for an RL algorithm that is supposed to control the entire HVAC system or an entire underfloor heating system.

To illustrate the scaling problem of having a single agent to control the entire system of a one, two, three, and four-zone UFH system, the calculations of the action-state space have been made for the four cases, see Table 1.1.

Table 1.1: Size of action-state space for one, two, three and four-zone UFH system. The action-state space grows exponentially with number of temperature zones. Assumptions about the number of discrete values each action or state has: T_{supply}; 15, Valve per zone; 2, T_{room} per zone; 12, T_{amb}; 30, heat consumption; 10, sun; 10.

	Action-state space size
One zone	$1 \cdot 10^6$
Two zones	$26 \cdot 10^6$
Three zones	$622 \cdot 10^6$
Four zones	$15 \cdot 10^9$

The above table shows that the action-state space grows exponentially with the number of temperature zones in the system, this means that RL control does not scale well in the UFH control task or many other control problems [48]. In this study is it therefore proposed to use Multi-Agent Reinforcement Learning MARL to overcome the scaling problem.

Robustness and fast convergence are in any type of online RL the key parameters when designing for HVAC systems. If, however offline RL can be used, these parameters only a concern when the algorithm is deployed. By using offline R,L existing building data can be used. This is of interest in a scenario where an existing building is recommissioned, and operation data is still available. Also, in a new commission of a building offline RL can be feasible. This is because existing traditional controllers can run for a period, this data can then be used to build a model in which the RL algorithm can train.

With the introduction, motivation, and potential for using RL in HVAC systems described, can the research questions this Ph.D. project has been concerned with, be outlined.

2 Research Questions

The research questions have been formulated on the basis of the information shared in Section 1.

The four research questions this Ph.D. thesis addresses are outlined below:

RQ1 How does model-free Reinforcement Learning perform in an Underfloor Heating System?

RQ2 How can a framework for robustness in Reinforcement Learning for HVAC systems be designed?

RQ3 Can Multi-Agent Reinforcement Learning be used to reduce convergence time for RL HVAC control?

RQ4 Can Offline Reinforcement Learning be a data efficient method for HVAC control?

In the following is outlined which papers contribute to the respective research questions:

- **RQ1** is answered in Papers **A** and **F**
- **RQ2** is answered in Papers **B** and **F**
- **RQ3** is answered in Papers **C** and **F**
- **RQ4** is answered in Papers **D** and **F**

3 Research Methodology

To answer the research questions formulated in section 2 the methodology used in this thesis is presented. The methodology will give a structure to the thesis, this will ensure that the questions are answered in a proper manner and that there is consistency trough out the thesis. The research methodology presented in this thesis is inspired by the Soft Systems Methodology (SSM) [19]. The SSM is a methodology structure which begins with a "soft problem" like "how can RL function in HVAC systems". Following this is the problem identified and solved, and lastly is the solution tested.

To illustrate the methodology used in this thesis Figure 1.3 is made.

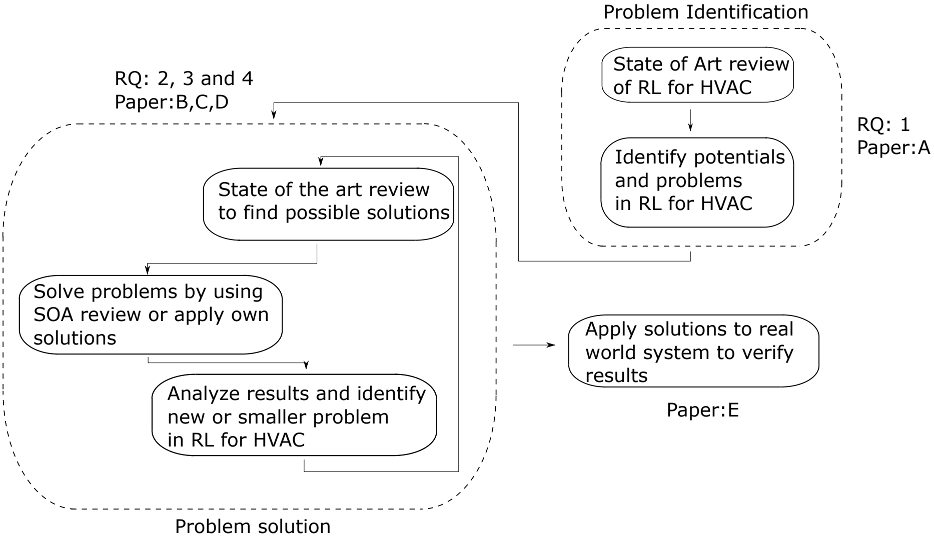

Problem Identification
State of Art review of RL for HVAC
Identify potentials and problems in RL for HVAC
RQ: 1
Paper:A
RQ: 2, 3 and 4
Paper:B,C,D
State of the art review to find possible solutions
Solve problems by using SOA review or apply own solutions
Analyze results and identify new or smaller problem in RL for HVAC
Problem solution
Apply solutions to real world system to verify results
Paper:E

Fig. 1.3: This Figure is a illustration of the methodology used in this thesis. From the Figure can it be seen in which steps the respective research questions are answered and how each paper are contributing to this.

From Figure 1.3 it can be seen that the methodology is split up into 3 stages. Stage 1 is the problem identification stage. This is the beginning of the thesis where RQ 1, has the purpose of identifying potentials and weaknesses in RL for HVAC. To identify potentials and weaknesses in RL for HVAC RQ1 has been answered, and Paper A has been written. Paper A is an SOA RL algorithm designed and deployed in a simulation environment.

Stage 2 is to solve the issues found in Paper A. It was found that robustness was the most immediate issue to combat, hence RQ2. With RQ2 answered the problem of fast convergence became relevant, hence RQ3 and finally RQ4.

Stage 3 is the laboratory and validation work. In this stage is new research not produced, however is an application paper (Paper E) made based on the results obtained in the real-world test environment.

4 Structure of Thesis

This thesis is structured in two parts; in part I, a summery of the work done during this Ph.D. Project is presented. In Chapter 1 the introduction, motivation, and research questions have been outlined. Following this Chapter 2 contains a state-of-the-art review of related work within the field of RL and HVAC. In Chapter 3 the problem is described and the overall problems in the control of HVAC systems are outlined. Furthermore, the three test environments of which the research is based on are explained. In Chapter 4 the solutions are presented, in this chapter the papers A, B, C, D and E are explained. Lastly, the is work concluded on in Chapter 5. Part II is a collection of papers that supports the work done during this Ph.D. project.

5 Publication and Submissions during PhD Project

This section presents the paper published and submitted during this Ph.D. project, the Papers in present in dated order, with the earliest paper first.

Paper	Reference
A	C. Blad, S. Koch, S. Ganeswarathas, C.S. Kallesøe, S. Bøgh, **Control of HVAC-systems with Slow Thermodynamic Using Reinforcement Learning**, Procedia Manufacturing, Volume 38, 2019, Pages 1308-1315, ISSN 2351-9789.
B	C. Blad, C. S. Kallesøe and S. Bøgh, **Control of HVAC-Systems Using Reinforcement Learning With Hysteresis and Tolerance Control**, 2020 IEEE/SICE International Symposium on System Integration (SII), 2020, pp. 938-942, doi: 10.1109/SII46433.2020.9026189.
C	C. Blad, C. S. Kallesøe and S. Bøgh, **A Multi-Agent Reinforcement Learning Approach to Price and Comfort Optimization in HVAC-Systems**, Energies 2021, 14, 7491. https://doi.org/10.3390/en14227491
D	C. Blad, C. S. Kallesøe and S. Bøgh, **Data-Driven Offline Reinforcement Learning for HVAC-Systems**, Energy, under second round of review
E	C. Blad, C. S. Kallesøe and S. Bøgh, **A Field study of Offline Multi-Agent Reinforcement Learning**, to be submitted to Energy and Buildings, Elsevier in June 2022

Below a short description of each paper along with a description of how the paper contributes to the state of the art within the field of RL and HVAC is given.

- Paper A
  - This paper is concerned with RQ1 and can best be described as a survey study of RL for UFH. In This paper is a Simulink model for UFH developed and a Deep Q-network (DQN) RL algorithm is tested on this model. these results are compared with a DQN algorithm with eligibility traces. It was found that Eligibility Traces do improve the performance of RL in HVAC.

  Furthermore, it was found that robustness and scalability are of concern when deploying RL in HVAC.
  - Moreover this paper contributes with testing RL on UFH and showing that robustness and scalability are problems when using RL in UFH systems. This paper also shows that eligibility traces improve performance in RL for UFH.

- Paper B
  - This paper is concerned with RQ2, where improving robustness is central. Firstly, a new Dymola model isd developed for simulating UFH systems, with a dedicated air to water heat pump as the heating source. This paper investigates how to implement robustness in RL without compromising the optimal policy and increasing the convergence time for this policy.
  - Moreover this paper contributes with a novel framework for robust RL control of HVAC systems. It is found that implementing a traditional controller on top of the RL algorithm and incorporating linear soft constraints in the reward function robustness can be ensured. Additionally, the convergence time can be reduced because the action/state space is reduced to what is known to be feasible.

- Paper C
  - This paper is concerned with RQ3, where improving scalability is central. In this paper the Dymola models developed in Paper B are used. This research explores MARL as a solution to the scaling problem. A novel approach where the communication of actions between agents is set up in a manner so it from a system point of view is sensible. Furthermore, the state space is divided such that each agent only receives states relevant for its own objective. Lastly, the MARL algorithm is compared to a SARL algorithm.
  - Moreover this paper contributes with a novel MARL algorithm for control of HVAC systems. This Algorithm reduced the convergence time for a 4 zone UFH system from 600 days when using SARL to 180 days when using MARL, this translates to a 70% reduction in the convergence time.

- Paper D
  - Paper D Investigate RQ4 regarding Offline RL for HVAC systems. This paper is a novel framework for Offline RL developed. The Framework is based on Black box modeling methods and is therefore completely commission-free. This paper investigate how much data is required to do Offline RL, and how this data can be gathered.

  - This paper contributes to the field of offline RL for HVAC systems by developing a novel framework for offline RL. It is shown that the framework can be applied to a four temperature zone UFH system and perform well with 60 days of data. Additionally, it shows that the framework can be used in recommissioning tasks and a near-optimal policy can be found from the first day of deployment. This naturally requires that data from the proviso system is available.

- Paper E
  - This paper is an application paper whose purpose is to verify the research done in Paper A-D. In this paper is the algorithm developed in Paper D deployed in a real world environment. Real data is gathered as a bench marking test during the spring of 2021 and the winter 2021/2022. This data is then used in the offline RL framework presented in Paper D. The offline trained algorithm is deployed mid January 2022 and is tested until 10th of Marts.
  - In this paper is it found that the Offline RL framework presented in Paper D can be deployed in a real world environment. An analysis of the test data shows that the control policy exhibits predictive control like behavior. Oscillation of the system is reduced by a minimum of 40% while the cost of heating is reduced by minimum 10 %.

Chapter 2

Literature Study

This chapter summarizes the literature studies done to make the research presented in the five papers. This part of the thesis is divided into four sections. Each section describes the background theory for the respective topic followed by the literature study of the given field.

1 Energy Consumption of HVAC Systems

As described in Chapter 1, the consumption of HVAC is one of the largest single sectors of energy consumers [44]. If dividing HVAC energy into heating and cooling energy sources it will naturally be different depending on the geographical location. When looking at the Western World (Europe and USA) heating in the residential sector in the US and Europe is accounting for 88% and 92% respectively [3, 38]. Hence, heating is by far the most energy requiring, for this reason is heating the prime focus of this thesis. One may argue that it is the same dynamic properties and therefore the same algorithms will also work for cooling. When looking at the energy sources natural gas is by far the most common method for heating, in Figure 2.1 the distribution of energy sources in the EU can be seen.

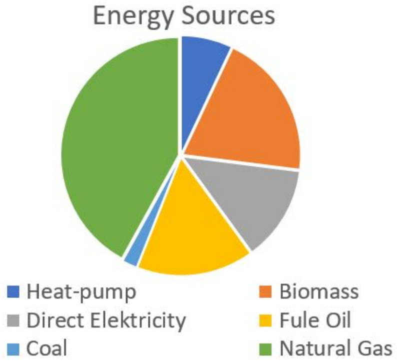


Fig. 2.1: Energy sources in EU [3].

Heating with electricity and heat pumps are, however, becoming more popular in both the EU and US where the heat pump market is growing 8.5% and 8% respectively [8, 9]. According to the EU commission this is also a part of the long-term strategy to convert more HVAC energy to electrical sources [3]. An obvious benefit of using heat pumps is that renewable energy can be used to heat the houses, but also the fact that production of renewable energy is varying not as a function of consumption but as a function of the season and weather. This variation in production will only increase as more energy comes renewable energy sources. For the consumer this variation is reflected in the cost, so energy is cheaper when the production is high and vice versa.

Because the HVAC energy sector for heat pumps is increasing and an interesting optimization problem arises when looking at building dynamics, the dynamics of a heat pump, user occupancy, and the price of electricity this Ph.D. thesis has focused on UFH systems with air to water heat-pumps as heating source.

2 Reinforcement Learning

The theory and results in the following section are based on excerpts from Paper C - "A Multi-Agent Reinforcement Learning Approach to Price and Comfort Optimization in HVAC-Systems" published in the Open Access journal Energies. Reinforcement Learning is as described in Chapter 1 an iterative learning method. Furthermore, it is described how this thesis focuses on model-free RL and how this can be divided into 1) online RL, 2) online off-policy RL, and 3) offline RL [48]. Additionally, model-free RL can be divided into three categories; value-based learning, policy-based learning, and actor-critic-based learning, where actor-critic is a combination of value-based and policy-based learning. Because this thesis focuses on online off-policy RL and offline RL value-based learning is central. This thesis has more specifically employed Q-learning as a backbone in all algorithms developed in Papers A-E.

Q-learning is based on the Bellman equation seen in Eq. (2.1).

$$Q^{*}(s,a) = \mathbb{E}[R_{t+1} + \gamma \max_{a^{'}} Q^{*}(s^{'},a^{'})] \tag{2.1}$$

The bellman equation simply states that if the further state's for all actions is known, the optimal policy is to choose the action that results in the highest Q value (Q*(s,a)). This is also what is referred to as a greedy policy [12].

Choosing the correct action is the easy part, updating the Q-function, so that Q_i converges towards Q_* is however difficult, or at least to do so within a reasonable number of iterations. In Eq. (2.2) the update strategy for Q-learning without function approximators can be seen. For the update strategy to converge to Q^* it is necessary for the environment to satisfy the conditions of a Markov Decision Process (MDP).

Definition:A MDP is defined by a tuple {S,A,P,R}; S is the finite number of states, A is a finite number of actions, P is the transition probability for s_t to transition to s_{t+1} under a given action a. R is the immediate reward for the expected transition from s_t to s_{t+1}. [43]

A MDP can only be defined as a MDP if s_{t+1} is dependent of s_t and not s_{t-n}. If the state space is not fully described, which often is the case in real world applications, the environment is defined as a partially observed MDP(POMDP). The same theory applies to a (POMDP). However, is the transition probability matrix naturally is associated with some degree of uncertainty.

$$Q^{update}(s,a) = Q^{current}(s,a) + \alpha \cdot (r_t + \gamma \max_{a^{'}} Q^{*}(s^{'},a) - Q^{current}(s,a)) \tag{2.2}$$

from Eq. (2.2) can it be derived that for the Q-function to converge to Q^* is it necessary to visit all state action pars. This is simply not feasible in large systems, therefore is function approximation used to approximate the Q function $Q(s,a;\theta) \approx Q(s,a)$. By using an artificial neural network (ANN) as function approximation has it been shown that a Q-function can converge [36]. Additionally, to function approximation with ANN's has double Q-learning [50], experience replay, prioritized experience replay [45] and a range of other methods been developed for reducing convergence time or improving convergence in value-based RL.

Many of the above mention methods have been applied in various forms in the papers presented in Chapter 1, this work is also elaborated on in Chapter 4.

As described in Chapter 1 RL in MDPs for HVAC systems has been studied In [11, 39]. In [51], RL is used to control airflow rates for up to five zones, where each zone has an individual actuator. It is found that it is possible to reduce energy cost, but training times increased drastically when going from one to four zones. This result supports the need for an Markov Game formulation, which makes it possible to use Multi Agent RL in an HVAC system. RL is from hereon referred to as *Single Agent RL* (SARL) and *Multi Agent RL* (MARL).

3 Multi Agent Reinforcement Learning

The theory and results in the following section are based on excerpts from Paper C - "A Multi-Agent Reinforcement Learning Approach to Price and Comfort Optimization in HVAC-Systems" published in the Open Access journal Energies. MARL is like SARL concerned with solving decision-making problems. However, instead of one agent deciding all actions in a system and receiving one reward, multiple agents decide the actions and receive individual rewards, or a joint reward based on the state of the system.

This thesis focuses on MARL systems formulated as an MG. The formal definition of an MG is as follows:

Definition: A Markov Game is defined as a discrete time-stochastic process, a tuple $\langle N, S, {A^i}_{i \epsilon N}, {R^i}_{i \epsilon N}, P \rangle$ where N is the number of agents, S is the state space observed by all agents, ${A^i}_{i \epsilon N}$ is the joint action space of all N agents. $R^i : S \times A \times S \to \mathbb{R}$ is the immediate reward received by agent i for transition from (s, A^i) to s' and P is the transition probability [18]. The definition of an MG can be interpreted as the following: at time T every i'th agent $i = [1..N]$ determines an action A^i according to the current state s. The system changes to state s' with probability P and each agent receives an individual reward R^i based on the new state s'. The goal for each agent is to maximize its long-term reward by finding the policy π^{i*}. [18]. The value function update for an MG is defined as follows:

$$V^i_{\pi^{i*}, \pi^{-i*}}(s) = \mathbb{E}[\sum_{t \geq 0} \gamma^t R^i(s_t, a_t, s_{t+1]}) \mid a^i_t \sim \pi^i(\cdot \mid s_t), s_0 = s] \tag{2.3}$$

When going from SARL to MARL an entirely new dimension is added to the problem. It is therefore necessary to define what type of MG the system is. The type describes how the agents are formulated and affect each other. A MARL problem can be formulated in three ways; 1) a Cooperative setting 2) a Competitive setting, and 3) a Mixed setting [56].

Cooperative, Competitive and Mixed Setting

In general a MARL algorithm in a cooperative setting is formulated with a common reward function so $R^1(s, a, s') = R^2(s, a_2, s') = R^n(s, a_n, s')$ and referred to as a Markov team game. A number of different algorithms exist for solving a Markov team game, these include *team-Q* [33] and *Distributed-Q* [30]. Distributed-Q is a MARL algorithm framework, which is proven to work for deterministic environments. It has been shown that all agents will converge to an optimal policy without sharing data between agents. This is very compelling, but because this paper works with a general-sum problem, this will not work.

A MARL algorithm in a competitive setting is formulated as a zero-sum game where one agent's win is the other agent's loss $\sum_i R^i(s, a_i, s') = 0$. There is for zero-sum games made a number of algorithms. An example of these is the *minimax-Q* [32]. Because an HVAC cannot be categorized as a zero-sum game, these algorithms of are little interest.

Lastly, the mixed setting, also referred to as a general-sum game, where the system is categorized by each agent having its own reward function, and therefore its own objective. In these types of systems, game theory and the Nash equilibrium play an essential role. In contrast to a cooperative setting where it is possible to assume that the system's overall best reward can be found by having all agents maximize their own reward, it is not possible in a general-sum setting. General-sum games have been designed for static tasks [17], this is of little interest unless adapted to dynamic tasks. Also, Single-agent algorithms are used in a mixed setting [20, 34], even though there is no grantee of conversion if applying SARL in a multi-agent system [18]. Algorithms which are designed for dynamic tasks, include *"Nash-Q"* [24], *"PD-WoLF"* [10] are of interest. The idea of a Q-learning algorithm that finds a Nash equilibrium is compelling. Succeeding papers have however argued that the application of Nash-Q is limited to environments that have a unique Nash equilibrium [56]. More recent work on fully decentralized MARL has been proven to converge under an assumption of using linear function approximators for the value function [57]. Even though this algorithm is distributed, a joint Q function is incorporated, which makes all agents aware of each other. This is necessary to prove general convergence, but it also increases complexity as the number of agents grows and hence makes it less scalable. In more recent work agents are distributed in a similar manner to our work [13]. The main difference between their architecture and our framework is that our agents only observe parts of the state space, as well as utilizing different methods to update the Q-function.

Additional to SARL and MARL has this thesis work with Offline RL. The goal of working with Offline RL is to use existing building data to build black box models on which the RL algorithm can train offline.

4 Offline Reinforcement Learning

Offline training of a RL algorithm requires a model of the real environment or data. When training from data there are multiple issues. The obvious issue is sparse data. If the high reward areas of the state-action space is not included in the data set, the value-function derived from the data naturally will not include these areas as well. Less obvious is how the data distribution and the shift in data distribution effects offline RL. In supervised learning, which effectually the problem are becoming when doing offline training directly from data, is the goal to predict some state S_{t+1} from S under the same data distribution. In RL the goal is to change the policy and hence do something different, presumably better, which easily can change the data distribution [31].

Training offline directly from data with Q-learning can be done by initializing the algorithm and load the data, consisting of the state action and reward transitions (s,a,r) into the replay buffer and allow the algorithm to approximate the value function [31]. This type of offline RL has been applied in [27]. In that paper, the task was for a robot to grap items on a table by using images.

In the work presented in this thesis offline training is done on a model, however not a pre-built and verified model, but a data-driven model. The argument for doing this and not loading the data into the Replay buffer is that it will be possible to generate synthetics experience by applying disturbances that are not represented in the collected data. This will combat the issue of shifting data distribution. The model is however naturally associated with some uncertainty for this reason all disturbances are not applied but only disturbances which to a large extent are represented in the collected data.

Black-box Model Generation for HVAC Systems

Model generation overall be split into 3 categories 1) Physics-based methods also referred to as white-box 2) black-box(data-driven) methods or 3) a combination of the two called grey-box methods. In grey-box methods is an overall structure defined by physics and data is then used to fit the parameters of the model [4]. A Physics-based model requires extensive modeling work, and because the dynamics of two houses are never the same, this work is more or less required for every building it is installed in, this not feasible. A grey-box method can be variable, it is however not commission-free, it does require expert knowledge to identify which type of installation it is used in [7]. Because this paper strives to develop a model-free approach a data-driven model is developed. Even though this paper uses a black-box model, it can be argued that a grey-box model will be more data-efficient and better at generalizing under a sparse data foundation, but for the reason stated above a grey-box model is not used.

There are several different methods to build data-driven models for HVAC systems. the work presented in this thesis uses ANN as function approximators, these have before has been deployed in black-box models for HVAC systems with success [6, 29]. Due to the slow and delayed responses, a radiant heating system is associated with it is beneficial to apply a recurrent neural network(RNN). An RNN is a broad term for neural networks that can identify patterns in a sequence of data [55], in an HVAC context is this time-sequence data. An Long Short Term Memory (LSTM) layer is a type of RNN that can identify patterns over shorter or longer periods of time depending on the problem [23]. LSTM networks have also been used in a black-box model context to predict load profiles of electricity consumption [47], one can argue that some of the same dynamic properties at least when talking about user behavior are present in electricity consumption as in HVAC systems.

Chapter 3

RL-Based Control of HVAC-systems

This chapter addresses the fundamental problems in control of HVAC systems, and the problems associated with developing control algorithms for these types of systems. This chapter will also present the simulation environments used for papers A-D and the experimental setup used in Paper E.

1 Problem Description

This problem description is an extension of the research questions presented in Chapter 1. The purpose is to outline the problems associated with each research question, and to present the background work that has made it possible to answer these questions.

The problem description can be summed up to the following question:

Can an RL-based control algorithm for UFH systems be developed with the same safe/robust behavior as a traditional event based controller without compromising the RL-based controller's ability to find the optimal control policy?

For the reader to gain an insight into the specific problems associated with control of UFH systems or building temperature, in general, the following section describes the thermodynamic properties of buildings and how these affect the control performance.

2 HVAC and Building Dynamics

To elaborate on the problems associated with control of HVAC systems an illustration of an HVAC system for a two zone house is made, see Figure 3.1

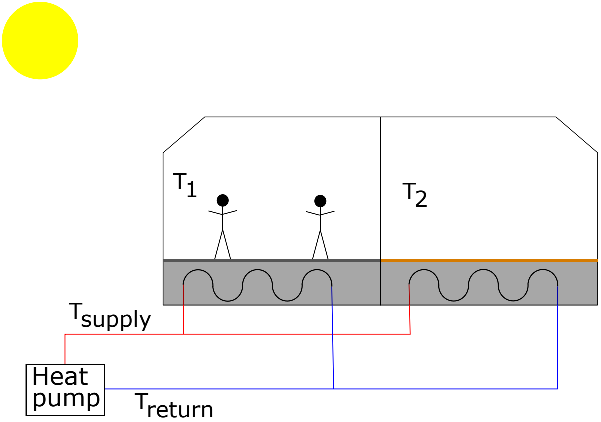


Fig. 3.1: An illustration of a two zone HVAC system, with a air-to-water heat pump. T_1 and T_2 refer to the temperature in zone 1 and 2, and T_{supply} and T_{return} refer to the supply temperature from the heat pump and the return temperature from the temperature zones.

For the purpose of explaining the dynamics of a system like the one in Figure 3.1 is the system divided into the HVAC-system dynamic and the building dynamics.

2.1 HVAC dynamics

The HVAC system in Figure 3.1 consists of a heat pump, two on/off valves, and x meters of pipe. The control actions for a system like this are;

- Opening or closing valve 1.
- Opening or closing valve 2.
- Adjusting the supply temperature.

The on/off valves in UFH systems are typically thermostatic actuated wax valves. The opening and closing cycle are for these valves a linear motion over 100s, meaning that it takes 100s from the control action is performed to the valve is 100% open or closed. This slow response of cause needs to be represented in the model. Furthermore, there are large transport delays associated with moving the water from the heat pump through the floor and back. Typically, is it recommended to use no more than 100 meters of pip pr. zone. if assuming that each zone has 100 meters of 17mm pipe, and the pump supplying water to the system at constant pressure where the flow is 150 L/hour, then the transport delay through the system is 720 s. This is not a problem if the supply temperature is constant. However, this also means that when changing the supply temperature, then it will take 12 minutes for the water with the new temperature to get all the way back to the heat pump when the valve is open. Adjusting the supply temperature is naturally also associated with some kind of dynamic when it is assumed that this is linear, and the response time is 60 s.

The dynamics of the heat pump are also relevant for this control problem because part of the control objectives is to reduce the price of heating. As described in Paper C [14] *"The price of heating with a heat pump can be simulated by knowing the cost of electricity, the dynamics of a heat pump, and the power consumption of the system. The cost of electricity is assumed to fluctuate during the day. The average danish price of electricity during the day can be seen in Figure 3.2a [1] . The dynamics of a heat pump can be described with the coefficient of performance (COP) which is a function of the ambient temperature and the supply temperature. The COP as a function of the ambient temperature can be seen in Figure 3.2b [26]. Additionally it is necessary to describe the part load factor (PLF). The PLF indicates how efficient the heat pump is dependent on the duty cycle, this can be seen in Figure 3.2c [41]."*

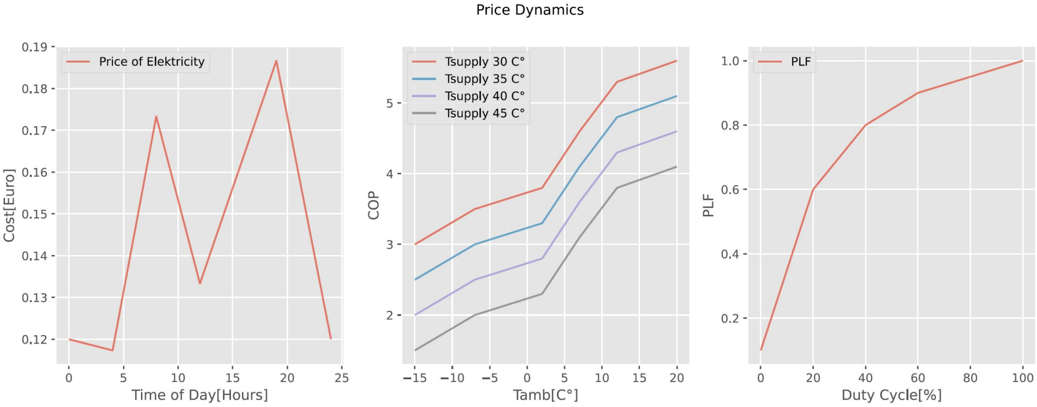


Fig. 3.2: Paper C [14] "Dynamics of a heat pump: Figure a) shows the average electricity prices including taxes in Denmark as a function of the time of day(tod). Figure b) Shows the COP as a function of the ambient temperature, for four different supply temperatures. Figure c) show the PLF as a function of the duty cycle(D)".

As stated, one of the objectives of the RL algorithm is to optimize the cost of heating.

For this reason, is it necessary to know how much energy the system consumes. In this thesis the flow is used to calculate the energy consumption, see Eq. (3.1)

$$\Delta E = (T_{supply} - T_{return}) \cdot Cp_{water} \cdot Q_{flow} \cdot \Delta T \tag{3.1}$$

$$cost = \frac{\Delta E}{COP(T_{amb}, T_{supply}) \cdot PLF(D)} \cdot CE(tod) \tag{3.2}$$

With the dynamics of the COP and a method to calculate the energy consumption, the price of heating using a heat pump can be calculated see Eq. (3.2).

The most important dynamics of the HVAC system is described above. Other factors such as wear and tear, clogging, etc. are not included in this thesis or the paper that support this thesis. Following this is the description of the building dynamic.

2.2 Building Dynamics

Building dynamics mostly concerns the transfer of thermal energy. With the above-described HVAC system energy can be supplied to the rooms by opening the valve to the specific zone and allowing hot water to run through the pipes in the floor. Heat energy will then flow into the floor and then into the room. This is naturally associated with some delay and slow dynamic response. This delay and response is deponent on how the pips are integrated into the floor. It is common to put the pipes into the concrete floor when casting the floor. This is referred to as type A in Figure 3.3. Alternatively, the pipes can be put on top of the floor foundation, these types are types of UFH systems are referred Type B or Ultra Fast UFH.

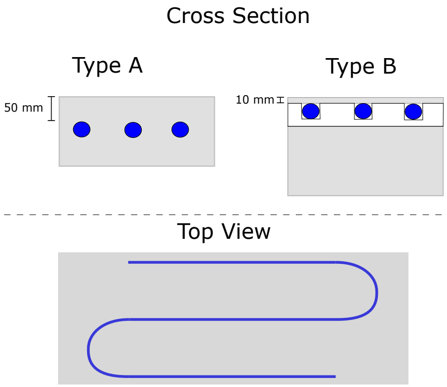


Fig. 3.3: Illustration of floor types, type A refers to pipes casted into the concrete flooring, type B refers to fast UFH, with minimal concrete on top of the piping.

The time constant for a UFH system can, depend on the type of floor, be between 30 minutes and 3 hours [28] hence the reason for using adaptive control for these systems.

The dynamic of the control system is now described, however, the dynamics of the disturbances for a building are equally important. There are six important factors when considering the disturbances in building dynamics:

- Radiation
- The ambient temperature
- Rain
- Wind
- Seasonal change
- User behavior

How a building environment will be affected by these factors naturally depend on how and with which materials the building is constructed, and how it is used.

3 Environments

The purpose of this section is to describe under which conditions the results of the papers presented in Chapter 4 are gathered.

This thesis uses 3 environments, two simulation environments, and one real-world environment. The first simulation is a simulink environment, this environment is used in Paper A. The simulink is not elaborated further on. In Paper B-D is a dymola environment used and in Paper E is the real-world environment used.

3.1 Dymola Simulation Environment

The theory and results in the following section are based on excerpts from Paper C - "A Multi-Agent Reinforcement Learning Approach to Price and Comfort Optimization in HVAC-Systems" published in the Open Access journal Energies.

Dymola is a Modelica-based multi-physics simulation software and, as such suitable for simulating complex systems and processes. Several libraries have been developed for Dymola. For the simulations in this work, the standard Modelica library and the Modelica Buildings library are used.

The simulation can be split into two parts: 1) the hydraulic part and 2) the thermodynamic part. The hydraulic part of the simulation can be described by a mixing loop, a pump, one valve per temperature zone, and some length of pipe per temperature zone.

The thermodynamic side of the simulation is constructed using the base element *"ReducedOrder.RC.TwoElements"* [52]. This element includes heat transfer from exterior walls, windows, and interior walls to the room. It furthermore includes radiation from the outside temperature and radiation from the sun. This means that wind and rain do not affect the simulation, as they are assessed to be smaller disturbances. These disturbances are not always negligible, but for the purpose of this work it is assumed to be. The simulation results will still indicate the saving potential that can be expected in real-life installation. The element is made in accordance with "VDI 6007 Part 1" which is the European standard for calculating transient thermal response of rooms and buildings [52].

The length of pipe used in each zone and parameters for the windows, walls, zone area, and volume are shown in Table 3.1.

Table 3.1: [14]"Parameters used for each temperature zone. A and B refer to if it is the one-zone simulation or the four-zone simulation."

Parameter	Zone1A	Zone1B	Zone2B	Zone3B	Zone4B
Length of pipe	$105m$	$56m$	$105m$	$42m$	$70m$
Window area	$20m^2$	$12m^2$	$25m^2$	$12m^2$	$24m^2$
Wall area	$39m^2$	$36m^2$	$40m^2$	$12m^2$	$30m^2$
Zone area	$30m^2$	$16m^2$	$30m^2$	$12m^2$	$20m^2$
Zone volume	$80m^3$	$48m^3$	$90m^3$	$36m^3$	$60m^3$

To simulate how the room receives heat from the floor, a floor element has been constructed. The floor element incorporates the pipe length, pipe diameter, floor thickness, floor area, and construction material. These parameters enable the floor element to simulate how the heat from the water running in the pipes will transfer through the concrete and into the room. The heat in the room is assumed to be uniformly distributed. This means that the temperature at the floor, at walls, and at the sealing of the room is the same. Modelling the temperature distribution uniformly is also in accordance with "VDI 6007part1".

3.2 Evaluation of Simulation

It is not possible to validate the simulation environment with data from a real-world system. We can, however, evaluate step responses to evaluate the dynamic convection of heat from the water in the pipes to the air in the room. Additionally, we can evaluate the amount of power the rooms require and compare it to a real-world house. Lastly, the daily and seasonal power consumption can be evaluated.

To evaluate the simulation environment, a run of the simulation with hysteresis control on the valves and an outdoor compensated supply temperature is executed. Note, that hysteresis control is the control method traditionally used in UFH system.

For the validation, a simulation with one temperature zone system is used. However, a similar simulation has been made for a four-temperature zone system with similar results.

All simulations are made with a hysteresis control with reference point 22°C and a dead band of ± 0.1°C. The outside compensated supply temperature is following a linear model, see Eq. (3.3).

$$T_{supply} = -0.6 \cdot T_{ambient} + 42 \tag{3.3}$$

Firstly, the room temperature of an entire heat season is plotted in Figure 3.4.

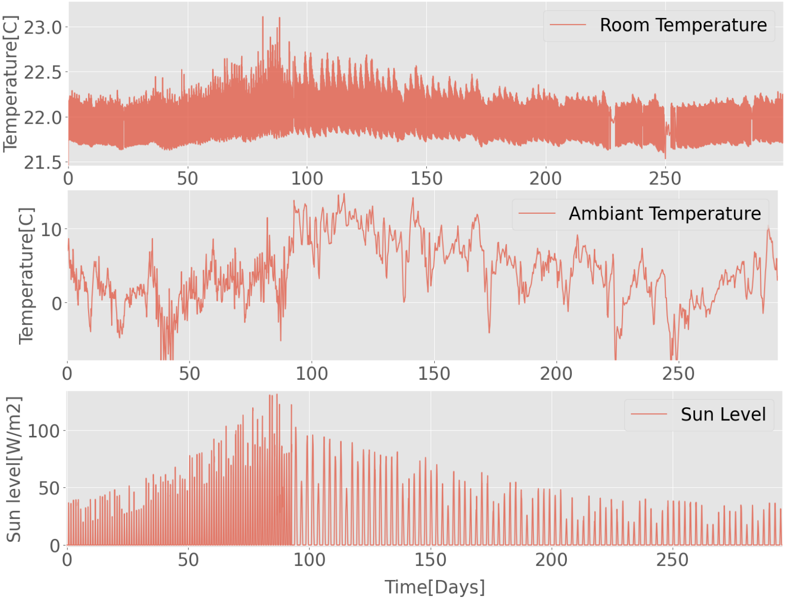


Fig. 3.4: [14]"Simulation results of one heating season, with a traditional controller and outside compensated supply temperature."

From Figure 3.4 it can be seen that a heating season is approximately 280 days. The heating season is in this work defined as the period of the year where the building needs energy to sustain a zone temperature of 22 °C. The simulation starts March 1st, and the period from June 1st to September 1st has been removed from the weather file as no heat is needed in this period. The seasonal effect can be seen in the figure, where occasional overshoots happen in the period from day 70 to day 140. Hence, in the Fall/Spring period, where heat is needed during the night and morning but not during the daytime, overshoots happen. The temperature is otherwise oscillating between 21.7 °C and 22.2 °C.

To investigate the response more closely the room temperature and the associated valve position are plotted over a period of two hours and 30 minutes, see Figure 3.5.

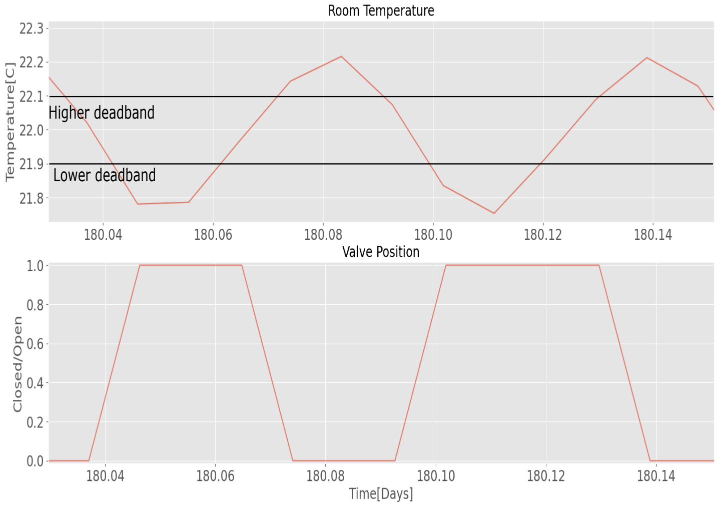


Fig. 3.5: [14]"Room temperature and associated valve position over a period of 8640s (0.1 days). By investigating the graph the dynamic response can be analyzed."

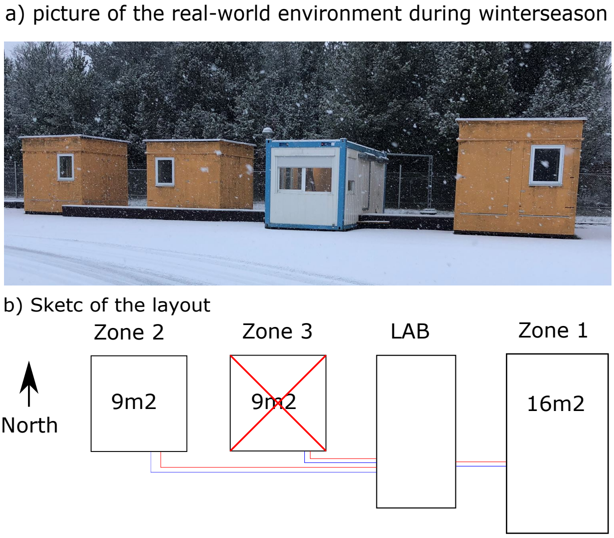
Figure 3.5 shows that when a valve is opened 1700s (0.02 days) will pass before the temperature gradient in the room becomes positive. Additionally, it can be seen that when the valve is closed the temperature will continue to rise an additional 0.1°C and about another 1700s will pass before the gradient becomes negative. This behavior is due to the slow dynamic properties that are expected of a UFH system, and therefore it also validates that this simulation resembles the typical dynamics of a UFH system [28].

lastly to verify the simulation environment the power consumption is reviewed. The temperature zone is 30 m^2 and consumes 3561 kWh over a heating season at a reference temperature of 22 °C, meaning an average of 118 kWh per m^2. An average Danish house uses 115 to 130 kWh per m^2 [2],which shows that the simulation is within what is considered average in a Danish climate.

3.3 Real-world Test Environment

The real world test environment is located at Marinus Svendsensvej 2, 8850 Bjerringbro, Denmark. Here we have set up 3 heating zones that are, from a thermal point of view, independent. However, they are all supplied by the same heating source, so from a hydraulic point of view they are connected.

In Figure 3.6 a) the real-world environment can be seen during the winter season. In the picture the 4 buildings this environment consists of can be seen. Tree heating zones and one laboratory. This can also clearly be seen from the layout sketch in Figure 3.6 b. Unfortunately due to installation complications have Zone 3 not been included in the experiments.


a) picture of the real-world environment during winterseason

b) Sketc of the layout

Fig. 3.6: In Figure a) can a picture of the test environment during winter be seen. 4 small buildings can be seen in the picture. In Figure b) can the layout of the buildings be seen and how they are hydraulic connected."

In Figure 3.7 a sketch of the hydraulic installation be seen.

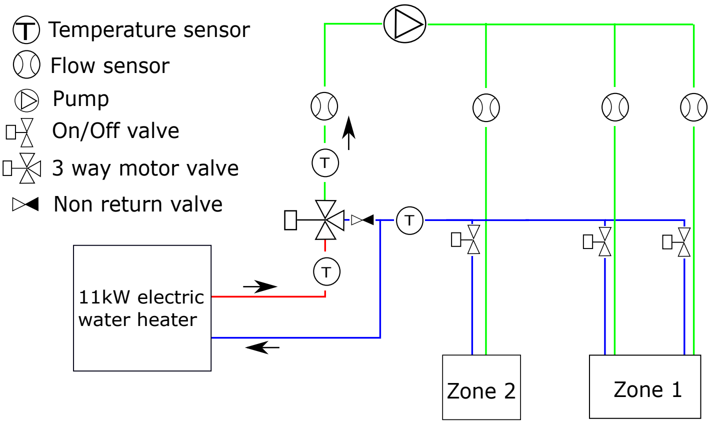


Fig. 3.7: Illustration of the hydraulic setup in the test environment. the arrows indicate the flow direction, red is the hot supply water green is the mixed water and blue is the cooled return water. the position of the sensors can also be seen in the illustration.

From the figure can it be seen where the temperature and flow sensors installed in the system.

Data collection and hardware

The real world test environment consist of tree data collection devises, one low level control device, one cloud controller and a local server. An illustration of this data collection setup can be seen in Figure 3.8.

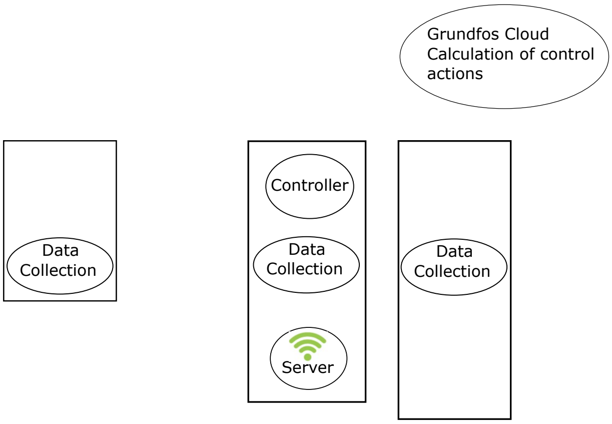


Fig. 3.8: Illustration of the hardware setup for the conducted real-world experiment.

The two of the tree data collection devices is configured alike and are placed in each temperature zone. The sensor input collected by these devices can be seen below:

- Room Temperature
- Supply Temperature
- Return Temperature
- Floor Temperature
- Lumen Sensor

The third data collection device is located in the lab. This device collects the following sensor inputs:

- Flow
- Supply Temperature
- Return Temperature
- Ambient Temperature

The server device is a Raspberry Pi 4b. This device creates a local Wi-Fi signal that the data collection devises connects to and sends their sensor signals to the server. The server has Grafana installed and is connected to the internet, making the data easy accessible through Grafana or an API. The code for the data collection devices, the server, and the API can be found in (GIT).

The low-level control device is also a Raspberry Pi 4b. This controller can perform the control actions the controller located in the Grundfos Cloud calculates. The low level controller can perform the following control actions:

- Control of Supply Temperature
- Open/close Valve 1
- Open/close Valve 2
- Open/close Valve 3

For the supply, temperature is a PI controller designed with sensible PI parameters. The supply temperature and the position of the valves are calculated in the Grundfos cloud and transmitted to the low level controller over an API. Both the low level controller and the cloud controller have access to the sensor data over the Grafana API.

In this chapter the fundamental problems associated with the control of HVAC systems are analysed. Additional has the two environments this thesis is based on been presented. The following chapter will present the solution to the research questions presented in Chapter 1.

Chapter 4

Solution

This chapter presents the results of the five papers that support this thesis. These results answer the five research questions formulated in Chapter 1. This chapter is structured with five sections, one for each paper. Each section includes an opening, an extended abstract, and closing remarks. This chapter includes excerpts from the papers supporting this thesis, making Part I of the thesis self-contained and readable without having to read the supporting papers.

1 How does model-free Reinforcement Learning perform in an Underfloor Heating System?

In this section the results of Paper A "Control of HVAC-systems with Slow Thermodynamic Using Reinforcement Learning" are presented. This paper is was presented at the *29th International Conference on Flexible Automation and Intelligent Manufacturing* and is published in *Procedia Manufacturing.* This section contains excerpts from the above mentioned paper.

The paper answers the first of the four research questions. It is found that eligibility traces are performing well in UFH systems. However, this paper also serves as a background study where issues/problems in RL-based control for HVAC are found. Based on this study is research questions 2-4 formulated.

1.1 Extended Abstract

This study is conducted in the "Simulink simulation environment". This environment is as described in Chapter 3 Section 1 a simplified environment, however for an initial study of how RL performs in Underfloor Heating and which methods are performing best is this environment sufficient.

In Chapter 2 is it argued that HVAC systems and especially underfloor heating is associated with slow and delayed responses. For this reason, is Q learning and Q learning with eligibility traces examined. Eligibility traces increase the agent's ability to correlate rewards received with proviso actions given and not only the action given at t-1. This is also elaborated in Section 2.

The objective of the RL algorithm is to keep the room temperature as close to $22C^{\circ}$ as possible. The objective function for the algorithm can be seen in (4.1).

$$R = \sum_{1}^{n} 1 - (T_{ref} - T_{room,n})^2 \tag{4.1}$$

This objective is simple when compared to the algorithms in Paper B-E, however, because the purpose of this paper is an examination of RL in HVAC is can it be used.

The architecture of the RL algorithm can be seen in Figure 4.1.

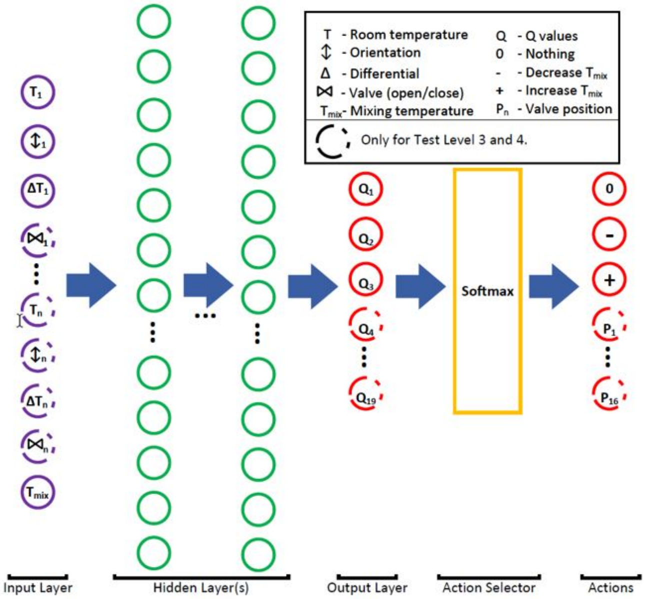


Fig. 4.1: [16]Illustration of the neural network, with input states; Room temperature, orientation to reference temperature, differential of room temperature, valve position, and mixing temperature. And output Q values and the Softmax action selector. The possible actions are; increase T_{mix}, decrease T_{mix}, open $valve_n$, close $valve_n$.

From Figure 4.1 it can be seen that a deep neural network is used. The activation function used in the neurons are "ReLu", different types of activation functions have been tried. From the empirical test was it found that "Relu" performed best. The exploration mechanism used is a softmax function. This exploration mechanism is elaborated in Section 2. However, the softmax is only used in Paper A, in Paper B-E is epsilon greedy exploration used. The algorithms tested in this paper have 2 hidden layers with 64 hidden neurons in each layer.

The results of Paper A include simulations of a 1 temperature zone system a 2-temperature zone system and a 4-temperature zone system. However, the conclusion can be derived from the simulations in the 1 and 2 temperature systems therefore only these results are presented here.

The Simulink environment is only affected by the ambient temperature, hence the reason for calling it simplified. Furthermore, is the ambient temperature constant the first $1 \cdot 10^7$S(115 days), then weekly oscillations of $\pm 3C^\circ$ are applied and at $1.5 \cdot 10^7$S(173 days) daily oscillations of $\pm 3C^\circ$ are applied. The reasoning behind doing this was, that it would make it easier for the agent to converge first to an environment where there are no disturbances and then to a little disturbance and finally to more disturbance. This can be seen from Figure 4.2.

In the simulation with 1 temperature zone, Q-learning, and Q-learning with eligibility traces tested and compared, 4 simulations are made, 2 with Q-learning and 2 with Q-learning and eligibility traces. The simulation time is $2 \cdot 10^7$S $\approx$ 230 days. The results of simulations 1 and 3 can be seen in Figure 4.3.

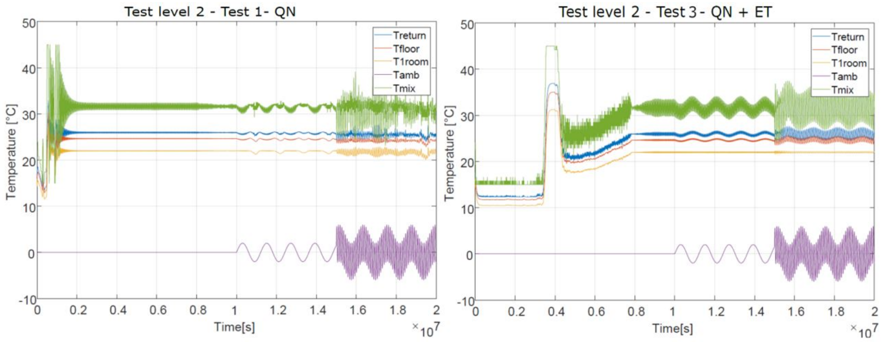


Fig. 4.2: [16]Results of one temperature zone simulink simulation, were RL control is applied.

In addition to the plots showed in Figure 4.2 are the standard deviation and mean of the 4-simulation displayed in Table 4.1.

Table 4.1: [16]Results of the one temperature zone simulink simulations. Two simulations with random seed with and with out eligibility traces are presented.

Test	Model Type	$T_{room1}[\bar{x}, \sigma]$	$T_{mix}[\bar{x}]$
1	QN	[21.52,0.54]	[40.47]
2	QN	[22.44,0.54]	[38.74]
3	QN+ET	[22.04,0.11]	[36,61]
4	QN+ET	[22.09,0.47]	[42.73]

From Figure 4.2 and Table 4.1 can it be seen that the Q-network with eligibility traces are performing better, the mean is closer to $22C^{\circ}$ in both test 3 and test 4, additional is the oscillations smaller.

In the simulation with two temperature zones the same tests are made. The simulation time is $2 \cdot 10^7$S $\approx$ 920 days. The results of simulation 1 and simulation 3 can be seen in Figure 4.3.

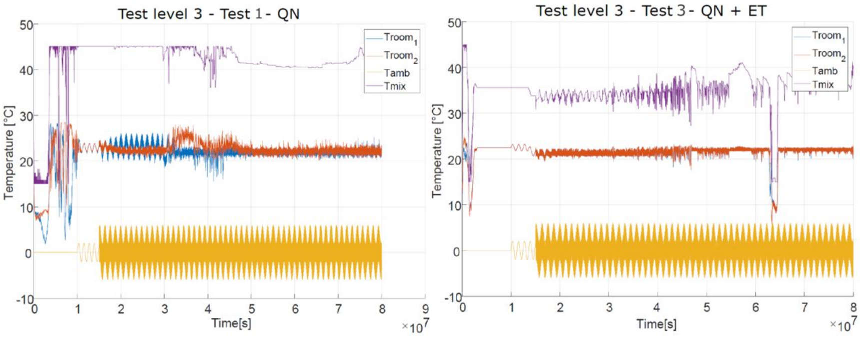


Fig. 4.3: [16]Results of two temperature zone simulink simulation, were RL control is applied.

To estimate how well each of the 4 simulations is performing the standard deviation and the average room temperature for each zone have been calculated, this can be seen in table 4.2.

Table 4.2: [16]Results of the two temperature zone simulink simulations. Two simulations with random seed with and with out eligibility traces are presented.

Test	Model Type	$T_{room1}[\bar{x}, \sigma]$	$T_{room2}[\bar{x}, \sigma]$	$T_{mix}[\bar{x}]$
1	QN	[20.12,2.11]	[21.8,1.67]	[40.47]
2	QN	[22.38,0.62]	[22.35,0.63]	[38.74]
3	QN+ET	[21.91,0.34]	[21.87,0.36]	[36,61]
4	QN+ET	[22.03,0.58]	[22.19,0.55]	[42.73]

From Figure 4.3 and Table 4.2 can it be seen that test 3 and 4 with eligibility traces are performing better then test 1 and test 2 in both the mean value and standard deviation.

From Figure 4.3 it can also be seen that a significant drop in performance around $6.5 \cdot 10^7$S this is to be expected because the algorithm is not limited and can therefore explore every part of the action/state space. This observation gives merit to research question number 2 regarding robustness.

When comparing Figure 4.2 and Figure 4.3 it can be seen that the converge time is much higher for the 2-zone system, this is also expected because the action state space is much larger for a 2 zone system. This gives merit to research question number 2 regarding Multi Agent RL.

1.2 Closing Remarks

As described in the introduction to this paper is this study an initial study to investigate how RL performs in HVAC and to investigate which issues/problems are worth investigating. It is from this study research question 2,3 and 4 has been formulated. So even dog the results are not as strong when compared to paper B-E and the simulation environment is to simple to make any conclusions as to how a RL controller will perform in a real-world scenario, this study has been vital for the results of Paper B-E.

2 How can a framework for robustness in Reinforcement Learning for HVAC systems be designed?

In this section is the results of Paper B *Control of HVAC-Systems Using Reinforcement Learning With Hysteresis and Tolerance Control* presented. This paper was presented at the 2020 *IEEE International Symposium on System Integration* and published in *IEEE Xplore*. This section contains excerpts from the above mentioned paper.

The paper answers the second of the five research questions. This paper focuses on how to construct a robust RL framework for controlling the valve position. This means that the mixing temperature in this paper is a traditional outdoor compensated controller. A robust framework for the mixing temperature have also been developed. This framework is not presented in Paper B, but is presented at the ending of this section and uses the same basic principle. This paper uses the Dymola simulation environment presented in Section 3.1.

2.1 Extended Abstract

In this paper the idea is that the RL controller can operate within some defined limits, and when the RL controller moves outside these limit's another controller in this case a tolerance controller overwrites the RL controller. An illustration of this can be seen in Figure 4.4.

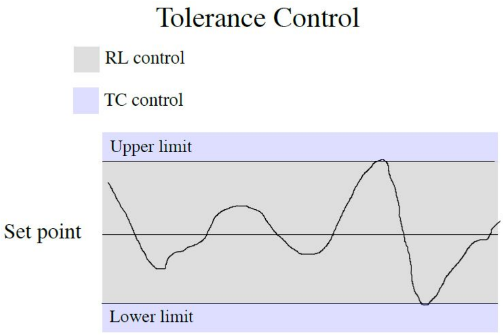


Fig. 4.4: [15]"Illustration of how the TC works as hard constraint in the AI control of a UFH system."

This control structure works well for UFH systems or similar systems where a default controller is available. However, this is naturally not always the case, therefore this principle can only be used in cases where a default controller is available.

How this framework is incorporated in the objective function affect performance significantly. Paper B explored how to incorporate soft constraints in the objective function so that the RL agent is aware when the default controller takes over.

For this purpose, 5 simulations are presented in Paper B;

- Simulation A: With the RL controller, but without the RL/TC framework
- Simulation B1: With the RL/TC framework, but no soft constraint.
- Simulation B2: With the RL/TC framework, and a constant value as a soft constraint.
- Simulation B3: With the RL/TC framework, and a linear soft constraint.
- Simulation C: The default controller(TC)

Simulation A and B1 uses Eq.(4.2) as reward function.

$$\mathrm{R}(T_z, V, H_c) = \begin{cases} 2-(T_z - T_{ref})\cdot & \text{if } 21.6 < T_z < 22 \\ -(T_z - T_{ref}) & \text{if } 21.6 > T_z \text{ or } T_z > 22 \end{cases}, \tag{4.2}$$

Form (4.2) can it be seen that no soft constraints are incorporated. Therefore, the agent is not aware, or at least, there is no information in the reward function indicating that the TC part of the RL/TC framework has taken over.

Simulation B2, B3 and B4 with the RL/TC framework is Eq. (4.3) and Eq. (4.4) used. Simulation B2 uses the soft constraint formulated in (4.5). Simulation B3 uses the soft constraint formulated in (4.6).

$$\mathrm{R}(T_z, V, H_c) = \begin{cases} 2 - (T_z - T_{ref}) \cdot & \text{if } 21.6 < T_z < 22 \\ -(T_z - T_{ref}) & \text{if } 21.6 > T_z \text{ or } T_z > 22 \\ -H_c & \text{if } SC = active \end{cases}, \tag{4.3}$$

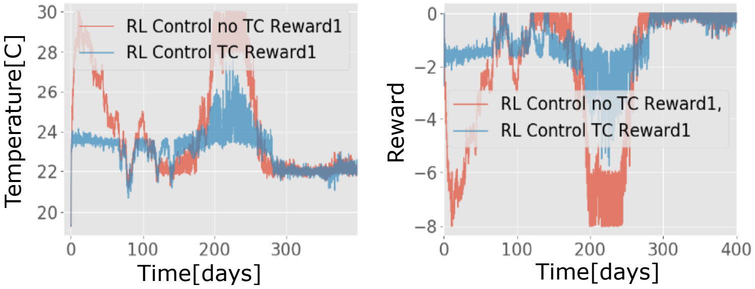
$$\mathrm{SC}(T_z, V) = \begin{cases} \text{not active} & \text{if } 21 < T_z > 23 \\ \text{active} & \text{if } T_z < 21 \text{ and } VP = 0 \\ \text{active} & \text{if } T_z < 23 \text{ and } VP = 1 \end{cases}, \tag{4.4}$$

$$H_c\ (\mathrm{SC}) = \quad 5 \qquad\qquad \text{if SC} = \text{active} \tag{4.5}$$

$$H_c\ (\mathrm{SC}) = \begin{cases} 1 + H_C & \text{if SC} = \text{active} \\ 5 & \text{if SC} = \text{not active} \end{cases} \tag{4.6}$$

In Simulation B2 is Eq. (4.5) used. From this can it be seen that the soft constraint is just a negative constant added if the agent violates the limits set in the RL/TC framework. In Simulation B3 is Eq. (4.6) used, this soft constraint is linear increasing as the agent continuously violates the limits set in the framework.

The Results of Simulation A and Simulation B can be seen in Figure 4.5.



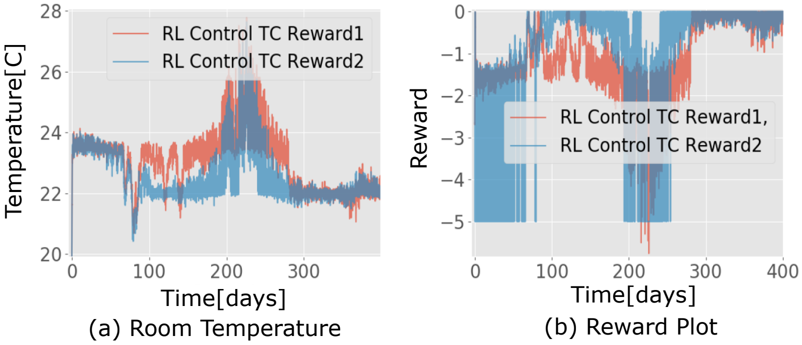
(a) Room Temperature

(b) Reward Plot

Fig. 4.5: [15]RL control without TC vs RL control with robustness framework, but without soft constraints.

From Figure 4.5 can it be seen that simulation A without the RL/TC framework converge faster, but is performing considerably worse than Simulation B where the RL/TC framework is included.

In Figure 4.6 can the results of Simulation B1 and Simulation B2 be seen in comparison.



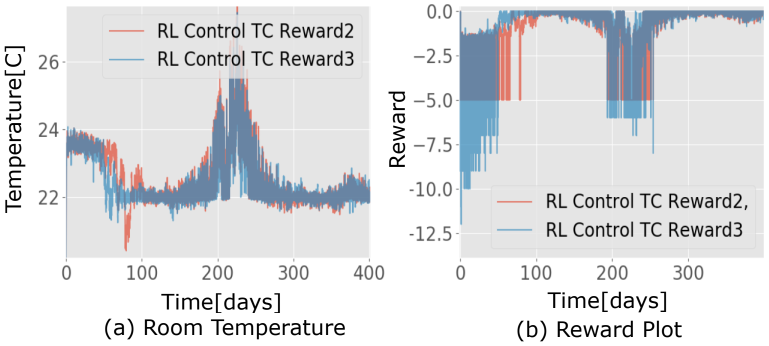
(a) Room Temperature

(b) Reward Plot

Fig. 4.6: [15]RL control with robustness framework with and with out soft constraints. Reward 1 referees to no soft constraint and Reward 2 referees to constant soft constraint.

From the results in Figure 4.6 can it be seen that the agent with soft constraint shown in Eq. (4.5) ("Reward 2") is perform significantly better when comparing convergence time. Simulation B1 converges after 280 days, and Simulation B2 converges after 110 days.

In Figure 4.7 can the results of Simulation B2 and Simulation B3 be seen in comparison.



(a) Room Temperature

(b) Reward Plot

Fig. 4.7: [15] RL control with the robustness framework, with constant soft constraint vs linear increasing soft constraint.

In Figure 4.7 can it be seen that the linear increasing soft constraint is performing better than the constant soft constraint. Simulation B3 converges after 60 days, whereas Simulation B2 converged after 110 days.

This concludes the results of Paper B. From the results can it be seen that the framework with linear increasing soft constraints is performing well. The same framework has been implemented with the same success for the supply temperature. Where the default controller is an outdoor compensated supply temperature.

2.2 Closing Remarks

This paper is a simple but robust solution for safe RL implemented. The main contribution is that by manipulating the soft constraint a considerable improvement to the performance can be observed. Additionally is it also in this paper the Dymola environment developed. This environment is used in the following papers.

3 How can Multi Agent Reinforcement Learning be used to reduce convergence time for RL HVAC control?

This section presents the results of Paper C *A Multi-Agent Reinforcement Learning Approach to Price and Comfort Optimization in HVAC-Systems*. This work is published in the Open Access journal *Energies*. This section contains excerpts from the above mentioned paper.

The results of this paper answers research question 3 and shows how MARL can reduce the convergence time significantly when compared to SARL. This paper is also the first paper in this thesis where the saving potential of RL in HVAC systems is investigated.

3.1 Extended Abstract

This paper uses a modified version of Q learning with eligibility traces as presented in Paper A. Additionally the Safety constraints developed in Paper B have also been implemented in this article.

The framework developed in this article is illustrated in Figure 4.8.

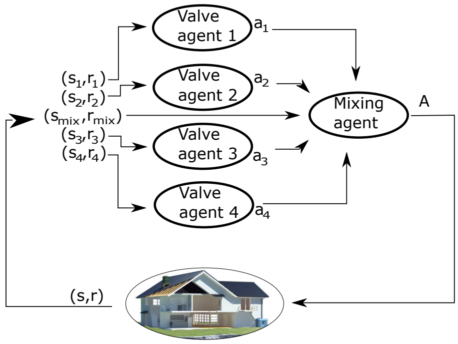


Fig. 4.8: [14]" Illustration of how the agents interact with each other and the environment. In the figure, four valve agents, one mixing agent, and a four-zone Underfloor Heating System can be seen. The sequence of interactions is as follows; all valve agents choose an action based on the state of the environment. These actions are passed to the mixing agent, the mixing agent chooses an action based on the state of the environment and the actions of the valve agents. All actions are passed to the environment and the environment returns states and rewards for the agents"

From Figure 4.8 it can be seen that the architecture of the agents are arranged such that the valve agents' action is chosen independent of each other and independent of the supply temperature. All valve agents report the chosen actions to the mixing agent, that based on these actions chose a mixing temperature for the system. This architecture reduces the complexity of the State/Action space significantly.

The mathematical formulation of the Q-functions for the architecture shown in Figure 4.8 can be seen in Eq. (4.7) and Eq. (4.8).

$$Q^{st}_{t+1}(s_{st}, a^{st}, a^{v_1..m}) = \mathbb{E}_{s,a,r,s'} \left[(1-\alpha)Q^{i}_{t}((a^{st}, a^{v_1..m}) + \alpha[r^{i}_{t} + \beta \max_{a^{st}} Q^{st}_{t}(s'^{st}, a^{st}, a^{v_1..m})]\right] \tag{4.7}$$

$$Q^{v_m}_{t+1}(s^{v_m}, a^{st}, a^{v_m}) = \mathbb{E}_{s,a,r,s'} \left[(1-\alpha)Q^{i}_{t}((s^{v_m}, a^{st}, a^{v_m}) + \alpha[r^{i}_{t} + \beta \max_{v_m} Q^{v_m}_{t}(s'^{v_m}, a^{st}, a^{v_m})]\right]. \tag{4.8}$$

From Eq. (4.7) and Eq. (4.8) it can be seen that both actions and states are split-up into local states and actions and only relevant data is shared between agents. The input states for the valve agent and supply agent can be seen below.

- Valve agent input:
  - Room Temp $i \in \{1, \cdots, n\}$, [°C];
  - Δ Room Temp $i \in \{1, \cdots, n\}$, [°C];

  - Hard constraint Valve $i \in \{1, \cdots, n\}$;
  - Supply temperature, [°C];
  - Ambient temperature, [°C];
  - Sun, [w/m^2];
  - Time of day, [hour and minutes].

- Supply agent input:
  - Room Temp$i \in \{1, \cdots, n\}$, [°C];
  - Δ Room Temp$i \in \{1, \cdots, n\}$, [°C];
  - Hard constraint Supply;
  - Ambient Temperature, [°C];
  - Sun, [w/m^2];
  - Time of day, [hour and minutes];
  - Price, [Euro].

From above it can be seen that each valve agent only receives states regarding the given zone that is being controlled. The supply agent receives information regarding each zone including the valve position and the price of heating.

The reward function for the valve agents can be seen in Eq. (4.9), with the two sub-functions in Eq. (4.10) and Eq. (4.11).

$$\mathrm{R}(T_z, V, H_c) = \begin{cases} 2 - (T_z - T_{ref})\cdot & \text{if } 21.6 < T_z < 22 \\ -(T_z - T_{ref}) & \text{if } 21.6 > T_z \text{ or } T_z > 22 \\ -H_c & \text{if } SC = active \end{cases}, \tag{4.9}$$

$$\mathrm{SC}(T_z, V) = \begin{cases} \text{not active} & \text{if } 21 < T_z > 23 \\ \text{active} & \text{if } T_z < 21 \text{ and } VP = 0 \\ \text{active} & \text{if } T_z < 23 \text{ and } VP = 1 \end{cases}, \tag{4.10}$$

$$H_c\ (\mathrm{SC}) = \begin{cases} 1 + H_C & \text{if SC = active} \\ 5 & \text{if SC = not active} \end{cases} \tag{4.11}$$

The abbreviations in the equations above are the following: R = Reward SC = Safety controller, T_z = Zone temperature, VP = Valve position, and H_c = Hard constraint.

The two sub-functions (4.10) and (4.11) are parts of the safety control and ensure a robust behavior. This is what is presented in Paper B.

Similar to the reward function for the valve agents the reward function for the supply agent can be seen in Eq. (4.12) with similar sub-functions Eq. (4.13) and Eq. (4.14).

$$R(T_z, V, H_c) = \begin{cases} 2 - (T_z - T_{ref}) - P & \text{if } 21.6 < T_z < 22 \text{ and } VP = 1 \\ -(T_z - T_{ref}) - P & \text{if } 21.6 > T_z \text{ or } T_z > 22 \end{cases} \tag{4.12}$$

$$SC(T_z, V) = \begin{cases} \text{not active} & \text{if } T_z, > 20.5 \\ \text{active} & \text{if } T_z, < 20.5 \text{ and VP} = 1 \end{cases} \tag{4.13}$$

$$H_c(SC) = \begin{cases} 1 + H_C & \text{if SC} = \text{active} \\ 5 & \text{if SC} = \text{not active} \end{cases} \tag{4.14}$$

The abbreviations in the equations above are the following: R = Reward SC = Safety controller, T_z = Zone temperature, VP = Valve position, H_c = Hard constraint and P=price.

From Equation (4.12), it is seen that the reward is much like the reward from the valve agent, with the difference that the +2 reward requires that the valve is open. When heating with a heat pump, it is optimal to have as much water circulation as possible. by Adding that the reward is highest when the valve is open enforces this behavior. The price is a scalar between 0 and 1, and the lower the price of heating, the better.

Like in the valve reward function, a safety controller is put on top of the RL algorithm. Simulation results show that it has some of the same effects as in the case of the valve agent reward. The safety controller is the *outside compensate supply temperature* used in the simulations in Section 3.2. The safety controller is activated whenever the temperature in a given zone is 1.5 °C lower than the reference temperature, and the associated valve is open.

Results

To verify that the above-described formulation of the learning problem reduces the training time/convergence rate is an experimental plan established. Five simulations are presented in this paper. Two of the simulations are made in a one-zone UFH system and three are made in a four zone UFH system. The five simulations and the purpose of each simulation are outlined Below:

- Simulation 1: MARL algorithm in one zone UFH system. This simulation will verify that the algorithm converges.
- Simulation 2: SARL algorithm in one zone UFH system. This simulation serves as a benchmark for the MARL algorithm.
- Simulation 5: Traditional control policy in one zone UFH system.

- Simulation 3: MARL algorithm in four-zone UFH system. This simulation will verify that it is possible to converge fast by reformulating the problem to a Markov Game.
- Simulation 4: SARL algorithm in four-zone UFH system. This simulation is a benchmark for simulation 3 and will show that MARL converges faster than SARL.
- Simulation 5: Traditional control policy in four-zone UFH system. This simulation will show the cost and comfort of a traditional control policy.

Figure 4.9 shows the results of Simulation 1 Simulation 2, and Simulation 3. From Figure 4.9 can it be seen that the MARL and SARL algorithms converge in approximately the same time, about 180 days. It can be seen from the figure that the reward signal oscillates after convergence, this is due to the seasonal changes in the price of heating, and how the reward function is formulated.

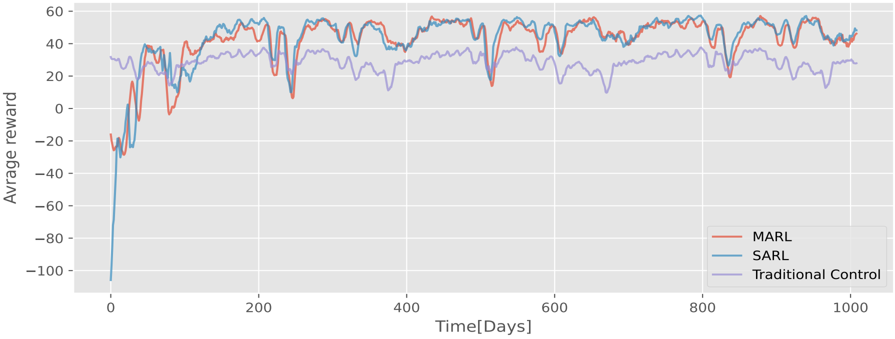


Fig. 4.9: [14]"Reward signal of the one-zone UFH system. Simulation time is 1000 days."

That the SARL and MARL algorithms converge at approximately the same time is in accordance with the assumption that the convergence time should correlate with the size of the action state space. Since the distribution of agents in a one-zone system almost results in the same action state space as if it was one agent, the convergence time is more or less the same. The performance of the RL algorithms, compared to a traditional controller, is better after 40 days of training. However, a drop in performance is seen after 80 days. This drop is due to a change of seasons and therefore load conditions that are unknown to the RL. After 120 days, the RL algorithms are shown to perform better than the traditional controller. There are a few days during a heating season where the MARL and SARL controllers are not performing better than the traditional controller. This can be found around day 220 and 240. These days are exceptionally cold and therefore the system is in saturation and therefore the RL-controller cannot improve the performance. From Figure 4.9 can it also be seen that both RL algorithms converge to a higher reward than the traditional control policy.

In Figure 4.10 is the results of Simulation 4, Simulation 5 and Simulation 6 presented.

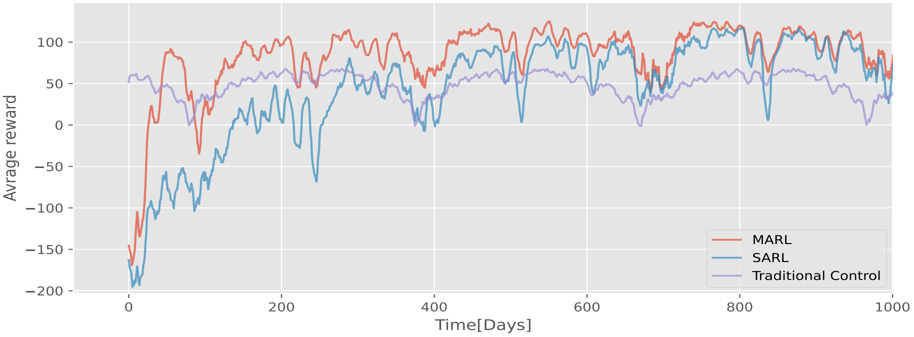
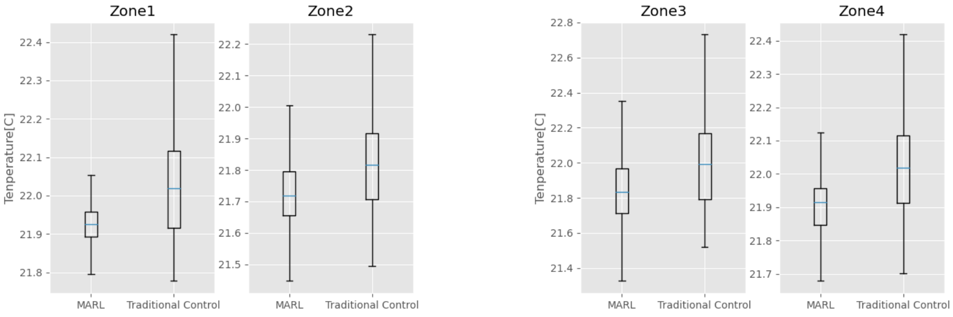


Fig. 4.10: [14]"Reward signal of the four-zone UFH system in, simulation time is 1000 days."

From the figure above can it be observed that the MARL agents converge after 180 days but are performing well after 40 days. The SARL agent converges to approximately the same as the MARL but after 600 days. This difference in convergence speed confirms the assumption on the relation between convergence speed and the size of the action space. Whereas SARL and MARL both works for a one-zone UFH system, the advantages with MARL are clear in the four-zone simulation. MARL results in faster convergence and marginal better convergence over 1000 day period.

Additional to the reward it is interesting to compare the RL performance to the traditional controller in terms of cost and comfort. For this comparison, MARL is compared to the traditional control policy.

To estimate the comfort is box plots of the temperature distribution of each of the 4 temperature zones made. These can be seen in Figure 4.11.



(a)

(b)

Fig. 4.11: [14]"Boxplots of temperature distribution in zone 1, zone 2,zone 3 and zone 4 in the four-zone UFH system for the MARL simulation and the simulation with a traditional control policy."

From the plots above, it can be deduced that the variation is about 40% less with MARL compared to a traditional controller. Note, outliers have been removed from the data before being used for the box plots. Smaller variations in the temperature give both better comfort and reduced price. Hence the MARL agents behave like this.

The cost of heating with the MARL algorithm, the traditional controller and the savings as a function of time can be seen in Figure 4.12. The cost is calculated based on the price of electricity, coefficient of performance (COP) and the partial load factor (PLF).

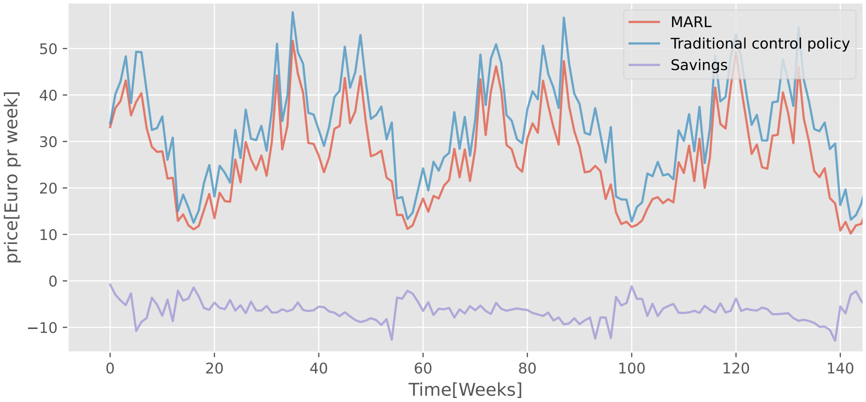


Fig. 4.12: [14]"Price of heating per week for MARL and traditional control for a four-zone UFH system over 1000 days period."

From this plot, it can be concluded that the savings vary over the season, but the MARL controller is performing better than the traditional controller at any point in time. The average savings are 19% when using MARL compared to a traditional controller. During the first 20 weeks can it be seen that the savings oscillate more than the remaining 120 weeks, the reason for this is naturally that the control policy of the MARL agent has not yet converged.

3.2 Closing Remarks

This paper is the first paper in this series where the savings potential of RL-based control for heat pump UFH based systems are presented. Additionally, have the convergence time been reduced significantly. These factors combined bring a control concept like this much closer to commercially viable.

4 How can Offline Reinforcement Learning be a data efficient method for HVAC control?

This section presents the results of Paper D *Data-Driven Offline Reinforcement Learning for HVAC-Systems*. This work has been submitted to the journal *Energy* and is currently under second round of review. In this paper the methods developed in Paper A, Paper B, and Paper C are used.

The results of this paper contribute to the field of RL for HVAC by presenting a framework for offline RL. This framework makes it possible to train the RL agents offline and deploy them into the real-world environment once it is certain that the performance is robust. The RL agents are also training online when deployed into the real-world system. This online training is categorized as online fine-tuning, which enables the algorithm to compensate for changes in the environment and include dynamics that were not a part of the offline training.

4.1 Extended Abstract

As mentioned, the idea is to train the agents offline before deploying them into the real-world environment. This means that an offline environment is required. This can be white-box simulation, grey-box simulation, or black-box simulation. Because this study strives to build commission-free algorithms has the work focused on a black-box simulation. A black-box simulation is a purely data-driven simulation, meaning that it is not required to set up the differential equations associated with the dynamics of a building. However, a black box model require data from the specific environment. This paper is therefore working under two scenarios:

- Scenario A: A new installation where there is no prior data from the environment. A Traditional control policy will run for a period to gather data.
- Scenario B: A recommission of an existing installation where prior operating data is available.

In Figure 4.13 scenario A can be seen. Scenario B can be derived from Figure 4.13 by removing step 1 where the "traditional controller" is interacting with the environment.

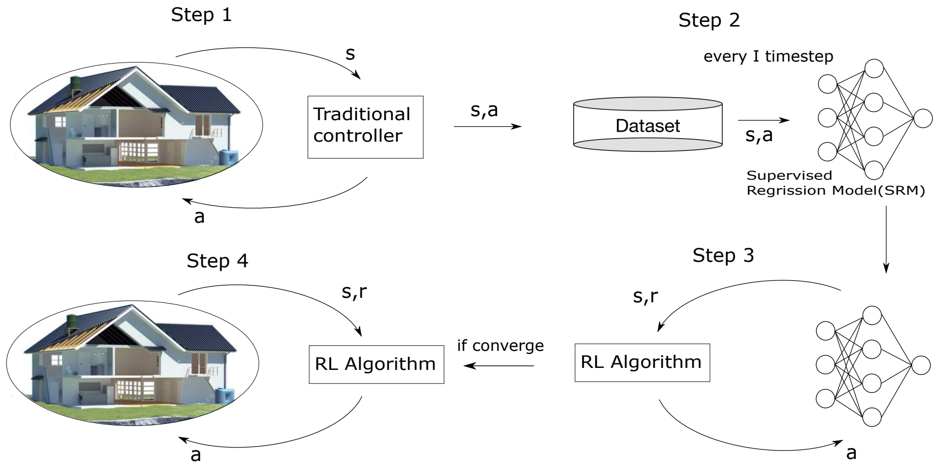


Fig. 4.13: Illustration of scenario A: The traditional controller interacts with the environment T time steps after each time step the action and state transition is saved in the data buffer. The data set is passed to the SRM ones trained, the SRM is used as an artificial environment for the RL agent to train until convergence. The trained agent is then deployed in the real environment, the agent can still do limited exploration for fine-tuning. Steps 2, 3, and 4 will be repeated until the SRM converges.

In Figure 4.13 Step 2 the black box model can be seen. This model is a Long Short Term Memory (LSTM) network which is a type of Recurrent Neural network (RNN) that enables events to be correlated over multiple time steps. The reason for this type of network is naturally that UFH systems are associated with slow and delayed responses. In Chapter 2 Section 4 the theory for LSTM networks is described. To verify that an LSTM network is suitable for this task a Multilayer Perceptron (MLP) network is also tested in Paper D. A MLP network is as explained in the background chapter a more traditional network for regression tasks. However, an MLP network does not have any ability to correlate events over multiple time steps.

The length of an episode in the RL framework is 30-time steps or 5 hours. Hence for the model to be useful it is required to predict the room temperature 30-time steps into the future. Because this is a control task, is it necessary for the model to predict every time step in between the current time and 30-time steps into the future dependent on which control actions are performed. In Figure 4.14 an illustration of such a model structure is showed.

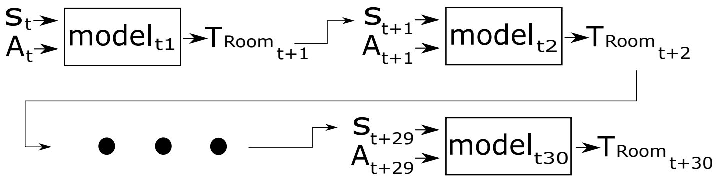


Fig. 4.14: Illustration of model architecture for supervised learning. This architecture will compensate for the prediction error that occurs in every time step.

From Figure 4.14 it can be seen that it is a different model for each time step. This is to compensate for the unavoidable error that will occur in each model. For this reason, a model is made for predicting each of the 30 time steps. Alternatively, to a model for each time-step, can this also be done with a single model that is used in all time-steps. To verify that a structure where a different model for each time step is suitable both methods are tested in Paper D.

Test of Black Box Models

To verify that the LSTM model with the model architecture presented in Figure 4.14 is suitable four tests are carried out. The prediction error for each time step for the four models are presented in Figure 4.15.

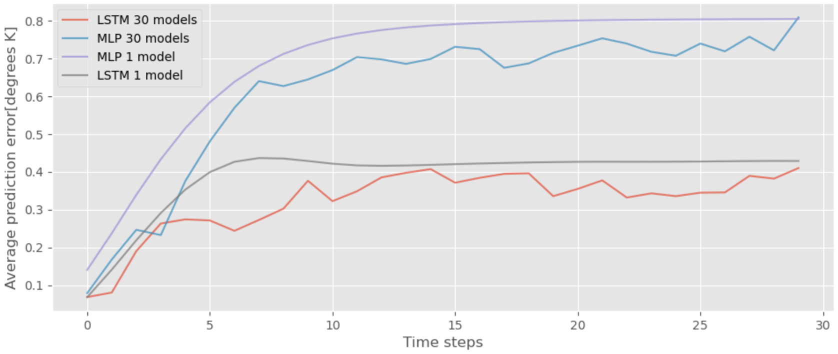


Fig. 4.15: Plot of the average prediction error for each time step. 4 plots can be seen, one for the 1 model LSTM, one for the 30 model's LSTM and the same for the MLP test. The data foundation is 60 days for the training of the model and 220 days for the evaluation.

The associated average prediction error pr time step is shown in Table 4.3.

Table 4.3: The average error pr. prediction, 30 time steps into the future, under the traditional control policy.($\frac{\sum error}{30}$)

	LSTM model	MLP
1 model	0.3894	0.6946
30 models	0.3246	0.6166

From Figure 4.15 and Table 4.3 it can be seen that the prediction error is 45% lower for the LSTM based model then the MLP model. Furthermore, it can be seen that by making a model for each time step and thereby compensating for prediction error the models perform 19% better on average.

Simulation Results

This section presents four simulations, two simulations with the RL/black box framework, one simulation only with the RL algorithm, and one simulation with a traditional controller. The four simulations are outlined below.

- Simulation 1: without RL/black box framework but with RL control. This simulation will serve as a benchmark for how the RL performs without training in the black box model environment.
- Simulation 2: with a traditional control policy, this will serve as a benchmark to estimate the RL algorithms capability to reduce heating costs while maintaining or increasing the comfort level.
- Simulation 3: with RL/black box framework, in scenario A.
- Simulation 4: with RL/black box framework, in scenario B with one heating season of data(280 days).

In Figure 4.16 the reward plot for simulation 1, Simulation 2 and Simulation 3 is shown.

From Figure 4.16 it can be seen that when using the RL/black box framework the performance is improved or equal to the normal MARL controller. Especially during the first 60 days, the performance is better. The reason for the improvement in this phase is that the RL/black box framework follows the traditional control policy. After approximately 580 days the MARL and RL/black box framework converge to approximately the same control policy.

In Figure 4.17 the results of simulation 1, Simulation 3 and Simulation 4 are shown.

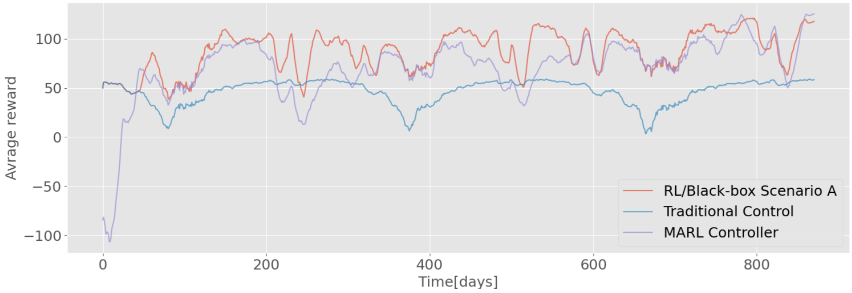


Fig. 4.16: Reward plot over 880 days for simulation 1, Simulation 2 and Simulation 3, where 1 is; MARL control, without the RL/black box framework, and 2 is; A traditional control policy. and 3 is; the RL/black box frame work in Scenario A.

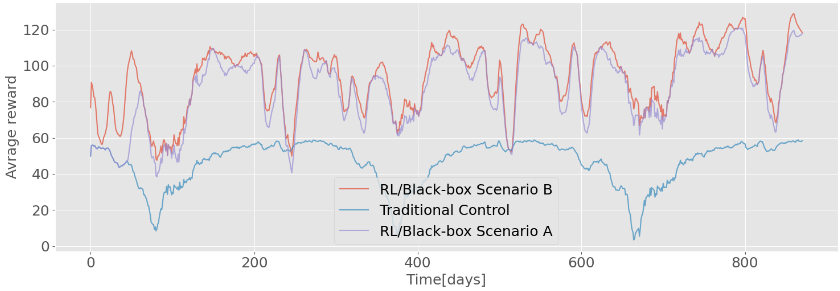


Fig. 4.17: Reward plot over 880 days for simulation 1, Simulation 3 and Simulation 4, where 1 is; MARL control, without the RL/black box framework, and 3 is; A traditional control policy. and 4 is; the RL/black box frame work in Scenario B.

From Figure 4.17 it can be seen that the RL/black box framework does perform better when more data is available. During the first period of 60 days is the performance of scenario B is notably better. After this period is the increase in performance only marginal. The reason for only a marginal increase is that the generated black box models do not become much better with the additional data.

4.2 Closing Remarks

This study marginally improves the convergence rate however it does improve the performance during early training, furthermore and more important does it allow for offline training. Offline training can be used in recommission tasks to deploy pre-trained agents based on data from the proviso system. Additionally, this framework can also be used in transfer learning situations where building dynamics and/or behavior patterns may be similar.

5 How will Offline RL perform in a Real-World field test?

This section present Paper E *A Field test of Offline Multi Agent Reinforcement Learning for HVAC-Systems* this is an answer to research question 5. This will be submitted to *Energy and Buildings.*

In Paper E is the methods presented in Paper A-D tested in a real-world field test. A bench marking test is firstly conducted, this bench marking test is used for comparison, furthermore is the data from the bench marking test used to train the offline RL policy.

5.1 Extended Abstract

This paper validate to some extent the research conducted in Paper A-D. The biggest concern when conducting real world experiments in HVAC systems are that a comparison between the bench marking test and the test with the RL policy are approximations since they have not been subjected to the same weather disturbances.

Bench-marking Test

This section presents the benchmarking test that is performed in the real-world environment, this test is performed with traditional controllers during spring 2021 and winter 2021/2022. The period of the test is from 1th of March to 21th of April and again from 1th of December to 4 January. In total, this is equivalent to 85 days of benchmarking data.

To conduct this benchmark test has two outdoor compensated supply temperatures, and hysteresis controllers been used. The functions for the outdoor compensated supply temperatures can be seen in (4.15) and (4.16). The hysteresis ban is set for $0.01C^{\circ}$ and the reference temperature is 22 C° for zone 1 and zone 2.

$$T_{supply} = -0.6 \cdot T_{ambient} + 45. \tag{4.15}$$

$$T_{supply} = -0.6 \cdot T_{ambient} + 38. \tag{4.16}$$

The reason two supply temperatures are used is that the first seen in Eq. (4.15) is used during the spring 2021 and the second seen in Eq. (4.16) is used during winter 2021/2022.

In the following is 5 sensor signals presented. In Figure 4.18 is the room temperature for each of the 2 zones presented. In Figure 4.19 is the Lux measurement for each zone presented and in Figure 4.20 is the ambient temperature plotted.

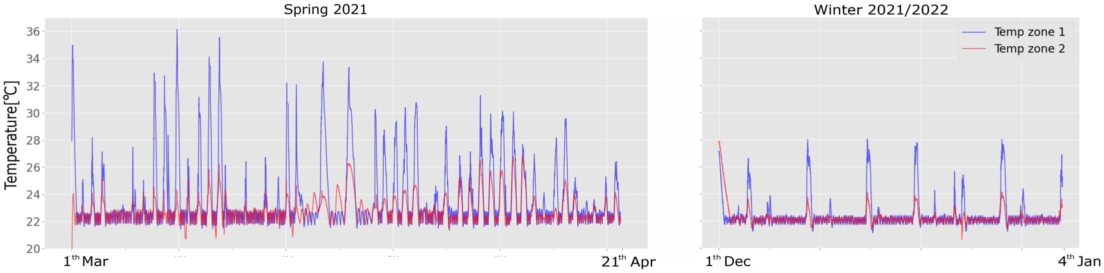


Fig. 4.18: Room temperature in each of the 2 temperature zones during the test period.

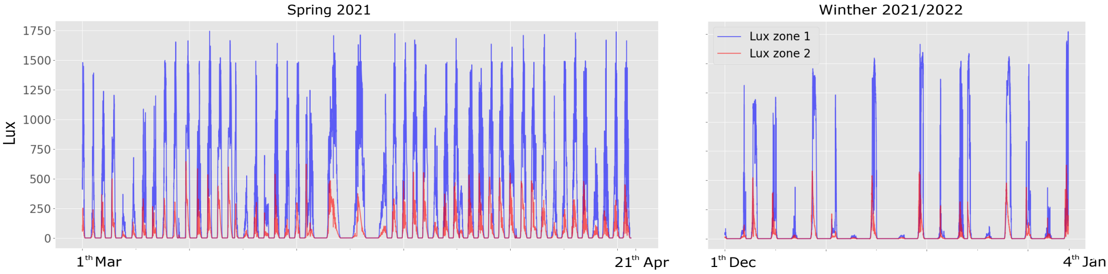


Fig. 4.19: Lux measurement in each of the 2 temperature zones during the test period.

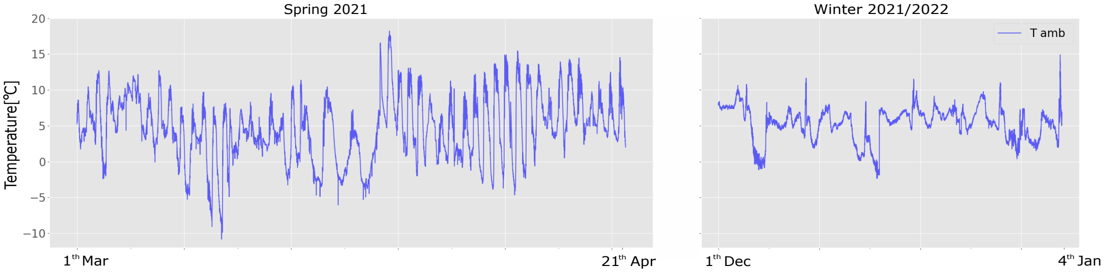


Fig. 4.20: The ambient temperature during the bench marking period.

With the data presented and the offline RL framework in Paper D have an offline RL policy been trained and deployed in the real world test envirnment. The results are presented in the following section.

Presentation of RL Data

In Figure 4.21 the room temperature in the two zones can be seen.

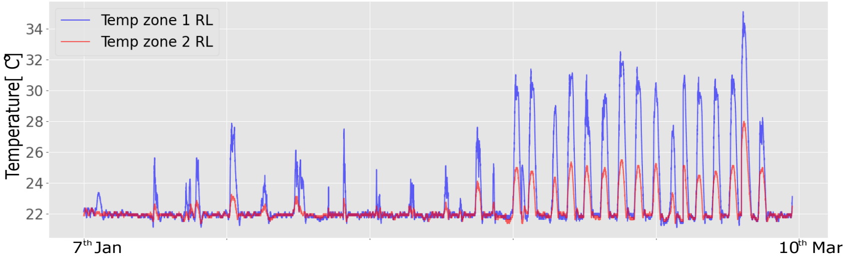


Fig. 4.21: Room temperature in each of the 2 temperature zones during the deployment of the RL algorithm.

From the room temperature plot can it be seen that the temperature is steady at 22 $\circ C$ except for periods where the radiation causes the temperature to rise.

In Figure 4.22 is the Lux measurements from the two zones presented.

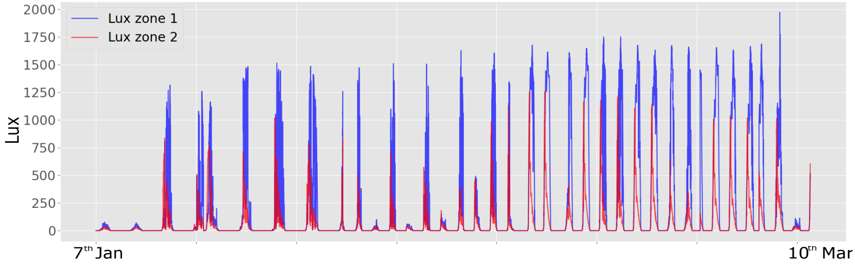


Fig. 4.22: Lux measurement in each of the 2 temperature zones during the deployment of the RL algorithm.

In Figure 4.22 it can be seen that the Lux measurements and the rise in temperature in Figure 4.21 are correlated.

In Figure 4.23 is the ambient temperature over the testing period shown

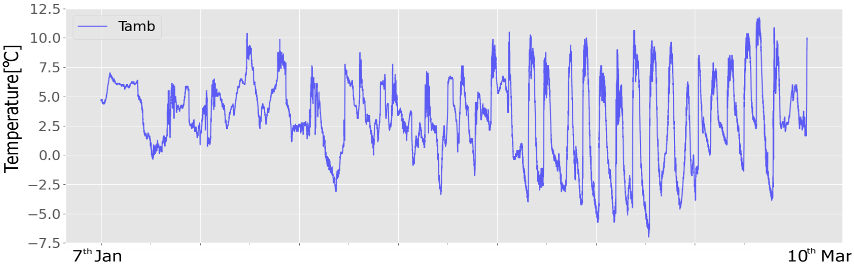


Fig. 4.23: The ambient temperatures during the deployment of the RL algorithm.

If comparing the ambient temperature from the RL deployment with the ambient temperature in the benchmarking in Figure 4.20 test it can be seen that some of the same tendencies can be seen.

Analysis and Comparison of Data

To asses if the RL control policy found with offline training is performing better than the policy used in the benchmarking test is the following investigated:

- Is the RL controller exhibiting predictive control-like behavior. this is investigated by looking at shorter time series of 24 hours.
- Compare the oscillations of the room temperatures in the benchmark data and the RL data during periods where the system is not saturated.
- Compare duty cycle and supply temperature to estimate if the RL control policy is reducing cost.

For this comparison is only benchmarking data from the winter period 2021 used. The reason for this is that two control policies are used in the benchmarking test. The control policy used during the winter is the best.

In Figure 4.24 is two 24 hours' time series shown. The time series are taken from the RL policy and benchmarking test, and periods have similar sun and ambient disturbances.

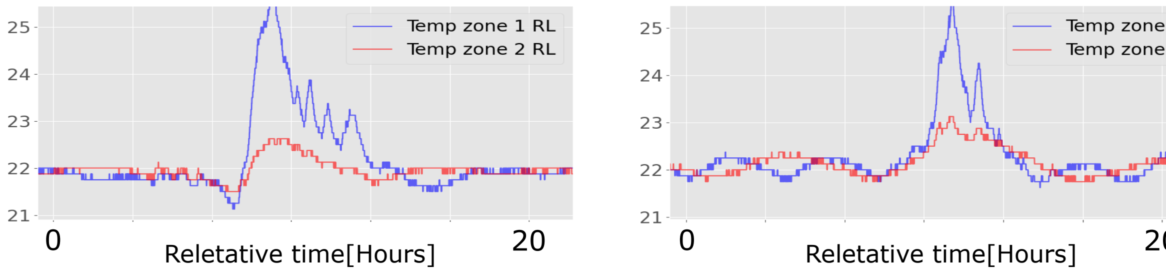


Fig. 4.24: 24 hour time series analysis. In this figure can 24 hours of room temperature data for two similar days for the RL control policy and the traditional control policy be seen.

In Figure 4.24 can it be seen that the RL policy reduces the temperature in the zones before the sun heats up the rooms, additional can it be seen that the oscillations are smaller. This indicates that the RL control policy exhibits predictive control-like behavior.

To verify that the RL control policy is performing better than the traditional policy is temperature distribution visualized in box plots seen in Figure 4.25.

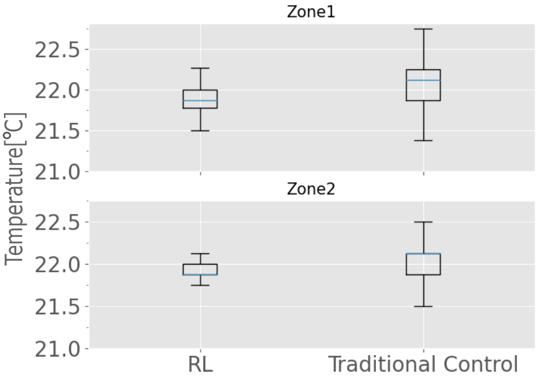


Fig. 4.25: Histograms showing the distribution of the room temperature for the RL control policy and the traditional control policy. The data used in these histograms are only data for periods where the system is not saturated. The system is estimated to be saturated when the temperature is above 23 $°C$ and the valves are closed.

The data visualized in Figure 4.25 is only from periods where the system is not saturated. In Figure 4.25 it can be seen that the distribution for both temperature zones is reduced. In zone 1 is the temperature distribution 43% lower and in zone 2 is 63% lower.

Table 4.4: Table showing the duty cycle over the periods of the benchmark test and the test of the RL policy. Additional is the duty cycle over when the system is not saturated shown.

	RL	RL not satuarted	TC	TC not saturated
Zone 1	63	88	42	65
Zone 2	67	83	43	63

Lastly, to estimate if the RL control policy is more optimal is the duty cycle investigated. The duty cycle is shown in Table 4.4.

In Table 4.4 it can be seen that the duty cycle for the RL policy is roughly 20% higher, this is naturally because a lower supply temperature is chosen by the RL controller.

Estimating the energy savings by comparing the price calculated in Eq.(3.2) cannot be done directly since the benchmarking test and the RL policy test have not been subject to the same disturbances. However, we can compare the duty cycle of the two tests. The relationship between the partial load factor (PLF) and the duty cycle can be seen in Figure 3.2, where the heat pump dynamics are elaborated on [41].

In Figure 3.2 can it be seen that the partial load factor is roughly 0.1 lower for the benchmarking test. The PLF has a linear relationship with the efficiency of the system [41], this is also seen in Eq. (3.2). Because of this, it can be concluded that the RL policy is 10% more energy efficient. If assuming that the price of electricity is the same for the two tests is the RL policy at least 10% more cost-efficient.

5.2 Closing Remarks

This Paper validates first and foremost that the offline RL policy is performing well when deployed in the real world environment. Additionally, is it shown that the RL policy is exhibiting predictive control like behavior and reduces the oscillations of the two zones 43 % and 63% respectively. Lastly is 10 % cost savings argued by investigating the duty cycle of the two test.

Chapter 5

Conclusion and Future Work

1 Research Contributions

This collection of papers summarized the contribution made during this PhD project. Three methods for data efficient and robust RL have been developed. These methods have additionally been field tested were they to some extent have been validated. How each of the three methods contributes to the field of model free optimal control of HVAC are outlined below:

- The robustness framework is a simple and safe method to ensure comfort regardless of the control policy of the RL agents. By manipulating the objective function of the RL agent with soft constraints, it is shown that the RL agent quickly learns that the action state space outside of the limits of the robustness framework is unfeasible. hence does this result in both fast and safe convergence.
- The Multi Agent RL framework makes RL for HVAC more scalable. It is shown that by formulating the HVAC environment as a Markov Game and by implementing a, from a system point of view, sensible reporting structure a 70% reduction of the convergence time can be archived. This framework is also supported by the above-mentioned robustness framework.
- The offline RL framework allows for more data efficient offline training. This framework uses a LSTM based regression model to model the action state space. By doing so is it shown that a near optimal control policy can be derived only from offline training. This framework also supports the MARL framework.

All of the above-mentioned contributions support each other therefore it has been possible to test all of the above mentioned methods in one field test. This field test is presented in Paper E and in this paper is it validated that the offline RL framework does converge to a better policy then the traditional control policy.

2 Future Research

Even dough this thesis presents a field test, is there still more verification work to be done before the final algorithm is commercially viable. Additionally, has there not been made any effort to reduce the computational power it requires to train the algorithms or reduced the amount of memory that is required to store it. Therefore, the following steps are suggested to mature this research:

- Minimize the amount of storage needed to host the algorithm.
- Minimize the computational power required to train the algorithm
- Conduct multiple full size field tests, in buildings with occupants.

Additional research that could improve the performance of RL-based control for HVAC is transfer learning. It may be possible to train an algorithm on data from a similar building. It will not converge 100%, due to the fact that the dynamic of two houses are never the same, but it may come close. The final training can then be done on data from the given building. This is similar to pre-training of the network.

Chapter 6

References

References

[1] “El-priser og afgifter.” [Online]. Available: https://www.vivaenergi.dk/el-priser-og-afgifter

[2] “Se det gns. varmeforbrug i husstande der ligner din,” Mar 2021. [Online]. Available: https://seas-nve.dk/kundeservice/forbrug/gennemsnitsforbrug/varmeforbrug/

[3] E. C. ., “Mapping and analyses of the current and future (2020 - 2030) heating/cooling fuel deployment,” 9 2016. [Online]. Available: https://ec.europa.eu/energy/sites/default/files/documents/mapping-hc-excecutivesummary.pdf

[4] A. Afram and F. Janabi-Sharifi, “Review of modeling methods for hvac systems,” *Applied Thermal Engineering*, vol. 67, no. 1, pp. 507–519, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359431114002348

[5] ——, “Theory and applications of hvac control systems – a review of model predictive control (mpc),” *Building and Environment*, vol. 72, pp. 343–355, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360132313003363

[6] ——, “Black-box modeling of residential hvac system and comparison of gray-box and black-box modeling methods,” *Energy and Buildings*, vol. 94, pp. 121–149, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778815001504

[7] ——, “Gray-box modeling and validation of residential hvac system for control system design,” *Applied Energy*, vol. 137, pp. 134–150, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261914010678

[8] N. P. Ankit Gupta, “Europe heat pump market size by product,” 2 2019. [Online]. Available: https://www.gminsights.com/industry-analysis/europe-heat-pump-market

[9] ——, “North america residential heat pump market size by product,” 2 2021. [Online]. Available: https://www.gminsights.com/industry-analysis/europe-heat-pump-market

[10] B. Banerjee and J. Peng, "Adaptive policy gradient in multiagent learning," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, 2003, pp. 686–692.

[11] E. Barrett and S. Linder, "Autonomous hvac control, a reinforcement learning approach," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2015, pp. 3–19.

[12] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[13] D. Bertsekas, "Multiagent reinforcement learning: Rollout and policy iteration," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 249–272, 2021.

[14] C. Blad, S. Bøgh, and C. Kallesøe, "A multi-agent reinforcement learning approach to price and comfort optimization in hvac-systems," *Energies*, vol. 14, no. 22, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/22/7491

[15] C. Blad, C. S. Kallesøe, and S. Bøgh, "Control of hvac-systems using reinforcement learning with hysteresis and tolerance control," in *2020 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2020, pp. 938–942.

[16] C. Blad, S. Koch, S. Ganeswarathas, C. Kallesøe, and S. Bøgh, "Control of hvac-systems with slow thermodynamic using reinforcement learning," *Procedia Manufacturing*, vol. 38, pp. 1308–1315, 2019.

[17] M. Bowling and M. Veloso, "Multiagent learning using a variable learning rate," *Artificial Intelligence*, vol. 136, no. 2, pp. 215–250, 2002.

[18] L. Buşoniu, R. Babuška, and B. De Schutter, "Multi-agent reinforcement learning: An overview," *Innovations in multi-agent systems and applications-1*, pp. 183–221, 2010.

[19] P. Checkland and J. Poulter, "Soft systems methodology," in *Systems approaches to making change: A practical guide*. Springer, 2020, pp. 201–253.

[20] R. H. Crites, A. G. Barto *et al.*, "Improving elevator performance using reinforcement learning," *Advances in neural information processing systems*, pp. 1017–1023, 1996.

[21] J. García, Fern, and o Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 42, pp. 1437–1480, 2015. [Online]. Available: http://jmlr.org/papers/v16/garcia15a.html

[22] S. Heiple and D. J. Sailor, "Using building energy simulation and geospatial modeling techniques to determine high resolution building sector energy consumption profiles," *Energy and Buildings*, vol. 40, no. 8, pp. 1426–1436, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778808000200

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] J. Hu, M. P. Wellman *et al.*, "Multiagent reinforcement learning: theoretical framework and an algorithm." in *ICML*, vol. 98. Citeseer, 1998, pp. 242–250.

[25] H. Huang, L. Chen, and E. Hu, "A new model predictive control scheme for energy and cost savings in commercial buildings: An airport terminal building case study," *Building and environment*, vol. 89, pp. 203–216, 2015.

[26] X. K. D. L. Jinzhe Nie, Zan Li, “Analysis and comparison study on different hfc refrigerants for space heating air source heat pump in rural residential buildings of north,” *Procedia Engineering*, vol. 205, pp. 1201–1206, 2017.

[27] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, “Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation,” 2018.

[28] T. M. Kull, R. Simson, M. Thalfeldt, and J. Kurnitski, “Influence of time constants on low energy buildings’ heating control,” *Energy Procedia*, vol. 132, pp. 75–80, 2017.

[29] A. Kusiak, G. Xu, and Z. Zhang, “Minimization of energy consumption in hvac systems with data-driven models and an interior-point method,” *Energy Conversion and Management*, vol. 85, pp. 146–153, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0196890414004646

[30] M. Lauer and M. Riedmiller, “An algorithm for distributed reinforcement learning in cooperative multi-agent systems,” in *In Proceedings of the Seventeenth International Conference on Machine Learning.* Citeseer, 2000.

[31] S. Levine, A. Kumar, G. Tucker, and J. Fu, “Offline reinforcement learning: Tutorial, review, and perspectives on open problems,” 2020.

[32] M. L. Littman, “Markov games as a framework for multi-agent reinforcement learning,” in *Machine learning proceedings 1994.* Elsevier, 1994, pp. 157–163.

[33] ——, “Value-function reinforcement learning in markov games,” *Cognitive systems research*, vol. 2, no. 1, pp. 55–66, 2001.

[34] M. J. Matarić, “Reinforcement learning in the multi-robot domain,” in *Robot colonies.* Springer, 1997, pp. 73–83.

[35] E. Mills, N. Bourassa, M. Piette, H. Friedman, T. Haasl, T. Powell, and D. Claridge, “The cost-effectiveness of commissioning new and existing commercial buildings: Lessons from 224 buildings,” *HPAC Engineering*, 11 2005.

[36] 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*, vol. 518, no. 7540, pp. 529–533, Feb. 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14236

[37] U. Nations, *The Sustainable Development Goals Report 2020.* United Nations, 2020. [Online]. Available: https://www.un-ilibrary.org/content/books/9789210049603

[38] U. D. of Energy., “Residential energy consumption survey 2015,” 5 2018. [Online]. Available: https://www.eia.gov/consumption/residential/reports/2015/methodology/pdf/RECSmethodology2015.pdf

[39] A. Overgaard, C. S. Kallesøe, J. D. Bendtsen, and B. K. Nielsen, “Mixing loop control using reinforcement learning,” in *E3S Web of Conferences*, vol. 111. EDP Sciences, 2019, p. 05013.

[40] A. Overgaard, B. K. Nielsen, C. S. Kallesøe, and J. D. Bendtsen, “Reinforcement learning for mixing loop control with flow variable eligibility trace,” in *2019 IEEE Conference on Control Technology and Applications (CCTA).* IEEE, 2019, pp. 1043–1048.

[41] K. Piechurski, M. Szulgowska-Zgrzywa, and J. Danielewicz, "The impact of the work under partial load on the energy efficiency of an air-to-water heat pump," 2017.

[42] S. Privara, 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.

[43] M. L. Puterman, "Chapter 8 markov decision processes," in *Stochastic Models*, ser. Handbooks in Operations Research and Management Science. Elsevier, 1990, vol. 2, pp. 331–434. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927050705801720

[44] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and Buildings*, vol. 40, no. 3, pp. 394–398, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778807001016

[45] T. Schaul, J. Quan, I. Antonoglou, and D. Silver.

[46] K. E. Seiferlein, "Annual energy review 2005," 7 2006. [Online]. Available: https://www.osti.gov/biblio/1212311

[47] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—a novel pooling deep rnn," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5271–5280, 2018.

[48] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[49] K. M. Tsui and S.-C. Chan, "Demand response optimization for smart home scheduling under real-time pricing," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1812–1821, 2012.

[50] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," 2015. [Online]. Available: http://arxiv.org/abs/1509.06461

[51] T. Wei, Y. Wang, and Q. Zhu, "Deep reinforcement learning for building hvac control," in *Proceedings of the 54th annual design automation conference 2017*, 2017, pp. 1–6.

[52] M. Wetter, W. Zuo, T. S. Nouidui, and X. Pang, "Modelica buildings library," *Journal of Building Performance Simulation*, vol. 7, no. 4, pp. 253–270, 2014.

[53] X. Yan, Q. Ren, and Q. Meng, "Iterative learning control in large scale hvac system," in *2010 8th World Congress on Intelligent Control and Automation*, 2010, pp. 5063–5066.

[54] L. Yu, T. Jiang, and Y. Zou, "Online energy management for a sustainable smart home with an hvac load and random occupancy," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 1646–1659, 2017.

[55] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 07 2019. [Online]. Available: https://doi.org/10.1162/neco_a_01199

[56] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *arXiv preprint arXiv:1911.10635*, 2019.

[57] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *International Conference on Machine Learning.* PMLR, 2018, pp. 5872–5881.

[58] Y. Zhang, C.-Q. He, B.-J. Tang, and Y.-M. Wei, “China’s energy consumption in the building sector: A life cycle approach,” *Energy and Buildings*, vol. 94, pp. 240–251, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778815002030

Part II

Papers

Paper A

PaperA

Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 38 (2019) 1308–1315

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2019), June 24-28, 2019, Limerick, Ireland.

Control of HVAC-systems with Slow Thermodynamic Using Reinforcement Learning

C. Blad[b,d]*, S. Koch[a], S. Ganeswarathas[a], C.S. Kallesøe[c,d], S. Bøgh[a,b]

[a]*Dept. of Materials and Production, Aalborg University, Fibigerstræde 16, Aalborg Øst, DK-9220, Denmark*
[b]*Robotics & Automation Group, Dept. of Materials and Production, Aalborg University, Fibigerstræde 16, Aalborg Øst, DK-9220, Denmark*
[c]*Dept. of Electronic systems, Aalborg Unicersity, Fredrik Bajersvej 7, Aalborg Øst, DK-9220, Denmark*
[d]*Grundfos A/S, Poul Due Jensens Vej 7, 8850 Bjerringbro*

Abstract

This paper proposes an adaptive controller based on Reinforcement Learning (RL), which copes with HVAC-systems consisting of slow thermodynamics. Two different RL algorithms with Q-Networks (QNs) are investigated. The HVAC-system is in this study an underfloor heating system. Underfloor heating is of great interest because it is very common in Scandinavia, but this research can be applied to a wide range of HVAC-systems, industrial processes and other control applications that are dominated by very slow dynamics. The environments consist of one, two, and four zones within a house in a simulation environment meaning that agents will be exposed to gradually more complex environments separated into test levels. The novelty of this paper is the incorporation of two different RL algorithms for industrial process control; a QN and a QN + Eligibility Trace (QN+ET). The reason for using eligibility trace is that an underfloor heating environment is dominated by slow dynamics and by using eligibility trace the agent can find correlations between the reward and actions taken in earlier iterations

Keywords: Sustainable Manufacturing Engineering and Resource-Efficient Production; Artificial Intelligence in Manufacturing; Modelling and Simulation; HVAC-Systems.

* Corresponding author. Tel.: +45 29320242
E-mail address: Cblad@m-tech.aau.dk

2351-9789 © 2019 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the Flexible Automation and Intelligent Manufacturing 2019 (FAIM 2019)
10.1016/j.promfg.2020.01.159

1. Introduction

To cope with rising energy demands and an ambition to reduce the carbon footprint from heat and energy production, regulation regarding insulation of buildings has increased. Another way to reduce energy consumption of buildings is to use more advanced controllers, which reduce energy waste and increase comfort. For large

buildings, Model Predictive Controllers (MPCs) have showed to be effective [1], but an MPC requires a full thermodynamic model of the building, which for normal households is not economically feasible to make.

A traditional controller for an underfloor heating system in a household is a hysteresis control with the room temperature as input. This controller opens and closes for the control valve supplying heat to the floor dependent on the room temperature. The core issues with using hysteresis control with the room temperature as input for controlling the room temperature is the slow thermodynamic properties of the floor, which can result in time constants between 10 minutes to 3 hours depending on the floor type and material. Because of the delayed responses in the system a hysteresis controller is not able to keep the temperature constant because of its inability to predict the energy need for the room.

This paper suggests an adaptive controller based on reinforcement learning with a neural network. Reinforcement learning is, like animal learning, based on learning by interacting with a given environment [2]. Because its learning capabilities, reinforcement learning-based control naturally adapts, to whatever environment it interacts with. Furthermore, the reinforcement learning algorithms suggested in this paper are also model-free, which, as stated earlier, is necessary for the controller to be economically feasible. In this paper two algorithms are tested in four simulation environments. Two conclusions will be derived from this; 1) by adding eligibility trace the algorithm will perform better in an environment dominated by slow dynamic, 2) by increasing the complexity of the state-action space the algorithm will become unstable and therefore limit the use to smaller state-action spaces.

2. Use case

To identify the initial problem a sketch of an underfloor heating system is shown in Fig. 1, illustrating heat fluxes in a room. In Fig. 1 the temperature of the water running through the pipes in the floor is controlled by a mixing unit. This mixing unit can be a thermostatic mixing unit or an electromechanically actuated mixing unit. A thermostatic mixing unit can be outdoor compensated, meaning it adjusts the temperature of the mixing water according to the outside temperature – high outside temperature, low mixing temperature and vice versa. An electromechanically actuated mixing unit is less common because it needs a control input, but it does allow for more control of the environment. In the work presented in this paper an electromechanical valve will be used due to its flexibility. The electromechanical valve is controlled by a step size controller. The control agent will still control the mixing temperature, but the incremental change in the temperature will be adjusted according to the distance to the given reference temperature. Meaning, if the distance between the room and reference temperature distance is high, the incremental change will be high and vice versa.

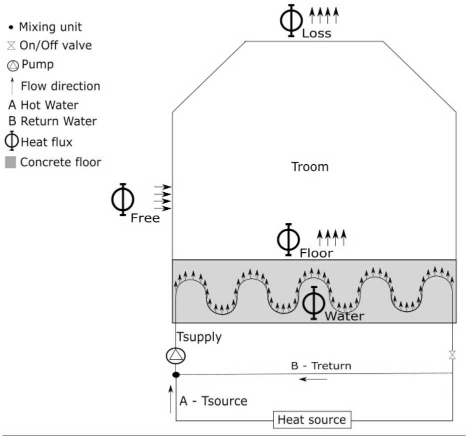


Fig. 1. An underfloor heating system with one temperature zone consisting of four heat fluxes Φ. Heat fluxes are in the simulation calculated with a 1D heat differential equation, the free heat flux Φ_{Free} is set to zero in all simulations. The hydraulic system of the underfloor floor heating is presented in the bottom of the figure, and it is assumed there is a local heat source, which also could have been a district heating source.

By using a reinforcement learning based controller it is theoretically possible for an agent to adapt to the thermodynamic properties of a given thermal zone, by making an internal predictive model and using predictive external data such as weather forecasts to enhance performance. In this paper, it is demonstrated that it is possible for the agent to learn to control a four-zoned underfloor heating system in a simulation environment in Simulink within a tolerance of 1 °C. The simulation environments have the thermal dynamic properties of an underfloor heating system, but is only affected by the ambient temperature, so no sun, wind etc. and there is no thermal transfer between interior walls.

3. Reinforcement Learning

This paper will use aspects of the deep reinforcement learning algorithm Deep Q-network (DQN), which was developed as a method to combine deep neural networks and reinforcement learning for learning directly from high-dimensional sensory inputs [3]. An underfloor heating system can in contrast to Atari games, which the DQN was developed for, be described as systems with low-dimensional sensory inputs. Therefore, the proposed algorithm utilizes a neural network with one hidden layer. The neural network with weights θ is used as a function approximator to approximate the action value function $Q(s, a) \approx Q(s, a, \theta)$. The reason for using a function approximator is because it is not computational efficient to store a large Q-table. As one might suspect using function approximators does also come with drawbacks [4]. Especially using nonlinear function approximators such as a neural network has proven hazards because of the risk of instability or divergence [5]. Using experience replay and fixed target Q in the DQN has proven that a neural network can be an efficient and stable function with improved convergence behavior [3].

Since the DQN was developed improvements has been made to the experience replay method, these include prioritized experience replay[6] and hindsight experience replay[7]. These techniques have not been considered for this paper.

Experience replay works by storing the experience from a given iteration at time *t* with the stats *s*, action *a*, reward *r* and the next state s_{t+1}, $e_t = (s_t, a_t, r_t, s_{t+1})$. The experience is stored in a memory $D[e_1, . . , . , e_t]$ and used to update the weights in the Q-network through a loss function and an optimizer. The purpose of experience replay is to reduce correlation between observation by randomly drawing experience from the matrix D, this also enables the agent to use rare experience more than ones[8]and thereby learn more efficiently from a limited amount of data. The loss function used to calculate the error between target Q and predicted Q is derived from the bellman equation and can be expressed by the following equation:

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')}\left[\left(Q_{target} - Q_{predicted}\right)^2\right] \tag{1}$$

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')}\left[\left(r + \gamma \max_{a'} Q\,(s',a';\theta_i^-) - Q(s,a;\theta_i)\right)^2\right] \tag{2}$$

An optimizer is used to update the weights in the neural network. In the original DQN is stochastic gradient descent used, in this algorithm is the Adam optimizer is used. The Adam optimizer is a first order gradient-based optimizer like stochastic gradient descent, but it also uses estimations of lower-order moments, which makes it suitable for non-stationary objectives and noisy and/or sparse gradients like a neural network can have [9].

To ensure that the agent during training explores the state action space and exploits what it has already learned, a Softmax function is used as an action selector. The Softmax function works by setting the parameter τ which indicates the agent's level of confident. τ =0 indicates full confident and to encourage more exploration τ is increased. The mathematical description of the Softmax function can be seen in the following equation: $\left(\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})\right)^{\tau}$

$$W_s\colon a \,\epsilon\, A \;\rightarrow\; \frac{e^{\left(\frac{Q(s_t,a_t)}{\tau}\right)}}{\Sigma a_{t+1} e^{\left(\frac{Q(s_t,a_{t+1})}{\tau}\right)}} \; with\, \tau > 0 \tag{3}$$

As stated in the abstract it is of interest to test if an eligibility trace implementation can perform well in an environment dominated by slow dynamics such as underfloor heating systems. Eligibility trace is a method that makes it possible to make a trade-off between Monte Carlo and Temporal Difference, where Monte Carlo has high variance because Monte Carlo only updates at the end of the episode. This MDP is considered continues as it does not have episodes, so Monte Carlo cannot be used. Temporal Difference on the other hand updates for every iteration but uses its own estimation to update, which means it has bias [2]. Eligibility trace or n-step learning uses a parameter n, where n is the number of iterations that will pass before an update is made. This means if n is the same size as the number of iterations in an episode it is Monte Carlo.

For experience replay to be compatible with eligibility trace, a few modifications has been made to way the data is drawn from the experience memory D. Instead of drawing random samples, the agent is drawing random batches of the same size as the eligibility trace

4. Experiment

The structure of the used Q-network with input states, Q-values, and the action selector, is illustrated in Fig. 2.

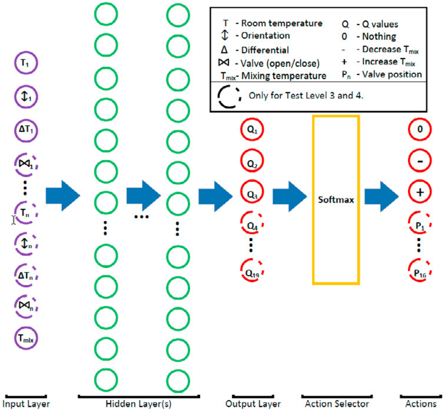


Fig. 2. Illustration of the neural network, with input states, Room temperature, orientation to reference temperature, differential of room temperature, valve position, and mixing temperature. And output Q values and the Softmax action selector. Two hidden layers is shown in the illustration, this is just to illustrate that it is possible to add more layers.

The experiments have been divided into four different test levels (TL's) to sort out which RL algorithm that is suited for the task. Both algorithms will start in the environment of TL1 and then continue to TL4 increasing the complexity where they need to satisfy the requirement (R): Room temperature(s) is allowed a standard deviation of 1 °C from the room reference temperature 22 °C. The TL1 environment consist of one temperature zone, but with no thermo-properties of an underfloor heating system. The only task of the agent is to control the mixing temperature. The environment of the TL2 is still one temperature zone but with the thermo-properties of an underfloor heating system has been added in this test level. By doing this it will be possible to investigate the effect of eligibility trace in a dynamic environment. TL3 and TL4 consist of multiple zones meaning the agent can control valves and the mixing temperature. TL3 has two zones and TL4 has four zones. The ambient temperature in the simulation environment is set to be constant in the beginning of the simulation and then a 1-day and a 14-days sine cycle is added to represent day and night changes and longer changes over 14 days. Hyperparameters and settings of the algorithms are shown in Appendix A. The following results from the test levels will consist of response temperature plot of the environment from 232 days' time per test in TL1 and TL2 and 926 days' time in TL3 and TL4.

4.1. Test Level 1 Results

Two tests were performed in TL1, one test of the QN and one test of the QN+ET, the results can be seen in Fig. 3.

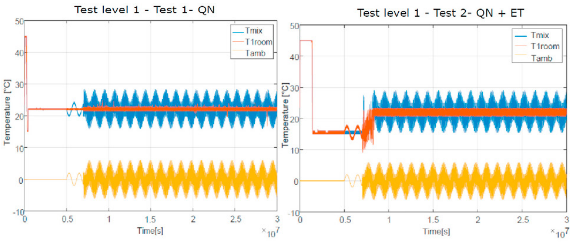


Fig. 3. Results of Test Level 1 with no thermo-properties of an underfloor heating system has been introduced. The mixing temperature T_{mix} is blue, room temperature $T1_{room}$ is red and ambient temperature T_{amb} is yellow.

It can be seen from the results of the two tests in Fig. 3 that the QN algorithm without eligibility trace does perform equally good or better than the algorithm with eligibility trace. The reason for this is that there are no slow responses in this system, because there is no reason that eligibility trace should improve performance.

4.2. Test Level 2 Results

Two tests are performed in TL2, where Test 1 is the QN algorithm and test 2 is the QN algorithm with eligibility trace, the results of the two tests are shown in Fig. 4.

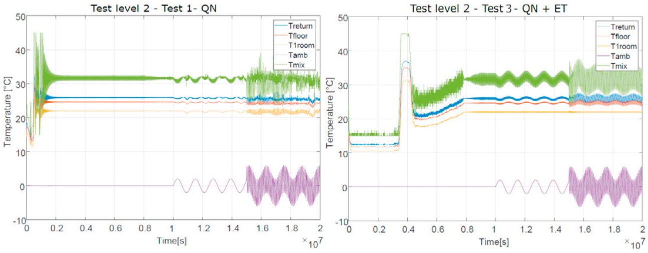


Fig. 4. Results of Test Level 2 with thermo-properties of an underfloor heating system has been introduced. The mixing temperature T_{mix} is green, ambient temperature T_{amb} is purple, return temperature T_{return} is blue, floor temperature T_{floor} is red and room temperature $T1_{room}$ is yellow.

Table 1. Satisfaction of requirement R from the period $1.8 \cdot 10^7$ to $2 \cdot 10^7$ seconds in Test Level 2 are grey scaled elements with tests where R is satisfied.

Test	Model type	$[\bar{x}, \sigma_x]$
1	QN	[21.52,0.54]
2	QN	[22.44,0.54]
3	QN+ET	[22.04,0.11]
4	QN+ET	[22.09,0.47]

From the results in Table 1 and Fig. 4 it can be seen QN+ET performs best, and that the QN algorithm does not meet the requirements in the first test but manages to do so in second test. A comparison to TL1 without slow dynamics reveals that the QN algorithm performed slightly better than QN+ET. This comparison leads to the conclusion that ET does improved performance when the system is dominated by slow dynamics.

4.3. Test Level 3 Results

TL3 consists of four tests; two with the QN algorithm and two with the QN+ET algorithm. The results of two successful tests are shown in Fig. 5.

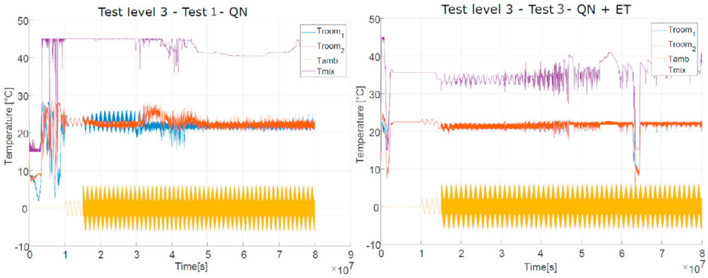


Fig. 5. Results of Test Level 3 with thermo-properties of an underfloor heating system has been introduced. On the left Test 1 and Test 3. Where T_{room} 1 is blue, T_{room} 2 is red, T_{mix} is purple and the ambient temperature T_{amb} is yellow

From Table 4 it is seen that the QN+ET algorithm satisfies the requirement two times in both rooms, where the QN algorithm only satisfied the requirements for one of the two tests and it did not perform as well in this test regarding standard deviation or mean value. Note that Test 2 with QN+ET also manages to have the lowest energy usage due to average lowest mixing temperature T_{mix}.

Table 2. Satisfaction of requirement R from the period $7{\cdot}10^7$ to $8{\cdot}10^7$ seconds in Test Level 3 with QN and QN+ET where grey scaled elements are tests where R is satisfied.

Test	Model type	$T1_{room}[\bar{x}, \sigma_x]$	$T2_{room}[\bar{x}, \sigma_x]$	$T_{mix}[\bar{x}]$
1	QN	[20.12,2.11]	[21.81,1.67]	[40.47]
2	QN	[22.38,0.62]	[22.35,0.63]	[38.74]
3	QN+ET	[21.91,0.34]	[21.87,036]	[36.61]
4	QN+ET	[22.03,0.58]	[22.19,0.55]	[42.73]

4.4. Test Level 4 Results

TL4 consists of three tests with the QN+ET algorithm due to it satisfied the requirement in TL3 and it was not possible to perform a satisfied test of the QN algorithm. The result of TL4 is shown in Fig. 6.

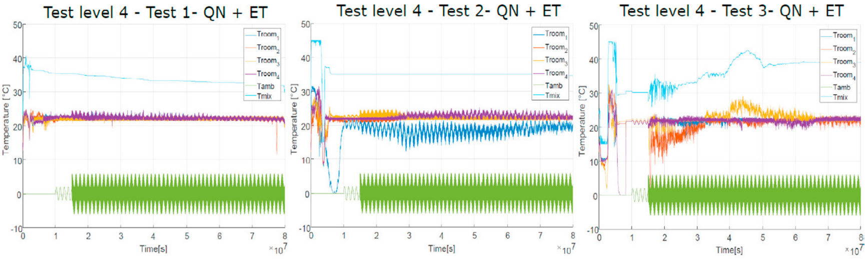


Fig. 6. Results from three tests from Test Level 3 of QN+ET.

From Table 4 it is observed that Test 3 is successful, it did however require three tests, which shows that the algorithm has become unstable due to the complexity of the state-action space.

Table 3. Satisfaction of requirement R from the period $7{\cdot}10^7$ to $8{\cdot}10^7$ seconds in Test Level 4 with QN+ET where grey scaled elements are tests where R is satisfied.

Test	Model type	$T1_{room}$	$T2_{room}$	$T3_{room}$	$T4_{room}$	T_{mix}
		$[\bar{x}, \sigma_x]$	$[\bar{x}, \sigma_x]$	$[\bar{x}, \sigma_x]$	$[\bar{x}, \sigma_x]$	$[\bar{x}]$
1	QN+ET	[21.89,0.36]	[21.77,1.13]	[21.99,0.37]	[21.99,0.38]	[32.13]
2	QN+ET	[19.69,0.85]	[22.07,0.27]	[22.2,0.25]	[22.63,0.6]	[34.73]
3	QN+ET	[22.08,0.26]	[21.95,0.61]	[22.2,0.28]	[22.35,0.34]	[39.05]

5. Conclusion

By reviewing the results of the 9 tests it can be concluded that it is possible for a reinforcement learning based controller to control the designed simulation environment of an underfloor heating system. In this study two different algorithms have been tested; 1) QN and 2) QN +ET. By comparing the performance of the two algorithms it can be concluded that the eligibility trace addition to the Q-network does increase performance slightly, but only when there are slow dynamics included in the simulation. Furthermore, it is concluded that the function approximator does become unstable when increasing the complexity of the state-action space. This means that one should be aware of the size of the state-action space when using this technique and additional research is needed to make the proposed reinforcement learning approach more robust.

6. Future Work

As stated in the introduction, the simulation environment is simplified i.e. it does not include interior wall transfer, windows, sun, wind etc. A more detailed simulation environment would make it possible to take all these parameters into account. To utilize the full potential of reinforcement learning it would be ideal to use forecasted weather data. This way the agent would not only depend on its internal model of the dynamic behavior, especially in an environment dominated by long delayed response this would be desirable. In this study the simulation environment has been defined as one single MDP, which can include 1, 2 and 4 temperature zones. It has been observed that by increasing the number of zones in the MDP the training time becomes longer and the agent does perform less desirable. This, of cause, makes sense, because the complexity of the state-action space is increased with the number of zones. It would be interesting to explore the possibility to design an agent with multiple MDPs, or a multi-agent controller, to control the temperature zones independently. All the above observations are subjects for future works.

The proposed reinforcement learning control is not robust enough to use in commercial applications yet. Additional research must be made into increasing the robustness of the controller. Multiple improvements have been made to the DQN algorithm, the latest is the Rainbow algorithm [10]. These improvements might also be better suited for the current and future tasks at hand.

References

[1] SamuelPrívara and Jan Širokýb and Lukáš Ferkla and Jiří Ciglera *Model predictive control of a building heating system: The first experience*. Energy and Buildings Volume 43, Issues 2–3, Pages 564-572, February–March 2011

[2] Richard S. Sutton and Andrew G. Barto *Reinforcement Learning: An Introduction*. 2. Edition. 2018.

[3] Volodymyr Mnih and Koray Kavukcuoglu and David Silver and Andrei A. Rusu and Joel Veness and Marc G. Bellemare and Alex Graves and Martin Riedmiller and Andreas K. Fidjeland and Georg Ostrovskivand Stig *Human-level control through deep reinforcement learning*. Nature , 529–533.

[4] Sebastian Thrun and Anton Schwartz *Issues in Using Function Approximation for Reinforcement Learning*. Proceedings of the Fourth Connectionist Models Summer School Lawrence Erlbaum Publisher, Hillsdale, NJ, Dec. 1993

[5] John N. Tsitsiklis and Benjamin Van Roy. *An Analysis of Temporal-Difference Learning. TRANSACTIONS ON AUTOMATIC CONTROL*. Transactions on automatuc control VOL. 42, NO. 5, May 1997.

[6] Tom Schaul, John Quan, Ioannis Antonoglou, David Silver. *Prioritized Experience Replay*. International Conference for Learning Representations, 2016.

[7] Andrychowicz, Marcin and Wolski, Filip and Ray, Alex and Schneider, Jonas and Fong, Rachel and Welinder, Pe ter and McGrew, Bob and Tobin, Josh and Pieter Abbeel, OpenAI and Zaremba, Wojciech *Hindsight Experience Replay*. Advances in Neural Information Processing Systems 30, pp. 5048–5058, 2017.

[8] Long-Ji Lin. *Self-improving reactive agents based on reinforcement learning, planning and teaching*. Machine Learning, Volume 8, Issue 3–4, pp 293–321 May 1992,

[9] Diederik P. Kingma, Jimmy Lei Ba. *ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION*. International Conference for Learning Representations, 2015.

[10] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, David Silver *Rainbow: Combining Improvements in Deep Reinforcement Learning*. The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18).

Appendix A. Settings of Algorithms for Test Levels

Table A1. Standardization of input variables.

Input variables	Standardisation Equations
T_{std}	$\frac{T}{35}$
T_{std}	$\lvert (T-T_{last})\cdot 10) \rvert$
T_{std}	$\frac{0.5\ if\ T_{room}\ \geq\ T_{ref}}{0.5\ if\ T_{room}\ <\ T_{ref}}$
T_{std}	$\frac{1\ if\ Q\ >\ 0}{0\ if\ Q\ =\ 0}$
T_{std}	T_{std}

Table A2. Setup for Q-networks for test levels.

Q-network Setup	Test Level			
	1	2	3	4
Input variables	4	4	9	17
Hidden layers	1	1	1	1
Hidden neurons per layer	30	30	30	30
Output variables	3	3	7	19

Table A3. Hyperparameters for algorithms in test levels.

Hyperparameters	Algorithm	
	QN	QN+ET
Learning rate T_{std}	0.001	0.001
Discount factor γ	0.9	0.9
Softmax temperature τ	100	100
Experience replay batch size	50	50
Experience replay capacity	100000	100000
Eligibility trace steps n	-	30

Paper B

PaperB

Paper C

PaperC

energies

MDPI

Article

A Multi-Agent Reinforcement Learning Approach to Price and Comfort Optimization in HVAC-Systems

Christian Blad [1,2,*,†,‡], Simon Bøgh [1,‡] and Carsten Kallesøe [2,3,‡]

[1] Robotics & Automation Group, Department of Materials and Production, Aalborg University, 9220 Aalborg, Denmark; sb@mp.aau.dk
[2] Technology and Innovation, Control Department, Grundfos, 8850 Bjerringbro, Denmark; csk@es.aau.dk
[3] Department of Electronic Systems, Aalborg University, 9220 Aalborg, Denmark
[*] Correspondence: cblad@m-tech.aau.dk
[†] Current address: Fibigerstræde 16, 9220 Aalborg, Denmark.
[‡] These authors contributed equally to this work.

Abstract: This paper addresses the challenge of minimizing training time for the control of Heating, Ventilation, and Air-conditioning (HVAC) systems with online Reinforcement Learning (RL). This is done by developing a novel approach to Multi-Agent Reinforcement Learning (MARL) to HVAC systems. In this paper, the environment formed by the HVAC system is formulated as a Markov Game (MG) in a general sum setting. The MARL algorithm is designed in a decentralized structure, where only relevant states are shared between agents, and actions are shared in a sequence, which are sensible from a system's point of view. The simulation environment is a domestic house located in Denmark and designed to resemble an average house. The heat source in the house is an air-to-water heat pump, and the HVAC system is an Underfloor Heating system (UFH). The house is subjected to weather changes from a data set collected in Copenhagen in 2006, spanning the entire year except for June, July, and August, where heat is not required. It is shown that: (1) When comparing Single Agent Reinforcement Learning (SARL) and MARL, training time can be reduced by 70% for a four temperature-zone UFH system, (2) the agent can learn and generalize over seasons, (3) the cost of heating can be reduced by 19% or the equivalent to 750 kWh of electric energy per year for an average Danish domestic house compared to a traditional control method, and (4) oscillations in the room temperature can be reduced by 40% when comparing the RL control methods with a traditional control method.

Keywords: deep reinforcement learning; artificial intelligence; HVAC-systems; underfloor heating; energy in buildings; predictive analytics

check for updates

Citation: Blad, C.; Bøgh, S.; Kallesøe, C. A Multi-Agent Reinforcement Learning Approach to Price and Comfort Optimization in HVAC-Systems. *Energies* **2021**, *14*, 7491. https://doi.org/10.3390/en14227491

Academic Editor: Adel Merabet

Received: 30 September 2021
Accepted: 3 November 2021
Published: 9 November 2021

1. Introduction

In the USA and Europe, roughly 35% of the energy consumption in 2008 was used in HVAC systems [1]. To reduce energy consumption and hence the carbon footprint from heat and energy production for HVAC systems, the regulation regarding insulation of buildings has increased [2]. Another way to reduce energy consumption in buildings is to use more advanced control techniques, reducing energy waste, and increasing comfort. For large buildings, Model Predictive Controllers (MPCs) have shown to be effective [3,4], but MPCs require a full thermodynamic model of the building, which is not economically feasible for regular households. Smart controllers based on scheduling according to energy prices are proposed [5,6]. These algorithms require less commissioning than an MPC's but are still comprehensive to commission. To handle the issues with commissioning and still harvest the benefits with advanced control techniques, model-free Reinforcement Learning (RL) is proposed for the control.

This article focuses specifically on control of Underfloor Heating systems (UFH) in domestic houses. Traditionally, hysteresis control with room temperature as input is used for controlling these UFHs. This hysteresis controller fully opens or closes the control valve supplying heat to the floor dependent on the room temperature. The main issue with using hysteresis control in UFH is the slow thermodynamic properties of the system, which can lead to time constants between 10 min to 3 h depending on the floor type and material [7]. Due to the slow responses in the system, the hysteresis controller is not able to keep the temperature constant because of its inability to predict the energy demand for the rooms. The temperature of the supply water is traditionally controlled by an *ambient temperature compensated controller*. This type of controller is, as the hysteresis controller, also affected by the slow response in the convection of heat from the water to the room, but also the delayed response associated with transporting the water in the pipes.

Reinforcement Learning (RL) is a model-free adaptive control method, and as such, it is possible to adapt to the specific dynamic properties of the environment [8]. This capability makes RL particularly interesting for the control of UFH or Heating, Ventilation, and Air-conditioning (HVAC) systems in general [9]. In particular, the commissioning phase is automated with RL. Moreover, the user behavior has a large effect on the comfort level of the temperature zones. Here, RL is able to take this behavior into account in the control as well [10]. In this paper, user behavior is however not included, because the goal is to investigate how the RL algorithms will adapt to the building environments. The user behavior will just complicate this analysis.

The RL algorithm studied in this article is an online learning method. Therefore, the agent/agents of the RL will not perform optimally during training. The training time is highly correlated with the complexity of the state-action space [8]. This correlation is an issue in the RL control design for UFH, as it makes it difficult to scale the RL algorithm to houses with multiple temperature zones [11]. To illustrate the scaling problem of having a single agent to control the entire system of a one, two, three, and four-zone UFH system, the calculations of the action-state space have been made for the four cases, see Table 1.

Table 1. Size of action-state space for a one, two, three and four-zone UFH system. The action-state space grows exponentially with the number of temperature zones. Assumptions about the number of discrete values each action or state has: T_{supply}; 15, Valve per zone; 2, T_{room} per zone; 12, T_{amb}; 30, heat consumption; 10, sun; 10.

	Action-State Space Size
One zone	1×10^6
Two zones	26×10^6
Three zones	622×10^6
Four zones	15×10^9

Table 1 shows that the action-state space grows exponentially with the number of temperature zones in the system, which means that RL control does not scale well in the UFH control task or many other control problems [8]. To deal with this scaling problem, we propose to incorporate Multi-Agent Reinforcement Learning (MARL). Instead of formulating the problem as a *Markov Decision Process* (*MDP*) and use Single Agent Reinforcement Learning (SARL), the problem is formulated as a *Markov Game* (*MG*), and MARL can be used, see Figure 1 for an illustration of an MDP and an MG.

From Figure 1, it is seen that the interaction with the environment changes, but not the environment itself. Whereas the environment in a single agent system receives one action, in a multi-agent system it receives an action vector with the same size as the number of agents in the system. The states and rewards received by the agents from the environment are distributed, such that it is possible to only pass relevant state information to a given agent. Formulating an environment as an MG to use MARL has been used in other applications as well. In [12], a voltage control for a power grid [12] is designed, and MARL is applied with success for route planning in road network environments in [13]. MARL is also used for training unmanned fighter aircrafts in air-to-air combat in [14].

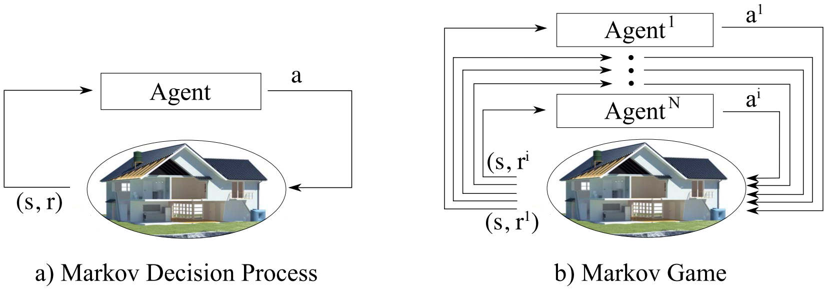


Figure 1. Illustration of the difference between a Markov Decision Process and a Markov Game. In (**a**) one agent and one environment are seen. The agent interacts with the environment by sending a tubule of actions and receiving one reward and the states of the environment. In (**b**) multiple agents and one environment are seen. The action space is split into *i* number of actions. Each agent receives a reward and the states of the environment.

RL for HVAC systems has previously been studied. In [15], Q-learning is used for a single thermostat. Here 10% energy saving is achieved by scheduling the reference temperature, such that comfort is only considered relevant if there are occupants in the room. RL is used in [16] for controlling the supply temperature and the pressure in a mixing loop. Though the mixing loop only represents a small part of the entire HVAC system, it shows that RL can outperform state-of-the-art industrial controllers. In [17], RL is used to control airflow rates for up to five zones, where each zone has an individual actuator. It is found that it is possible to reduce energy cost, but training times increased drastically when going from one to four zones. This result supports the need for an MG formulation, which makes it possible to use MARL in an HVAC system.

To the authors' knowledge, the results presented in this paper are the first to introduce MARL for the control of UFH systems. Prior papers on MARL in other parts of HVAC systems do exist, though primarily for HVAC systems for commercial buildings. In [10], an air handling unit controlled by a MARL algorithm formulated as a Markov Game in a cooperative setting is presented. This formulation means that each agent is aware of the state of the other agents, but not the current action taken by other agents. The algorithm is based on an actor-critic model. In that paper, a 5.6% energy saving is achieved. Looking slightly beyond HVAC systems, MARL is used in hot water production for domestic houses in [18]. Here, MARL is used in a distributed setting where agents are not aware of other agents' actions.

The paper is organized as follows. First, Section 2 presents the background and contributions of the work. The general SARL and MARL theory is explained, and which methods inspired the present work, finally elaborating on the contributions to the research field. Then, Section 3 presents the system design and an evaluation of the designed Dymola multi-physics simulation. This simulation serves as the training and test environment for the designed SARL and MARL algorithms. Next, Section 4 presents the underlying math and design of the RL systems, along with hyperparameters, input states, and reward functions. Finally, Section 5 presents the experiments and the analysis of experimental results, and Section 7 concludes the paper.

2. Background and Contributions

This section introduces the theory behind SARL and MARL, and argues why MARL is relevant in the control of UFH. Furthermore, the contributions of this paper are explained.

2.1. Reinforcement Learning

RL can be divided into three categories: Value-based learning, policy-based learning, and actor-critic-based learning, where actor-critic is a combination of value-based and policy-based learning. In this paper, we focus on value-based learning, specifically Q-learning as a backbone technology, but the benefits of using MARL transfer across all three types [19]. The central idea of value-based learning is to find the optimal action-value function, which needs to satisfy the Bellman optimality equation (Equation (1)) [8]. Let $Q^*(s,a)$ be the optimal value-action function, then $Q^*(s,a)$ is given as follows:

$$Q^*(s,a) = \mathbb{E}[R_{t+1} + \gamma \max_{a'} Q^*(s^{'}, a^{'}) \mid s, a]. \tag{1}$$

The bellman equation states that if the future state s' for all actions is known, then the optimal policy is to choose the next action a' that results in the highest Q value ($Q^*(s,a)$). This approach for choosing a is referred to as a greedy policy.

An RL algorithm learns about the environment it interacts with by iteratively updating its estimate of the action-value function, such that the action-value function converges towards $Q_i \to Q^*$ for $i \to \infty$.

Choosing the correct action if the policy is greedy is simple. Updating the Q-function so that Q_i converges towards Q_* is, however, difficult, at least within a reasonable number of iterations. The update strategy for Q-learning without so-called function approximators is given by Equation (2).

For the update strategy to converge to Q^*, it is necessary for the environment to satisfy the conditions of a Markov Decision Process (MDP).

Definition 1. *A MDP is defined by a tuple $\{S, A, P, R\}$; S is the finite number of states, A is a finite number of actions, and P is the transition probability for s_t to transition to s_{t+1} under a given action a. R is the immediate reward for the expected transition from s_t to s_{t+1}.*

$$Q^{update}(s,a) = Q^{current}(s,a) + \alpha \cdot (r_t + \gamma \max_{a'} Q(s^{'}, a) - Q^{current}(s,a)). \tag{2}$$

Equation (2) shows that for the Q-function to converge to Q^*, it is necessary to visit all state-action pairs. This is simply not feasible in large systems. Therefore, function approximation is used to approximate the Q function $Q(s,a;\theta) \approx Q(s,a)$. By using an Artificial Neural Network (ANN) as a function approximator, it is shown that a Q-function can converge [20]. Additionally, for function approximation with ANNs, a range of methods have been developed to reduce convergence time or improving convergence in value-based RL, for example double Q-learning [21], experience replay, prioritized experience replay [22], and several other methods.

Even though the above-mentioned methods do improve generalization and reduce training, RL, and machine learning, in general, do suffer from what Richard E. Bellman refers to as *"the curse of dimensionality"* [23]. For this reason, MARL is explored for RL control of HVAC systems.

2.2. Multi-Agent Reinforcement Learning

MARL is like SARL, concerned with solving decision-making problems. However, instead of one agent deciding all actions in a system based on one reward, multiple agents decide the actions and receive individual rewards or a joint reward dependent on the type of MARL setting.

This paper focuses on MARL systems formulated as an MG. The formal definition of an MG is as follows:

Definition 2. *A Markov Game is defined as a discrete time-stochastic process, a tuple $\langle N, S, {A^i}_{i\epsilon N}, {R^i}_{i\epsilon N}, P \rangle$ where N is the number of agents, S is the state space observed by all agents, and ${A^i}_{i\epsilon N}$ is the joint action space of all N agents. $R^i : S \times A \times S \to \mathbb{R}$ is the immediate reward received by agent i for transitioning from (s, A^i) to s', and P is the transition probability [24]. The definition of an MG can be interpreted as the following: At time T every i'th agent $i = [1 \ldots N]$ determines an action A^i according to the current state s. The system changes to state s' with probability P and each agent receives an individual reward R^i based on the new state s'.*

When going from SARL to MARL, an entirely new dimension is added to the problem. It is, therefore, necessary to define what type of MG the system is. The type describes how the agents are formulated and affect each other. A MARL problem can be formulated in three ways: (1) A cooperative setting, (2) a competitive setting, and (3) a mixed setting [25].

Cooperative, Competitive, and Mixed Setting

In general, a MARL algorithm in a cooperative setting is formulated with a common reward function so $R^1(s, a, s') = R^2(s, a_2, s') = R^n(s, a_n, s')$ and referred to as a Markov team game. A number of different algorithms exist for solving a Markov team game, these include *team-Q* [26] and *Distributed-Q* [27]. Distributed-Q is a MARL algorithm framework, which is proven to work for deterministic environments. It has been shown that all agents will converge to an optimal policy without sharing data between agents. This is very appealing. However, because this article is concerned with a general sum problem, the approach will not work.

A MARL algorithm in a competitive setting is formulated as a zero-sum game where one agent's win over the other agents. Such a setting can be formulated as $\sum_i R^i(s, a_i, s') = 0$. There exists a number of zero-sum game algorithms, see for example *minimax-Q* [28]. This MARL algorithm setup is however of little interest for this paper, as rewards for a UFH system cannot be formulated as a zero-sum game.

Finally, in the mixed setting each agent has its own reward function, and therefore its own objective. This mixed setting is also referred to as a general sum game.

Game theory and the Nash equilibrium play an essential role in the analysis of these systems. In contrast to a cooperative setting, where it is assumed that the system's overall best reward can be found by having all agents maximize their own reward, this is not possible in a general sum setting. A number of different algorithms for general sum games have been designed for static tasks [29]. However, UHF is a dynamics system meaning that static tasks must be adapted to dynamic systems to be interesting for our work.

In addition, single-agent algorithms are used in a mixed setting [30,31]. That is, even though there is no grantee of conversion if applying SARL to a multi-agent system [24]. Algorithms, which are designed for dynamic tasks, include "*Nash-Q*" [32] and "*PD-WoLF*" [33], and will form the starting point for our work.

The idea of a Q-learning algorithm that finds a Nash equilibrium is compelling. Succeeding papers have however argued that the application of Nash-Q is limited to environments that have a unique Nash equilibrium for each iteration [25]. More recent work on fully decentralized MARL has been proven to converge under the assumption of using linear function approximators for the value function [34]. Even though this algorithm is distributed, a joint Q function is incorporated, which makes all agents aware of each other. This is necessary to prove general convergence, but it also increases complexity as the number of agents grows and therefore makes the approach less scalable. In a more recent work, agents are distributed in a similar manner to our work [35]. The main difference between their architecture and our framework is that our agents only observe parts of the state space. Moreover, different methods are utilized for updating the Q-function.

2.3. Contributions

This paper extends the current state-of-the-art for model-free control of UFH systems, including testing SARL on UFH systems and presenting a novel MARL approach to HVAC systems. The novelty lies in the interaction between agents in the MARL algorithm. In distributed Q and Nash-Q, agents are either not aware or completely aware of each other. In this paper, each agent acts according to a well-defined structure as described in Section 4. Furthermore, the comparison between the SARL simulation and MARL simulation validates the hypothesis that MARL can reduce training times in HVAC systems. Lastly, we present a novel method to ensure robustness for controlling the supply temperature in HVAC systems.

3. System Design and Evaluation

To test hyperparameters, input states, and algorithms, it is necessary to have a simulation environment. A simulation environment is never a 1:1 representation of the real world, but for a simulation environment to be applicable in this study, it is necessary that the simulation, to a large extent, has the same dynamic behavior. To accomplish this, Dymola has been used as the simulation tool.

3.1. Dymola Multi-Physics Environment

Dymola is a Modelica-based multi-physics simulation software and, as such, is suitable for simulating complex systems and processes. Several libraries have been developed for Dymola. For the simulations in this paper, the standard Modelica library and the Modelica Buildings library are used.

The simulation can be split into two parts: (1) The hydraulic part and (2) the thermodynamic part. The hydraulic part of the simulation can be described by a mixing loop, a pump, one valve per temperature zone, and some length of pipe per temperature zone.

The thermodynamic side of the simulation is constructed using the base element "*ReducedOrder.RC.TwoElements*". This element includes heat transfer from exterior walls, windows, and interior walls to the room. It furthermore includes radiation from the outside temperature and radiation from the sun. This means that wind and rain do not affect the simulation, as they are assessed to be smaller disturbances. These disturbances are not negligible, but for the purpose of this paper, the simulation results will still indicate the saving potential that can be expected in real-life installation. The element is made in accordance with "VDI 6007 Part 1", which is the European standard for calculating the transient thermal response of rooms and buildings [36].

The length of pipe used in each zone and parameters for the windows, walls, zone area, and volume are shown in Table 2.

Table 2. Parameters used for each temperature zone. A and B refer to if it is the one-zone simulation or the four-zone simulation.

Parameter	Zone1A	Zone1B	Zone2B	Zone3B	Zone4B
Length of pipe	105 m	56 m	105 m	42 m	70 m
Window area	20 m^2	12 m^2	25 m^2	12 m^2	24 m^2
Wall area	39 m^2	36 m^2	40 m^2	12 m^2	30 m^2
Zone area	30 m^2	16 m^2	30 m^2	12 m^2	20 m^2
Zone volume	80 m^3	48 m^3	90 m^3	36 m^3	60 m^3

To simulate how the room receives heat from the floor, a floor element has been constructed. The floor element incorporates the pipe length, pipe diameter, floor thickness, floor area, and construction material. These parameters enable the floor element to simulate how the heat from the water running in the pipes will transfer through the concrete and into the room. The heat in the room is assumed to be uniformly distributed. This means that the temperature at the floor, at walls and at the sealing of the room is the same. Modeling the temperature distribution uniformly is also in accordance with "VDI 6007part1".

3.2. Evaluation of Simulation

It is not possible to validate the simulation environment with data from a real-world system. We can, however, evaluate step responses to evaluate the dynamic convection of heat from the water in the pipes to the air in the room. Additionally, we can evaluate the amount of power the rooms require and compare it to a real-world house. Lastly, the daily and seasonal power consumption can be evaluated.

To evaluate the simulation environment, a run of the simulation with hysteresis control on the valves and an outdoor compensated supply temperature is executed. Note, that hysteresis control is the control method traditionally used in the UFH system. For the validation, a simulation with a one-temperature zone system is used. However, a similar simulation has been made for a four-temperature zone system with similar results.

All simulations are made with a hysteresis control with reference point 22 °C and a dead band of ±0.1 °C. The outside compensated supply temperature follows a linear model, see Equation (3).

$$T_{supply} = -0.6 \cdot T_{ambient} + 42. \tag{3}$$

Firstly, the room temperature of an entire heat season is plotted in Figure 2.

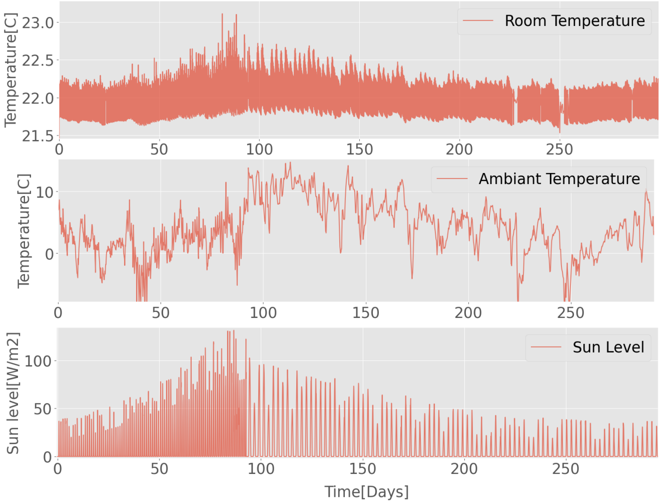


Figure 2. Simulation results of one heating season, with a traditional controller and outside compensated supply temperature.

From Figure 2, it can be seen that a heating season is approximately 280 days. The heating season is defined here as the period of the year where the building needs energy to sustain a zone temperature of 22 °C. The simulation starts 1 March, and the period from 1 June to 1 September has been removed from the weather file as no heat is needed in this period. The seasonal effect can be seen in the figure, where occasional overshoots happen in the period from day 70 to day 140. Hence, in the fall/spring period, where heat is needed during the night and morning but not during the daytime, overshoots happen. The temperature is otherwise oscillating between 21.7 °C and 22.2 °C.

To investigate the response more closely, the room temperature and the associated valve position is plotted over a period of 2 h and 30 min, see Figure 3.

Figure 3 shows that when a valve is opened, 1700 s (0.02 days) will pass before the temperature gradient in the room becomes positive. Additionally, it can be seen that when the valve is closed, the temperature will continue to rise an additional 0.1 °C and another about 1700 s will pass before the gradient becomes negative. This behavior is due to the slow dynamic properties that are expected of a UFH system, and therefore it also validates that this simulation resembles the typical dynamics of a UFH system.

The price of heating over one heating season is plotted in Figure 5. Before reviewing the plot, it is necessary to explain how this price is calculated. The price will also serve as a benchmark to prove that significant cost savings are possible by utilizing reinforcement learning in UFH. To that end, in this article, it is assumed that the heat supply is an air-to-water heat pump.

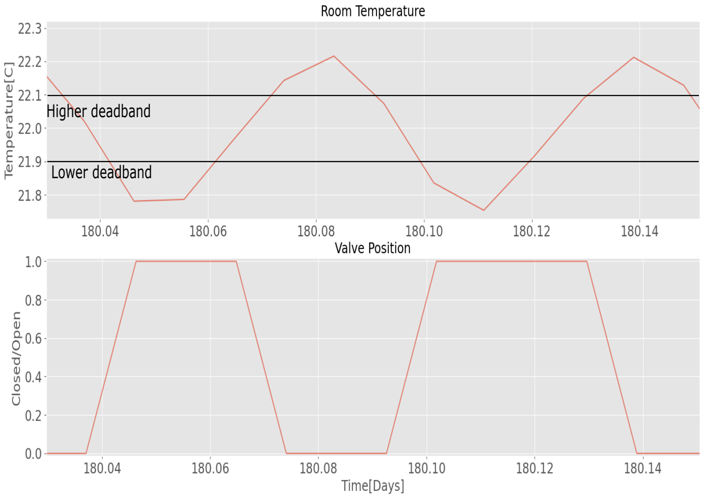


Figure 3. Room temperature and associated valve position over a period of 8640 s (0.1 days). By investigating the graph, the dynamic response can be analyzed.

The price of heating with a heat pump can be simulated by knowing the cost of electricity, the dynamics of a heat pump, and the power consumption of the system. The cost of electricity is assumed to fluctuate during the day. The average Danish price of electricity during a day can be seen in Figure 4a [37]. The dynamics of a heat pump can be described with the Coefficient of Performance (COP), which is a function of the ambient temperature and the supply temperature. This COP can be seen in Figure 4b [38]. Additionally, it is necessary to describe the Part Load Factor (PLF), which describes how the efficiency of the heat pump depends on the duty cycle. This PLF is shown in Figure 4c [39].

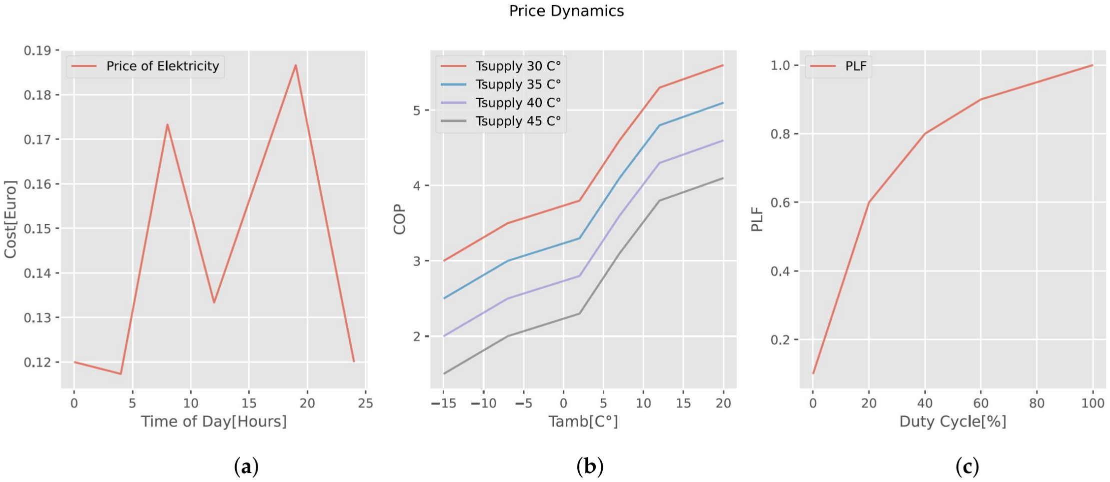


Figure 4. Dynamics of a heat pump: (**a**) Shows the average electricity prices, including taxes in Denmark as a function of the time of day (tod). (**b**) Shows the Coefficient of Performance(COP) as a function of the ambient temperature, for four different supply temperatures. (**c**) Shows the Partial Load Factor(PLF) as a function of the duty cycle (D).

With the Cost of Electricity (CE), the COP and the PLF described and the power consumption of the system (ΔE) available from the simulation, the cost of heating with a heat pump can be simulated with Equation (4):

$$cost = \frac{\Delta E}{COP(T_{amb}, T_{supply}) \cdot PLF(D)} \cdot CE(tod). \quad (4)$$

From Figure 5, it is evident that the price of heating over one heating season varies. The lowest cost is found in the spring/autumn period and highest during winter. Though there is a yearly trend, it is also evident that the price during a 14-day period can vary 30%, as seen from day 200 to 210.

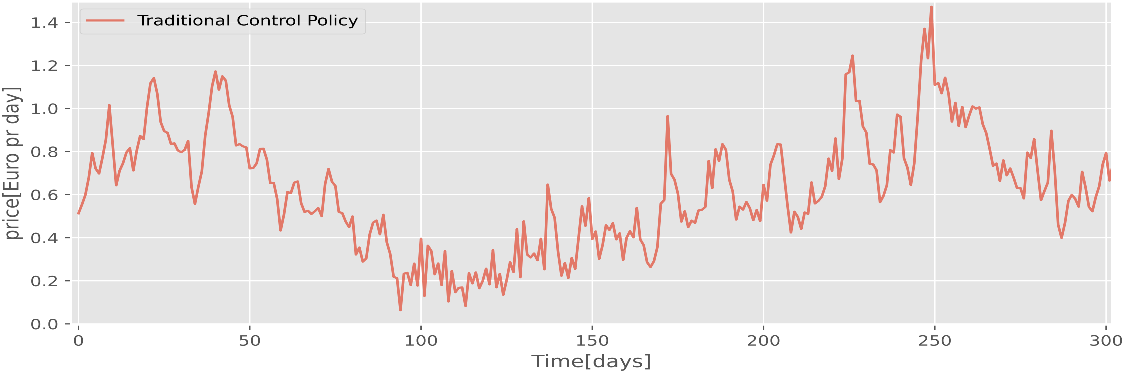


Figure 5. Price of heating over one heating season for a one-temperature zone system, of 30 m^2. The price is price per day in Euro.

Finally, the power consumption is reviewed. The temperature zone is 30 m^2 and consumes 3561 kWh over a heating season at a reference temperature of 22 °C, meaning an average of 118 kWh per m^2. An average Danish house uses 115 to 130 kWh per m^2 [40], which shows that the simulation is within what is considered average in a Danish climate.

The use case is now described, the simulation has been evaluated, and it has been shown that it resembles a traditional Danish house. The RL algorithm can therefore be designed and tested on this simulation.

4. RL Algorithm Design

Figure 6 illustrates the hydraulic system for a four-zone UFH system. The system in the figure consists of a heat pump with a supply temperature and four on/off valves controlling the temperature of each of the four zones. The MARL algorithm is designed as an MG in a general sum setting as explained in Section 2.2. By reviewing Section 3, it can be seen that the natural way of dividing the UFH environment into multiple agents is achieved by having one agent control the supply temperature and one agent for each of the temperature zones. Each of the temperature zone agents will, in this setting, control the on/off valves supplying the zones with hot water. This setup means that for a four-temperature zone UFH system, there are five agents, as illustrated in Figure 6.

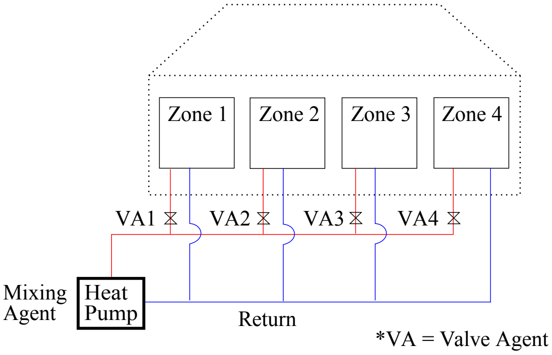


Figure 6. Four-zone temperature system, with a air to water heat pump. From the illustration, it can be seen how the agents are divided into one mixing agent and four valve agents. The mixing agent controls the supply temperature and the valve agents control the flow to each zone.

By splitting the environment into five agents instead of one, the action space is changed. The result of this change is shown in Table 3. From the table, it is seen that the actions can be formulated as one action with 240 discrete values or as 5 separate actions, which are either 15 or 2 discrete values. With this formulation of the action space, $A = [a_1, a_2, a_3, a_4, a_5]$. A distributed RL formulation of the problem can be written as in Equation (5).

$$\begin{aligned} Q^i_{t+1}(s, a^1, ., a^n) = \\ \mathbb{E}_{s,a,r,s'}\left[(1-\alpha)Q^i_t(s, a^1.a^n) + \alpha[r^i_t + \beta \max_{a^i} Q^i_t(s', a^1, ., a^n)]\right]. \end{aligned} \tag{5}$$

Table 3. Action space for Single Agent RL ann Multi Agent RL.

	SARL	MARL
Single Agent/Mixing Agent	240	15
Valve Agent 1	0	2
Valve Agent 2	0	2
Valve Agent 3	0	2
Valve Agent 4	0	2

From Equation (5), it can be seen that all agents have a Q-function and have full observability of the actions of all other agents. However, if the UFH system is investigated, it can be argued that each valve agent has little to no effect on each other. For this reason, it can be argued that the connections between the valve agents are unnecessary and therefore should be removed. Removing these connections will give the following formulation of the Q-functions, see Equations (6) and (7). Here, $1, \cdots, m$ refers to valve 1 to valve m and st refers to the supply temperature:

$$\begin{aligned} Q^{st}_{t+1}(s, a^{st}, a^{v_{1.m}}) = \\ \mathbb{E}_{s,a,r,s'}\left[(1-\alpha)Q^i_t((a^{st}, a^{v_{1.m}}) + \alpha[r^i_t + \beta \max_{a^{st}} Q^{st}_t(s', a^{st}, a^{v_{1.m}}))]\right], \end{aligned} \tag{6}$$

$$\begin{aligned} Q^{v_m}_{t+1}(s, a^{st}, a^{v_m}) = \\ \mathbb{E}_{s,a,r,s'}\left[(1-\alpha)Q^i_t((s, a^{st}, a^{v_m}) + \alpha[r^i_t + \beta \max_{v_m} Q^{v_m}_t(s', a^{st}, a^{v_m})]\right]. \end{aligned} \tag{7}$$

Since it is argued that zones have little to no effect on each other, it can also be argued that parts of the state space should only be locally observed. For this reason, the local state space $[s_{st}, s_{v_1}, s_{v_2}, s_{v_3}, s_{v_4}]$ are defined, and the Q-functions can be rewritten to Equations (8) and (9). Elaboration on which states are relevant for which agents are given in Section 4.3.

$$Q^{st}_{t+1}(s_{st}, a^{st}, a^{v_{1..m}}) = \mathbb{E}_{s,a,r,s'} \left[(1-\alpha)Q^{i}_{t}((a^{st}, a^{v_{1..m}}) + \alpha[r^{i}_{t} + \beta \max_{a^{st}} Q^{st}_{t}(s'^{st}, a^{st}, a^{v_{1..m}})]\right] \tag{8}$$

$$Q^{v_m}_{t+1}(s^{v_m}, a^{st}, a^{v_m}) = \mathbb{E}_{s,a,r,s'} \left[(1-\alpha)Q^{i}_{t}((s^{v_m}, a^{st}, a^{v_m}) + \alpha[r^{i}_{t} + \beta \max_{v_m} Q^{v_m}_{t}(s'^{v_m}, a^{st}, a^{v_m})]\right]. \tag{9}$$

With the Q-functions formulated, the foundation for the MARL algorithm is established. An illustration of the structure of how the agents are interacting with the environment can be seen in Figure 7.

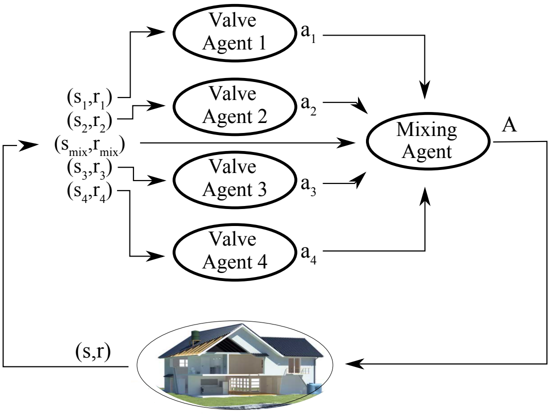


Figure 7. Illustration of how the agents interact with each other and the environment. In the figure, four valve agents, one mixing agent, and a four-zone Underfloor Heating System can be seen. The sequence of interactions is as follows; all valve agents choose an action based on the state of the environment. These actions are passed to the mixing agent, the mixing agent chooses an action based on the state of the environment and the actions of the valve agents. All actions are passed to the environment and the environment returns states and rewards for the agents.

The following explains (1) which RL methods are used, (2) the pseudo-code for the algorithm, (3) how the reward functions are formulated, (4) how the state-action space is distributed, and (5) which hyper-parameters are used.

4.1. Reinforcement Learning Methods

The backbone RL algorithm used in this paper is the deep double Q-network algorithm with experience replay, N-step learning, and epsilon greedy decay exploration.

The N-step eligibility trace used in the algorithm is also used in [41,42]. This approach is chosen due to the slow dynamic of the UFH system, making eligibility traces desirable. Experience replay is also used to enhance data efficiency. The implementation of experience replay is customized to N-step eligibility trace and MARL. This is done by maintaining the experience in mini-batches with the same length as the N-step eligibility trace. Additionally, the experience replay is synchronized between the agents, so the experience for $agent_1$ at time t has the same timestamp as the experience for $agent_i$ at time t. The pseudo-code for the MARL algorithm can be seen in Algorithm 1.

Algorithm 1 MARL Deep Q-Learning.

```
for each iteration k do
    for each environment step do
        Observe state $S_t$ and distribute local states $s_t^1, s_t^2 ... s_t^n$ to respective agents.
        Valve agents select action $a^k_{v(n)} = \max_{a_{v(n)}} Q(s, a^{k-1}_{mix}; \theta)$ or pick random action with probability epsilon.
        Mixing agent select action $a^k_{mix} = \max_{a_{mix}} Q(s, a^k_{v(n)}; \theta)$ or pick random action with probability epsilon.
        Collect action $a_t^1, a_t^2 ... a_t^n$ to A execute $A_t$ and observe next state $S_{t+1}$ and reward.
        Store $(s_t^n, a_t^n, r_t^n, s_{t+1}^n)$ in replay buffers $\mathcal{D}^n$.
        Decay epsilon
    end for
    for each update step do
        Agent n sample experience of size B each with length n-step from $\mathcal{D}^n$.
        Compute Q-values as described in Equations (7) and (9).
        Calculate losses for all agents.
        Calculate gradients with respect to the network weights $\theta$ and perform gradient step.
        Every C environment step, update target networks.
    end for
end for
```

4.2. Reward Functions

To gain intuition about the reward functions, two base functions are defined—one for the valve system and one for the supply temperature.

4.2.1. Valve Reward

The valve reward is shown in Equation (10). The reward function depends on two sub-functions shown in Equations (11) and (12).

$$\mathrm{R}(T_z, V, H_c) = \begin{cases} 2 - (T_z - T_{ref}) \cdot & \text{if } 21.6 < T_z < 22 \\ -(T_z - T_{ref}) & \text{if } 21.6 > T_z \text{ or } T_z > 22 \;, \\ -H_c & \text{if } SC = active \end{cases} \tag{10}$$

$$\mathrm{SC}(T_z, V) = \begin{cases} \text{not active} & \text{if } 21 < T_z > 23 \\ \text{active} & \text{if } T_z < 21 \text{ and } VP = 0 \;, \\ \text{active} & \text{if } T_z < 23 \text{ and } VP = 1 \end{cases} \tag{11}$$

$$H_c\;(\mathrm{SC}) = \begin{cases} 1 + H_C & \text{if SC = active} \\ 5 & \text{if SC = not active} \end{cases} \tag{12}$$

The abbreviations in the equations above are the following: R = Reward SC = Safety controller, T_z = Zone temperature, V = Valve position, and H_c = Hard constraint.

The two sub-functions Equations (11) and (12) are a part of a safety framework for ensuring robust behavior in RL [43]. In [43], it is demonstrated that by implementing a safety controller, in this case a tolerance controller, on top of the RL algorithm, robust behavior can be ensured. The safety controller is activated when the RL controller is trying to explore the action-state spaces that are known to be outside safety boundaries. The H_C variable is the soft constraint that iteratively increases linearly as the agent continuously tries to explore parts of the action-state spaces, which is known to have negative comfort characteristics. The immediate +2 reward, which is received when the agent is less than

−0.4 °C from the reference point, enforces that the comfort is highest when the temperature is in this range. When the reward function is used in the MARL setting, the reward is simply distributed so that each agent receives the reward for the zone it is controlling. When the rewards are used in the SARL setting, all rewards are summed into one reward.

4.2.2. Supply Temperature Reward

The reward function for the supply temperature is shown in Equation (13), and the associated sub-functions are shown in Equations (14) and (15).

$$R(T_z, V, H_c) = \begin{cases} 2 - (T_z - T_{ref}) - P & \text{if } 21.6 < T_z < 22 \text{ and } VP = 1 \\ -(T_z - T_{ref}) - P & \text{if } 21.6 > T_z \text{ or } T_z > 22 \end{cases} \tag{13}$$

$$SC(T_z, V) = \begin{cases} \text{not active} & \text{if } T_z, > 20.5 \\ \text{active} & \text{if } T_z, < 20.5 \text{ and VP = 1} \end{cases} \tag{14}$$

$$H_c(SC) = \begin{cases} 1 + H_C & \text{if SC = active} \\ 5 & \text{if SC = not active} \end{cases} \tag{15}$$

The abbreviations in the equations above are the following: R = Reward, SC = Safety controller, T_z = Zone temperature, V = Valve position, H_c = Hard constraint, and P = Price.

From Equation (13), it is seen that the reward is much like the reward from the valve agent, with the difference that the $+2$ reward requires that the valve is open. When heating with a heat pump, it is optimal to have as much water circulation as possible. Adding that the reward is highest when the valve is open enforces this behavior. The price is a scalar between 0 and 1, and the lower the price of heating, the better.

Like in the valve reward function (Equation (10)), a safety controller is put on top of the RL algorithm. Simulation results show that it has some of the same effects as in the case of the valve agent reward. The safety controller is the *outside compensate supply temperature* used in the simulations in Section 3.2. The safety controller is activated whenever the temperature in a given zone is 1.5 °C lower than the reference temperature, and the associated valve is open.

4.3. Input States

The input states to the RL agent are a mix of actual states of the system and parameters that are functions of the system's states. The input states are divided into valve input states and supply input states. The input state space is explained from a MARL point of view. From a SARL point of view, the same states are used. These are combined in a single tuple and sent to the single agent.

4.3.1. Valve Input States

Seven states are used in the valve agents. All states are normalized so they assume a value from 0 to 3. These states can be seen in the list below.

- Valve agent input:
  - Room Temp $i \in \{1, \cdots, n\}$, [°C];
  - Δ Room Temp $i \in \{1, \cdots, n\}$, [°C];
  - Hard constraint Valve $i \in \{1, \cdots, n\}$;
  - Supply temperature, [°C];
  - Ambient temperature, [°C];
  - Sun, [w/m^2];
  - Time of day, [hour and minutes].

From the list above, seven input states can be seen, the "Δ Room Temp" is the gradient of the room temperature, and the "Sun" is the strength of the Sun. In the weather file, this strength is measured in [W/m^2].

4.3.2. Supply Temperature Input States

The input states for the supply temperature can be seen in the list below.

- Supply agent input:
  - Room Temp$i \in \{1, \cdots, n\}$, [°C];
  - Δ Room Temp$i \in \{1, \cdots, n\}$, [°C];
  - Hard constraint Supply;
  - Supply Temperature, [°C];
  - Ambient Temperature, [°C];
  - Sun, [w/m^2];
  - Time of day, [hour and minutes];
  - Price, [Euro].

From the above, it can be seen that many of the states are the same as in the valve agent, only the price and the hard constraint states are different. An overview of how the number of states increases as the number of temperature zones also increases is given in Table 4.

Table 4. Overview of how the number of input states is increasing when more zones are added to the system for both MARL and SARL.

	Supply Agent	Valve Agent	Single Agent
One zone	8	7	10
Four zones	17	7	22

4.3.3. Action Space

As explained in Section 3, the action space for the SARL formulation for a four-zone UFH system is a discrete value from 0 to 239 and the action space for a MARL system is a vector as follows: $A = [a_1, a_2, a_3, a_4, a_5]$, see Table 3.

Tests in the simulation have shown that when doing simulations with SARL, it is necessary to manipulate the action state space, so that there are 31 actions instead of 240 for a four-zone system. This reduction is done by separating the control of the valves and the control of the supply temperature. That is, the agent can either control the valves or the supply temperature at a given step. This reduction results in 16 actions for the valves, and 15 for the supply temperature, hence the 31 actions all in all. The reduction of the action space is also done in SARL for the simulation of the one-zone system resulting in the 16 actions. That is, 15 mixing actions and one action for closing the valve, hence the action state space for the one-zone SARL and MARL algorithms are similar and therefore, it is expected that the convergence time will be similar. An overview of the action space in the different settings can be seen in Table 5.

Table 5. Overview of how the input states are increasing when more zones are added to the system and how MARL and SARL are affected by this.

	Supply Agent	Valve Agent	Single Agent
One zone	15	2	16
Four zones	15	2	31

4.3.4. Hyperparameters

The hyperparameters used for training and testing the algorithms are displayed in Table 6. The values seen below are found from empirical tests of the algorithms.

Table 6. Hyperparameters used for training the agents.

	Supply Agent	Valve Agent	Single Agent
Learning rate	0.01	0.01	0.01
Epsilon decay	0.0005	0.0005	0.0005
Epsilon max	1	1	1
Epsilon min	0.1	0.1	0.1
batch size	432	432	432
N_steps	45	45	45
gamma	0.9	0.9	0.9
ANN	60 × 60 × 60	60 × 60 × 60	90 × 90 × 90
Target update rate	540	540	540

The following section will present a test plan explaining which experiments are required to prove scalability in MARL and increased performance when compared to a traditional controller. Furthermore, the results of the experiments will be presented and analyzed. All raw data obtained from these simulations can be found in (https://github.com/ChrBlad/MARL_data, accessed on 30 March 2021).

5. Experiments and Results

A test plan is established to validate that the MARL formulation reduces training time and hence improves scaling capabilities compared to a SARL formulation. The test plan includes two test levels, consisting of environments with one and four temperature zones, respectively. By introducing these test levels, it is verified that the scaling problem stated in the introduction is solved using MARL. The test plan is outlined in Table 7 and consists of six tests and three comparisons. Both test levels consist of simulations of 1000 days. To review how MARL performs in comparison to SARL, the reward of the MARL is compared to the reward of SARL.

Table 7. This test plan elaborates on which experiments are necessary to prove scalability in MARL and better performance in Reinforcement Learning when comparing with traditional control (TC).

	SARL	MARL	TC
Test Level 1	1	1	0
Test Level 2	1	1	1

5.1. Test Level 1

In Test level 1, a single zone UFH system is simulated with a MARL and a SARL controller, respectively.

Figure 8 shows that the MARL and SARL algorithms converge in approximately the same time, about 180 days. This is in accordance with the assumption that the convergence time should correlate with the size of the action state space. Since the distribution of agents in a one-zone system almost results in the same action state space as if it was one agent, the convergence time is more or less the same. The performance of the RL algorithms, compared to a traditional controller, is better after 40 days of training. However, a drop in performance is seen after 80 days. This drop is due to change of seasons and therefore load conditions that are unknown to the RL. After 120 days, the RL algorithms are shown to perform better than the traditional controller. There are a few days during a heating season where the MARL and SARL controllers are not performing better than the traditional controller. This can be found around day 220 and 240. These days are exceptionally cold and therefore the system is in saturation and therefore the RL-controller cannot improve the performance.

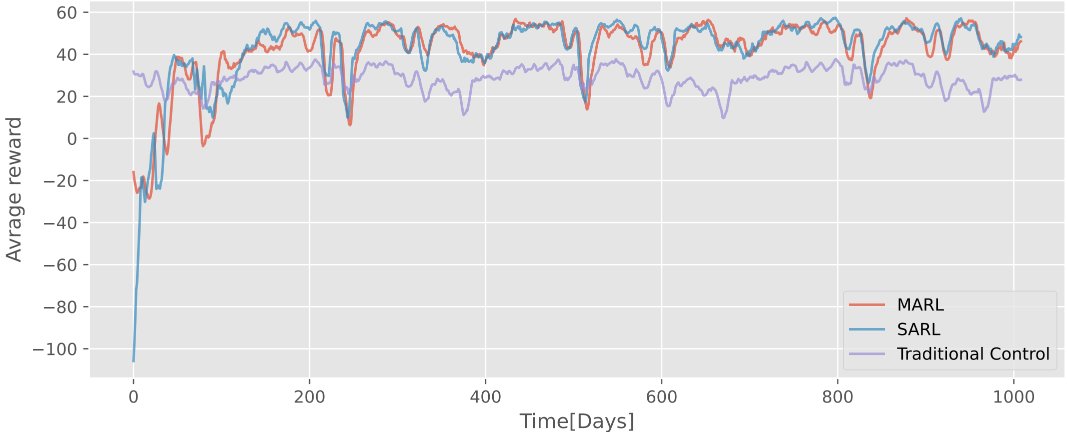


Figure 8. Reward signal of the one-zone UFH system. Simulation time is 1000 days.

Some variation in the reward signal is observed after the algorithms have converged. This is due to the seasonal effect on the comfort and prices of heating. This is to be expected and may be different for the reward plots seen in other articles, where the reward converges to a constant value.

5.2. Test Level 2

In test level 2, the SARL and MARL performance in a four-zone UFH system is compared. The result of test level 2 is shown in Figure 9.

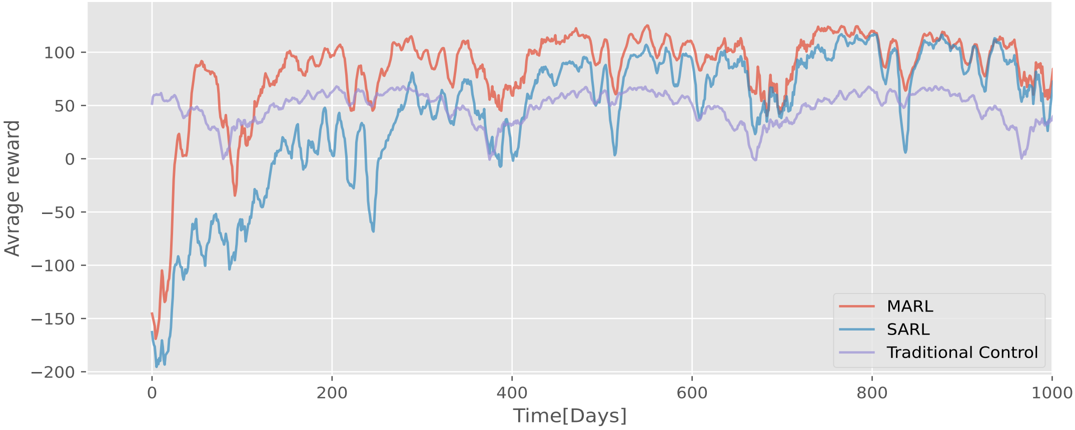


Figure 9. Reward signal of the four-zone UFH system in, simulation time is 1000 days (working on 2000-day simulation results).

From the reward plot of test level 2, some of the same behavior as in test level 1 can be observed. The MARL agents converge after 180 days but are performing well after 40 days. The SARL agent converges to approximately the same as the MARL but after 600 days. This difference in convergence speed confirms the assumption on the relation between convergence speed and size of the action space. Whereas SARL and MARL both works for a one-zone UFH system, the advantages with MARL are clear in the four-zone simulation. MARL results in faster convergence and marginal better convergence over a time period of 1000 days.

As the SARL and MARL converge to almost the same policy with identical performance, there is no reason to compare the performance in terms of comfort and price. However, it is interesting to compare the RL performance to the traditional controller. For this comparison, MARL is used.

Traditional Control vs. MARL

The reward signal is a good measurement of performance. It does, however, not explain how much better a MARL algorithm performs in terms of comfort and price compared to a traditional controller. The performance in terms of comfort and price is therefore evaluated.

Firstly, the room temperature data for zone #1 is evaluated from a histogram to determine the distribution of the temperature data.

As shown in Figure 10, the temperature distribution for a traditional controller is far from normally distributed. For this reason, standard deviations or variance cannot be used to calculate performance. A box plot for each temperature zone is therefore used. From this plot, it is possible to conclude on the variation in the room temperature for each zone.

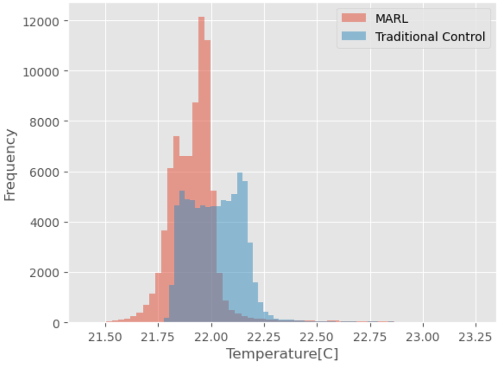


Figure 10. Histogram of room temperature in zone #1 from simulation with four temperature zones.

Figure 11 contains four box plots showing the temperature variations in each of the four temperature zones. From these plots, it can be deduced that the variation is about 40% less with MARL compared to a traditional controller. Note, outliers have been removed from the data before being used for the box plots. Smaller variations in the temperature give both better comfort and reduced price. Hence the MARL agents behave like this.

The cost of heating with the MARL algorithm, the traditional controller, and the savings as a function of time can be seen in Figure 12. The cost is calculated based on the description in Section 3.2 and Equation (4).

From this plot, it can be concluded that the savings vary over the season, but the MARL controller performs better than the traditional controller at any point in time. The average savings are 19% when using MARL compared to a traditional controller. During the first 20 weeks, it can be seen that the savings oscillate more then the remaining 120 weeks, the reason for this is naturally that the control policy of the MARL agent has not jet converged.

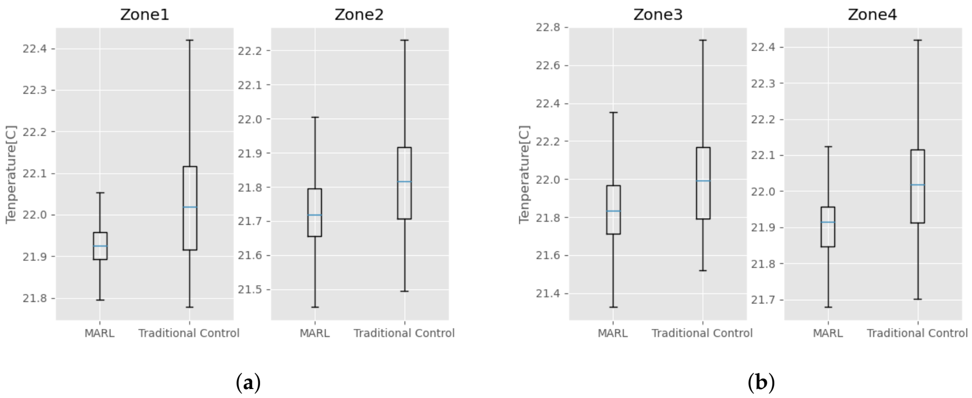


Figure 11. Boxplots of temperature distribution in (**a**) zone 1, zone 2, (**b**) zone 3, and zone 4 in the four-zone UFH system for the MARL simulation and the simulation with a traditional control policy.

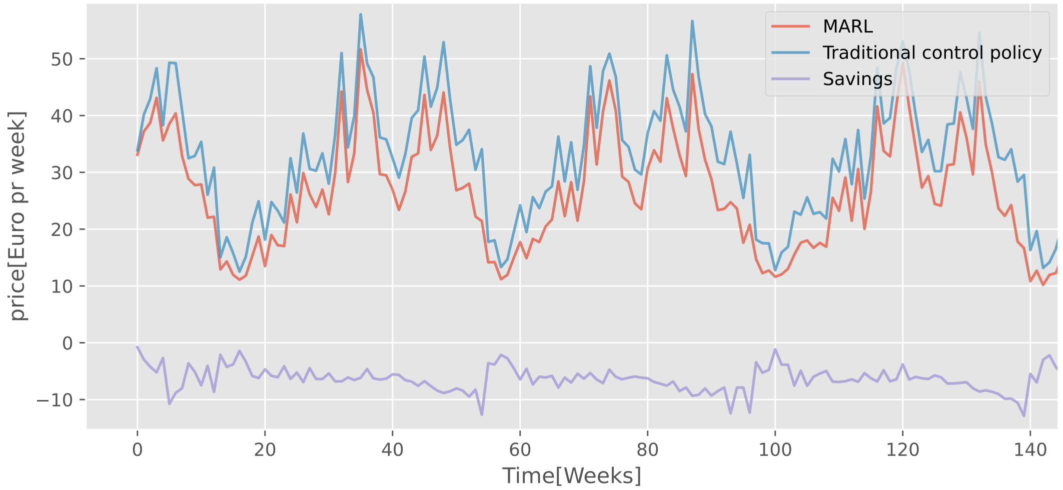


Figure 12. Price of heating per week for MARL and traditional control for a four-zone UFH system over a 1000-day period.

6. Discussion

The results of this paper is based on a unique simulation environment, for this reason general savings of 19% can not be proven, and there is no standard model to benchmark against. However when reviewing Sections 3 and 3.2, it is shown that both the benchmark controller and the simulation environment gives a valid view of a real world test. In the Git repository, the simulation environments and control code are available. The authors do encourage researchers to benchmark further algorithms against this code. Since this is a simulation environment and not a real world test, the algorithm has not been tested for robustness towards sensor faults, sliding errors in sensor measurement, or stochastic behavior. The main purpose of this paper is to prove that this algorithm can converge fast to the thermal properties of a house. Further research will prove if this algorithm is resilient towards stochastic behavior in the forms mentioned above.

7. Conclusions

In this paper, we explore a novel approach for building control strategies with Multi-Agent Reinforcement Learning for underfloor heating systems. Formulating the Underfloor Heating (UFH) system as a Markov Game (MG) instead of a Markov Decision Process (MDP) makes it possible to distribute the action space locally between agents. Moreover, it is argued that it is possible to have some states of the system observed locally.

By using a multi-agent structure and observe states locally, it is demonstrated in simulation that convergence time can be reduced by more than 70% when compared with a single agent approach. Furthermore, it is shown that as the complexity of the state-action space increases, the convergence time of the MARL agent will remain acceptable. In comparison, a SARL agent becomes almost unfeasible. Additionally, the simulation experiments show that MARL is a better alternative to traditional control methods when comparing heating costs. The simulation shows a 19% reduction of the heating cost. If assuming a Seasonal Coefficient of Performance (SCOP) of 4, the average size of a house is 140 m^2 [44], and the average heat consumption is 115 kWh per m^2, these savings sum to approximately 750 kWh of electric energy per year in an average Danish household.

Author Contributions: All authors has contributed equally to this study. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the Innovation Fund Denmark.

Data Availability Statement: The data supporting this study can be found in https://github.com/ChrBlad/MARL_data, accessed on 30 March 2021.

Acknowledgments: Supported by Innovation Fund Denmark and Grundfos A/S. A special thanks to Erik B. Sørensen, Control Specialist, Grundfos A/S for supporting the work on the simulation model.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pérez-Lombard, L.; Ortiz, J.; Pout, C. A review on buildings energy consumption information. *Energy Build.* **2008**, *40*, 394–398. [CrossRef]
2. Gaglia, A.G.; Tsikaloudaki, A.G.; Laskos, C.M.; Dialynas, E.N.; Argiriou, A.A. The impact of the energy performance regulations' updated on the construction technology, economics and energy aspects of new residential buildings: The case of Greece. *Energy Build.* **2017**, *155*, 225–237. [CrossRef]
3. Privara, S.; Širokỳ, J.; Ferkl, L.; Cigler, J. Model predictive control of a building heating system: The first experience. *Energy Build.* **2011**, *43*, 564–572. [CrossRef]
4. Huang, H.; Chen, L.; Hu, E. A new model predictive control scheme for energy and cost savings in commercial buildings: An airport terminal building case study. *Build. Environ.* **2015**, *89*, 203–216. [CrossRef]
5. Yu, L.; Jiang, T.; Zou, Y. Online energy management for a sustainable smart home with an HVAC load and random occupancy. *IEEE Trans. Smart Grid* **2017**, *10*, 1646–1659. [CrossRef]
6. Tsui, K.M.; Chan, S.C. Demand response optimization for smart home scheduling under real-time pricing. *IEEE Trans. Smart Grid* **2012**, *3*, 1812–1821. [CrossRef]
7. Kull, T.M.; Simson, R.; Thalfeldt, M.; Kurnitski, J. Influence of time constants on low energy buildings' heating control. *Energy Procedia* **2017**, *132*, 75–80. [CrossRef]
8. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
9. Vázquez-Canteli, J.R.; Nagy, Z. Reinforcement learning for demand response: A review of algorithms and modeling techniques. *Appl. Energy* **2019**, *235*, 1072–1089. [CrossRef]
10. Yu, L.; Sun, Y.; Xu, Z.; Shen, C.; Yue, D.; Jiang, T.; Guan, X. Multi-agent deep reinforcement learning for HVAC control in commercial buildings. *IEEE Trans. Smart Grid* **2020**, *12*, 407–419. [CrossRef]
11. Ruelens, F.; Claessens, B.J.; Vandael, S.; De Schutter, B.; Babuška, R.; Belmans, R. Residential demand response of thermostatically controlled loads using batch reinforcement learning. *IEEE Trans. Smart Grid* **2016**, *8*, 2149–2159. [CrossRef]
12. Wang, S.; Duan, J.; Shi, D.; Xu, C.; Li, H.; Diao, R.; Wang, Z. A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning. *IEEE Trans. Power Syst.* **2020**, *35*, 4644–4654. [CrossRef]
13. Zolfpour-Arokhlo, M.; Selamat, A.; Hashim, S.Z.M.; Afkhami, H. Modeling of route planning system based on Q value-based dynamic programming with multi-agent reinforcement learning algorithms. *Eng. Appl. Artif. Intell.* **2014**, *29*, 163–177. [CrossRef]
14. Sun, Z.; Piao, H.; Yang, Z.; Zhao, Y.; Zhan, G.; Zhou, D.; Meng, G.; Chen, H.; Chen, X.; Qu, B.; et al. Multi-agent hierarchical policy gradient for Air Combat Tactics emergence via self-play. *Eng. Appl. Artif. Intell.* **2021**, *98*, 104112. [CrossRef]

15. Barrett, E.; Linder, S. Autonomous hvac control, a reinforcement learning approach. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Porto, Portugal, 7–11 September 2015; pp. 3–19.
16. Overgaard, A.; Kallesøe, C.S.; Bendtsen, J.D.; Nielsen, B.K. Mixing Loop Control using Reinforcement Learning. In *E3S Web of Conferences*; EDP Sciences: Ulis, France, 2019; Volume 111, p. 05013.
17. Wei, T.; Wang, Y.; Zhu, Q. Deep reinforcement learning for building HVAC control. In Proceedings of the 54th Annual Design Automation Conference 2017, Austin, TX, USA, 18–22 June 2017; pp. 1–6.
18. Kazmi, H.; Suykens, J.; Balint, A.; Driesen, J. Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads. *Appl. Energy* **2019**, *238*, 1022–1035. [CrossRef]
19. Gupta, J.K.; Egorov, M.; Kochenderfer, M. Cooperative multi-agent control using deep reinforcement learning. In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, São Paulo, Brazil, 8–12 May 2017; pp. 66–83.
20. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
21. Van Hasselt, H.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-learning. *arXiv* **2015**, arXiv:1509.06461.
22. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized Experience Replay. *arXiv* **2015**, arXiv:1511.05952.
23. Bellman, R. Dynamic programming. *Science* **1966**, *153*, 34–37. [CrossRef] [PubMed]
24. Buşoniu, L.; Babuška, R.; De Schutter, B. Multi-agent reinforcement learning: An overview. In *Innovations in Multi-Agent Systems and Applications-1*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 183–221.
25. Zhang, K.; Yang, Z.; Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv* **2019**, arXiv:1911.10635.
26. Littman, M.L. Value-function reinforcement learning in Markov games. *Cogn. Syst. Res.* **2001**, *2*, 55–66. [CrossRef]
27. Lauer, M.; Riedmiller, M. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In Proceedings of the Seventeenth International Conference on Machine Learning, Standord, CA, USA, 29 June–2 July 2000.
28. Littman, M.L. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings*; Elsevier: Amsterdam, The Netherlands, 1994; pp. 157–163.
29. Bowling, M.; Veloso, M. Multiagent learning using a variable learning rate. *Artif. Intell.* **2002**, *136*, 215–250. [CrossRef]
30. Crites, R.H.; Barto, A.G. Improving elevator performance using reinforcement learning. *Adv. Neural Inf. Process. Syst.* **1996**, 1017–1023.
31. Matarić, M.J. Reinforcement learning in the multi-robot domain. In *Robot Colonies*; Springer: Berlin/Heidelberg, Germany, 1997; pp. 73–83.
32. Hu, J.; Wellman, M.P. Multiagent reinforcement learning: Theoretical framework and an algorithm. In Proceedings of the International Conference on Machine Learning (ICML), Madison, WI, USA, 24–27 July 1998; Volume 98, pp. 242–250.
33. Banerjee, B.; Peng, J. Adaptive policy gradient in multiagent learning. In Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, Melbourne, Australia, 14–18 July 2003; pp. 686–692.
34. Zhang, K.; Yang, Z.; Liu, H.; Zhang, T.; Basar, T. Fully decentralized multi-agent reinforcement learning with networked agents. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5872–5881.
35. Bertsekas, D. Multiagent reinforcement learning: Rollout and policy iteration. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 249–272. [CrossRef]
36. Wetter, M.; Zuo, W.; Nouidui, T.S.; Pang, X. Modelica buildings library. *J. Build. Perform. Simul.* **2014**, *7*, 253–270. [CrossRef]
37. El-Priser og Afgifter. Available online: https://www.vivaenergi.dk/el-priser-og-afgifter (accessed on 30 March 2021).
38. Nie, J.; Li, Z.; Kong, X.; Li, D. Analysis and Comparison Study on Different HFC Refrigerants for Space Heating Air Source Heat Pump in Rural Residential Buildings of North. *Procedia Eng.* **2017**, *205*, 1201–1206. [CrossRef]
39. Piechurski, K.; Szulgowska-Zgrzywa, M.; Danielewicz, J. The impact of the work under partial load on the energy efficiency of an air-to-water heat pump. *E3S Web Conf.* **2017**, *17*, 00072. [CrossRef]
40. Se Det gns. Varmeforbrug I husstande der Ligner Din. Available online: https://seas-nve.dk/kundeservice/forbrug/gennemsnitsforbrug/varmeforbrug/ (accessed on 30 March 2021).
41. Blad, C.; Koch, S.; Ganeswarathas, S.; Kallesøe, C.; Bøgh, S. Control of hvac-systems with slow thermodynamic using reinforcement learning. *Procedia Manuf.* **2019**, *38*, 1308–1315. [CrossRef]
42. Overgaard, A.; Nielsen, B.K.; Kallesøe, C.S.; Bendtsen, J.D. Reinforcement Learning for Mixing Loop Control with Flow Variable Eligibility Trace. In Proceedings of the IEEE Conference on Control Technology and Applications (CCTA), Hong Kong, China, 19–21 August 2019; pp. 1043–1048.
43. Blad, C.; Kallesøe, C.S.; Bøgh, S. Control of HVAC-Systems Using Reinforcement Learning With Hysteresis and Tolerance Control. In Proceedings of the IEEE/SICE International Symposium on System Integration (SII), Honolulu, HI, USA, 12–15 January 2020; pp. 938–942.
44. Danmarks Statestik. Available online: https://www.statistikbanken.dk/bygv06/ (accessed on 30 March 2021).

Paper D

PaperD

Pre-Print
Data-Driven Offline Reinforcement Learning for HVAC-Systems

Christian Blad[a,c,*], Simon Bøgh[a], Carsten Skovmose Kallesøe[b,c]

[a] *Robotics & Automation Group*
Department of Materials and Production, Aalborg University, Denmark
[b] *Department of Electronic Systems, Aalborg University, Denmark*
[c] *Grundfos A/S, Poul Due Jensens Vej 7, Bjerringbro, DK-8850 Denmark*

Abstract

This paper presents a novel framework for Offline Reinforcement Learning (RL) with online fine tuning for Heating Ventilation and Air-conditioning (HVAC) systems. The framework present a method to do pre-training in a black box model environment, where the black box models are build on data acquired under a traditional control policy. The paper focuses on the application of Underfloor Heating (UFH) with an air-to-water-based heat pump. However, the framework should also generalize to other HVAC control applications. Because Black box methods are used is there little to no commissioning time when applying this framework to other buildings/simulations beyond the one presented in this study. This paper explores and deploys Artificial Neural Network (ANN) based methods to design efficient controllers. Two ANN methods are tested and presented in this paper; a Multilayer Perceptron (MLP) method and a Long Short Term Memory (LSTM) based method. It is found that the LSTM-based method reduces the prediction error by 45% when comparing with a MLP model. Additionally, different network architectures are tested. It is found that by creating a new model for each timestep, performance can be improved additionally 19%. By using these models in the framework presented in this paper, it is shown that a Multi-Agent RL algorithm can be deployed without ever performing worse than an industrial controller. Furthermore, it is shown that if building data from a Building Managements System (BMS) is available, an RL agent can be deployed which performs close to optimally from the first day of deployment.

[*] Corresponding author, Fibigerstræde 16, 9220 Aalborg East, Denmark
Email addresses: cblad@m-tech.aau.dk (Christian Blad), sb@mp.aau.dk (Simon Bøgh), csk@es.aau.dk (Carsten Skovmose Kallesøe)

Preprint submitted to Elsevier *May 11, 2022*

1. Introduction

Heating Ventilation and Air-Conditioning (HVAC) systems are today consuming approximately 40% of the annual energy consumption in the US which is assumed to be true for much of the western world as well [1]. There are multiple ways of making these systems more efficient, one of them being improving the control algorithms. Traditionally, control systems for HVAC systems are event-based controllers typically based on; the temperature of the zone (hysteresis control), the ambient temperature (outside-compensated supply temperature)', and the time of day (scheduling) [2].

Event-based controllers, like the one described, do not allow for any predictive control and because of the delayed and slow responses associated with HVAC, especially for radiant heating or cooling, this is not optimal. Furthermore, the cost of energy and the efficiency is not constant. For compressor systems the efficiency dependents on the ambient temperature, the part load factor, and energy prices. Hence, the price of heating is highly dependent on what happens not only in the current time step but also what happens in the following time steps [2].

A common method to do predictive control is Model Predictive Control (MPC). This has previously been described in the literature in relation to HVAC systems [3, 4, 5]. When doing MPC a model is required, however, not two buildings are alike and the dynamic of a building can also change over its lifetime, which requires a new model for each scenario. For these reasons, MPC controllers for buildings are both expensive to make and can also be expensive to maintain.

Other smart controllers are scheduling energy usage according to energy prices [6, 7]. These controllers naturally need a model to predict energy usage and are therefore, like the MPC controller, expensive to commission.

An expensive commissioning phase is a cause for concern. A study of 150 existing commercial buildings showed that a recommissioning could reduce the energy consummation by 15% on average [8]. Model-free Reinforcement Learning (RL) is, as the name suggest, a model-free method to do predictive control [9, 10], hence do not require the commissioning of a model. Numerous papers concerning the usages of RL in HVAC systems have been published [11, 12, 13, 14]. These papers show that RL algorithms compared to traditional event-based controllers can reduce costs between 5.5% and 15%. The papers describe the problem with using RL and how it requires a substantial amount of time/data to converge towards a optimal solution.

To overcome slow convergence, Multi-Agent Reinforcement Learning (MARL) for HVAC systems has been proposed in [15, 16, 17]. In MARL, the environment is formulated as a Markov Game which reduces the complexity of the action space. In [17], additional steps have been taken to reduce the complexity of the action state space, hence reducing the convergence time.

This paper proposes a model-free offline MARL algorithm as a solution to the problem of poor behavior during early training of the RL agents. This is done under the assumption that a traditional controller is accessible and has an acceptable performance.

Offline training of a RL agent has been applied in HVAC systems in [18]. However, this approach is based on an extensive model, which is as expensive to commission as an MPC controller. Offline RL for HVAC systems based on available data, has been proposed in [19] but the idea has not been developed. In [20], model based RL is used in an online fashion to control airflow. The model in the paper is a grey-box model, hence based on an actual model where the parameters are approximated by an artificial neural network (ANN) and thereby more generic than MPC. However, grey-box models can only model the dynamics of the model on which it is based on which is a limitation when taking use for general purpose and not any specific environment,

The idea

This paper explores the possibility of training the RL/MARL agents in black box model environments, before deploying the control algorithms in the real-world system. Because this paper strives to use a data-driven model/models, data is required for the model to be obtained, which is why this article works with two scenarios:

- Scenario A: A new installation where there is no prior data from the environment.
- Scenario B: A recommission of an existing installation where prior operating data is available.

The model is a black box nonlinear regression model that to a large extend is able to model the dynamics of the real world environment. How this model is designed is explained in section 4. In Figure 1 scenario A can be seen. Scenario B can be derived from Figure 1 by removing step 1 where the ”traditional controller” is interacting with the environment.

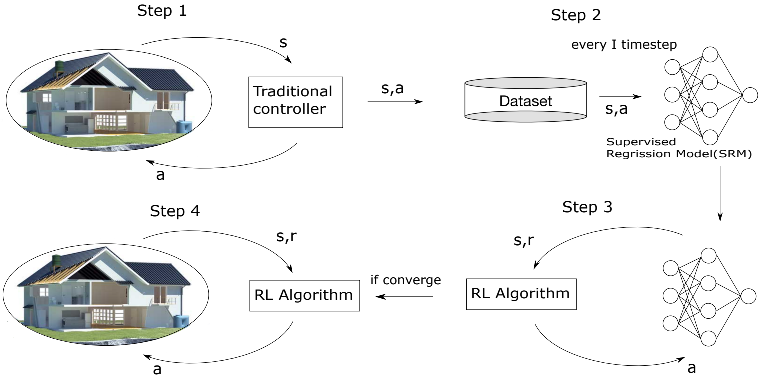


Figure 1: Illustration of scenario A: The traditional controller interacts with the environment T time steps, after each time step the action and state transition is saved in the data buffer. The data set is passed to the SRM ones trained, the SRM is used as a artificial environment for the RL agent to train until convergence. The trained agent is then deployed in the real environment, the agent can still do limited exploration for fine tuning. Step 2, 3 and 4 will be repeated until the SRM converges.

The goals of this paper is to verify, in a Dymola Simulation, that the above described framework does work in both new commissions of buildings and commissions where data is available.

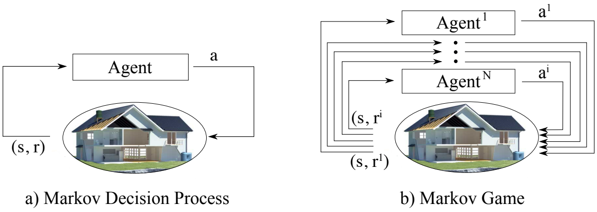
Above, the motivation, related work, and the overall idea for this paper has been described. Following this, in section 2 the background for this paper is explained. The simulation environment, on which the results of this paper is based, is described and evaluated in section 3. This environment is built in Dymola, hence the results of this paper is purely simulation based. The reason for doing a simulation based study, is that it takes years of data to complete this study which in real-time would make this study difficult to realize. In section 4 the black box model is designed and evaluated. The evaluation is done by comparing the black box model to the results of the Dymola simulation model which is considered the ground truth. In section 5 the RL framework presented in section 1 is deployed in the Dymola simulation and a comparison with a normal RL deployment, and a traditional deployment is made. Lastly, the results are concluded in section 7.

2. Background and Contributions

This section gives insights into model-free RL, MARL, Offline training, and black box model generation.

2.1. Reinforcement Learning

Model-free RL is a learning method that by interacting with the environment learns an optimal control policy π^* [9]. In Single Agent RL (SARL), the interaction between environment and agent is defined as a Markov Decision Process (MDP) and for Multi Agent RL it is a Markov Game (MG). An illustration of this interaction is shown in Figure 2.



a) Markov Decision Process

b) Markov Game

Figure 2: Illustration of difference between a Markov decision process and a Markov game. In figure 1.a can one agent and one environment be seen. The agent interacts with the environment by sending a tubule of actions and receiving one reward and the states of the environment. In figure 1.b can multiple agents and one environment be seen. The action space is split into i number of actions, each agent is receiving a reward and the states of the environment.

As seen in the illustration in Figure 2, the difference between an MDP and an MG is that multiple agents are controlling the environment. The reason for formulating a problem as an MG is often due to the complexity of the action-state space. Richard Bellman formulated it as RL suffer under ”the curse of dimensionality” which means as the complexity increases, so does the time required to converge towards a solution [21]. Hence, this paper uses MARL to converge as data efficient as possible.

In RL there are several different updating/learning methods; policy-based, actor-critic, and value-based. This paper focuses on value-based RL, more specifically Q-learning. In Q-learning the central idea is to satisfy the Bellman optimality equation Eq. (1) [9]. In $Q^*(s, a)$ is given as follows:

$$Q^*(s, a) = \mathbb{E}[r + \gamma \max_{a'} Q^*(s^{'}, a^{'}) \mid s, a] \tag{1}$$

In Eq. (1) is Q* the optimal Q-function of the system. Q* is given by the reward at time t(r), and the discounted reward of future states. Where γ is the discount factor and $Q^*(s^{'}, a^{'})$ is the future reward. Q* is not typical known, the entire reason for doing value based RL is to learn Q*. this can be done effeciently with Artificial Neural Networks (ANN's). The Q function then looks like the following $Q^*(s, a; \theta)$ where θ is the weights of the ANN.

To learn the Q* function a back propagation trough the ANN is performed, this is done by calculating the difference between the calculated value of the Q function, and the estimated Q function. This can be done for every integration or in batches. It is typical to do so in batches[9, 10].

Eq. (1) is a single agent formulation of the Q-learning algorithm. However, as stated in the introduction, this paper formulates the environment as a MG and uses MARL. A Q-learning algorithm for an MG can be formulated with Eq. (2) [22].

$$Q^{*,m}(s, a_1, .., a_m) = \mathbb{E}[R_{t+1} + \gamma \max_{a_m^{'}} Q^{*}(s^{'}, a_1^{'}, .., a_m^{'}) \mid s, a_1, .., a_m] \tag{2}$$

In Eq. (2) $Q^{*,m}$ refers to the Q function of the m'th agent in the system. It can be seen that all m agents observe all states (s) and all actions $a_1, ..., a_m$. This ensures convergence. However, this formulation is data expensive because it does not reduce the complexity of the problem.

In [17] it is shown that by making assumptions about the environment, the Q-function can be formulated as shown in Eq. (3) and (4). This formulation can only be made because we know that it is a UFH system with a supply temperature and on/off valves.

$$Q^{st}(s^{st}, a^{st}, a^{v_{1..m}}) = \mathbb{E}\left[r^{st} + \beta \max_{a^{st}} Q^{st}(s^{'st}, a^{st}, a^{v_{1..m}})\right] \tag{3}$$

$$Q^{v_m}(s^{v_m}, a^{v_m}, a^{st}_{t-1}) = \mathbb{E}\left[r^{v_m} + \beta \max_{a^{v_m}} Q^{v_m}(s'^{v_m}, a'^{v_m}, a'^{st}_{t-1})\right] \tag{4}$$

In is Q^{st} the Q funciton for the supply temperature agent and Q^{v_m} is the Q function for the m'th valve agent. It can be seen that local states are made for each valve agent and the supply temperature agent. Furthermore can it be seen that the valve agents are not aware of current action of the supply agent, but only past actions.

An illustration of the communication structure can be seen in Figure 3. Additional can a illustration of a UFH system, with the valve and supply temperature, be seen in Figure 5 in section 3.

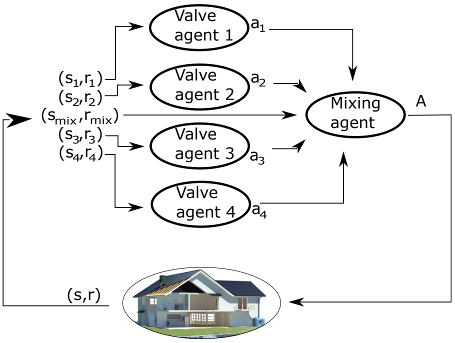


Figure 3: Illustration of how the agents interacts with each other and the environment. In the figure four valve agents, one mixing agent and a four-zone UFH system are be seen. The sequence of interactions is as follows; all valve agents choose an action based on the state of the environment, these actions are passed to the mixing agent, the mixing agent chooses an action based on the state of the environment and the actions of the valve agents. All actions are passed to the environment and the environment returns states and rewards for the agents.

From Figure 3 it can be seen that all valve agents and the mixing agent have individual reward functions and observable states. The communication structure is structured such that valve agents are communicating their actions to the mixing agent, and then the joint actions of the controllers are sent to the environment. A similar way of communicating the actions is used in [23] in a general purpose RL setting. The reward function for each agent, and the corresponding states and action are elaborated on in section 4.

2.2. Offline Training

Offline training of an RL algorithm requires a model of the real environment or data. When training from data there are multiple issues, the obvious one is a limited amount of data. If the high reward areas of the state-action space is not included in the data set, the value-function derived from the data naturally will not include these areas as well. Less obvious is how the data distribution and the shift in data distribution affects offline RL.

In supervised learning, which effectually the problem is becoming when doing offline training directly from data, the goal is to predict some state S_{t+1} from S under the same data distribution. In RL the goal is to change the policy, hence do something different, presumably better, which easily can change the data distribution[24].

Training offline directly from data with Q-learning can be done by initializing the algorithm and load the data consisting of the state action and reward transitions (s,a,r) into the replay buffer and allow the algorithm to approximate the value function [24]. This type of offline RL has been applied in [25], where the task was to enable a robot to grasp objects from a table by using image observations.

In our work we do offline training on a model, however not a pre-built and verified model, but a data-driven model. The argument for doing so, and not loading the data into the replay buffer, is that it will be possible to generate synthetic experience by applying disturbances that is not represented in the collected data. This will combat the issue of shifting data distribution. The model is however naturally associated with some uncertainty. For this reason not all possible disturbances are applied but only disturbances which to a large extend are represented in the collected data.

2.3. Black box Model Generation

Model generation can broadly be split into 3 categories: 1) Physics-based methods also refereed to as white-box, 2) black box (data-driven) methods, or 3) a combination of the two called grey-box methods [26]. In grey-box methods the overall structure is defined by physics and data is then used to fit the parameters of the model [27]. A Physics-based model requires extensive modelling work, and because the dynamics of two houses are never the same, this work is required for every building for which the model is to be used. This is not feasible. A grey-box method can be variable, however it is not commission free, it does require expert knowledge for every installation it is used in [28]. Because this paper strives to develop a model free approach a data-driven model is developed. Even though this paper uses a black box model, it can be argued that a grey-box model will be more data efficient and better at generalizing from a small amount of data, but for the reason stated above a grey-box model is not used.

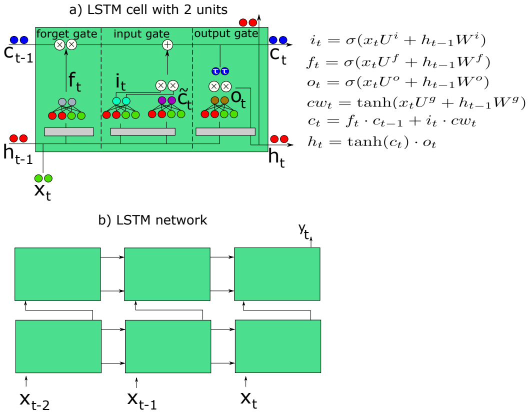
There are several different methods to build data-driven models for HVAC systems. This paper uses ANN as function approximators, which before has been deployed in black box models for HVAC systems with success [29, 30]. Because of the slow and delayed responses associated with a radiant heating system, it can be beneficial to use a Recurrent Neural Network (RNN). An RNN is a broad term for neural networks that can recognize patterns in a sequence of data[31]. In an HVAC context this is naturally time-series data. A long short term memory (LSTM) layer is a type of RNN that can identify patters over shorter or longer periods depending on the problem [32]. LSTM networks has also been used in a black box model context to predict load profiles of electricity consumption [33]. One can argue that some of the same dynamic properties, at least with respect to user behavior, are present in electricity consumption as in HVAC systems.

For the purpose of investigating if LSTM networks are suited for this task is a model with Multilayer Perceptrons (MLP) also investigated in section 4. MLP is the typical type of artificial neuron (ANN) that is used in most supervised learning methods. These are computational efficient, however they do not have the benefits of the LSTM network.

A LSTM network can have several layers each layer can then have several LSTM cells. The LSTM cells can be designed differently. The method used in this paper, is a LSTM cell with a forget gate [34]. Other methods include

Gated Recurrent Unit (GRU), LSTM without forget cell, LSTM with a Peephole Connection etc. [31].

In Figure 4 two illustrations of LSTM can be seen. In a) an LSTM cell with two units and an input size of two can be seen. In b) an example of a supervised regression model, with an LSTM network with two layers, and three time step dependencies can be seen.



$$
\begin{aligned}
i_t &= \sigma(x_t U^i + h_{t-1} W^i) \\
f_t &= \sigma(x_t U^f + h_{t-1} W^f) \\
o_t &= \sigma(x_t U^o + h_{t-1} W^o) \\
cw_t &= \tanh(x_t U^g + h_{t-1} W^g) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot cw_t \\
h_t &= \tanh(c_t) \cdot o_t
\end{aligned}
$$

Figure 4: a)illustration of LSTM cell: h is the hidden cell state, c is the cell state, x is the state from the environment, f, i cw and o are function dependencies and σ and tanh are activation functions. The mathematical expression of a LSTM cell can be seen to the right. b) Illustration of supervised regression model with a simple LSTM network.

In Figure 4 a it can be seen how a LSTM cell can be divided up into gates. In the forget gate is it calculated weather or not information from the past cell is passed to the new cell state c_t. The input gate calculated how much information from the new state x_t are included in c_t and in the output gate the hidden cell h_t is calculated.

2.4. Contributions

This paper extends the current state-of-the-art for offline RL for HVAC systems. The framework presented in this paper ensures robust behavior during deployment by using a traditional control strategy to collect data, and then build a black box model from this data. The training can then take place in the black box model environment where exploration does not affect occupants of the real-world environment. Furthermore, state of art for black box model generation for HVAC systems is expanded by testing LSTM layers and sequential layers for black box model generation for UFH systems. The following section presents the simulation environment that will serve as a test environment for this algorithm

3. Simulation and evaluation

This section elaborates on the simulation environment used in this paper and which limitation this environment has when compared to a real-world environment. The reason for doing a simulation based study is that it takes years of data to complete this study, which makes real-time tests infeasible for the tests and comparison studies presented here.

Firstly, a general UFH system in a domestic building is described. This description will help the reader to gain an understanding of how these systems work and the disturbances that affect them. Secondly,the simulation is presented and a short evaluation is made.

Figure 5 illustrates a UFH system with n zones. We simulate a 4-temperature zone system, however the dynamic is best described from a general point-of-view.

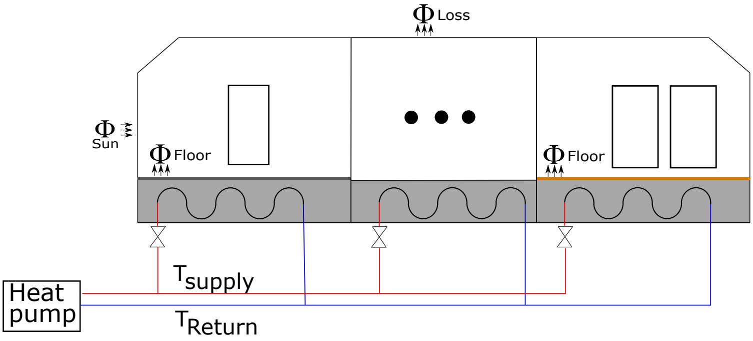


Figure 5: Illustration of a n zone underfloor heating system. From the illustration can it be seen that the heat supply is a air to water heat pump, and that flow to the individual zones are controlled by on/off valves. additionally can the heat fluxes, Φ_{Sun},Φ_{Floor} and Φ_{Loss} be seen.

,

From Figure 5 the three primary heat fluxes can be seen. These all contributes to the temperature of the zone, which ultimately is what we want to control. The objective is to keep the temperature as close to a defined reference point at the lowest cost possible.

The heat flux Φ_{floor} is controlled by the supply temperature from the heat-pump and the flow in each zone is controlled by an on/off valve. The response of Φ_{floor} is however strongly affected by two factors 1) The slow response in the concrete floors and the type of flooring, wood tiles etc. 2) The delayed response in the transportation of water from the heat-pump to the floor.

The heat fluxes Φ_{sun} and Φ_{loss} are the disturbances effecting the system. These are dependent on the window area, wall area, insulation type, roof etc.

and disturbances such as, sun, ambient temperature, rain, and wind.

3.1. Simulation Environment

The simulation environment is built in the Dymola simulation software. Dymola is a Modelica-based multi-physics simulation software, and as such suited to do simulations of complex systems and processes where there both is a hydraulic part and a thermodynamic part [35]. For Dymola, several libraries have been developed. For this simulation, the standard Modelica library and the Modelica Buildings libraries are used. The simulation environment presented in this paper is described in more details in [17].

To simulate the hydraulic part of the system the length of the pipe in each zone is defined along with the flow of water from the heat pump. Because a UFH system is built with on/off valves and not proportional valves that can regulate the flow to each zone, the UFH system is commissioned to balance the flow resistance of individual branches. This means that the pressure drop over each zone is adjusted such that the flow is 1.5L/h pr meter of pipe in each zone, hence the flow through a zone with a 100m of pipe is 150L/h.

Table 1: Parameters used in the Dymola simulation for each of the four temperature zones.

Parameter	Zone1	Zone2	Zone3	Zone4
Length of pipe	$56m$	$105m$	$42m$	$70m$
Window area	$12m^2$	$25m^2$	$12m^2$	$24m^2$
Wall area	$36m^2$	$40m^2$	$12m^2$	$30m^2$
Zone area	$16m^2$	$30m^2$	$12m^2$	$20m^2$
Zone volume	$48m^3$	$90m^3$	$36m^3$	$60m^3$

The thermodynamic side of the simulation is constructed using the base element *"ReducedOrder.RC.TwoElements"*. This element includes heat transfer from exterior walls, windows, and interior walls to the room. It furthermore includes radiation from the outside temperature and radiation from the sun. Wind and rain is not included in the simulation, as they are assessed to be smaller disturbances and are therefore not included. The element is made in accordance with "VDI 6007 Part 1" which is the European standard for calculating transient thermal response of rooms and buildings [36]. An evaluation of this simulation is made in [17].

To calculate the cost of heating with an air-to-water heat pump a model of a heat pump is developed. This model take into account the COP[37], partial load factor[38] and the cost of electricity. In Figure 6 the price of electricity as a function of time of day, the COP as a function of the ambient temperature, the supply temperature, and the partial load factor as a function of the duty cycle can be seen.

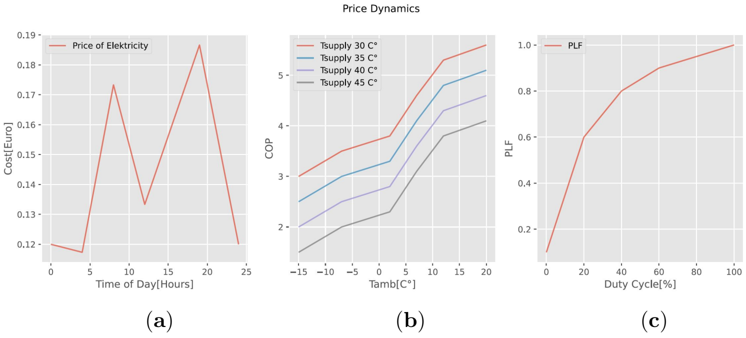


Figure 6: Dynamics of a heat pump: (**a**) Shows the average electricity prices, including taxes in Denmark as a function of the time of day (tod). (**b**) Shows the Coefficient of Performance(COP) as a function of the ambient temperature, for four different supply temperatures. (**c**) Shows the Partial Load Factor(PLF) as a function of the duty cycle (D).

With the Cost of Electricity (CE), the COP and the PLF described and the power consumption of the system (ΔE) available from the simulation, the cost of heating with a heat pump can be simulated with Eq. (5):

$$cost = \frac{\Delta E}{COP(T_{amb}, T_{supply}) \cdot PLF(D)} \cdot CE(tod). \tag{5}$$

The following section is elaborating on how the black box models are designed, lastly the black box models are tested and evaluated in the simulation environment described above.

4. Design and test of black box model

In this section the black box model, that will be used for offline training, is presented. Firstly, the requirements of the model are presented, followed by the limitations and design of the model, and lastly a test of the model.

The episode length for the RL algorithm is defined as 30 time-steps or 5 hours, hence the model to useful is required to predict the room temperature 30 time steps into the future. Because this is a control task, is it necessary for the model to predict every time step in between the current time and 30 time steps into the future dependent on which control actions is performed. An illustration of a system like this can be seen in Figure 7, the reason why there is a different model for each time step is to compensate for the unavoidable error that will occur in each model, for this reason is a model made for predicting each of the 30 time steps. Alternatively to a model for each time-step, can this also be done with a single model that is used in all time-steps, this can be visualized in Figure 7 by replacing the 30 different models with the same model for all predictions. The performance of a 30 model architecture and a single model architecture is also investigated.

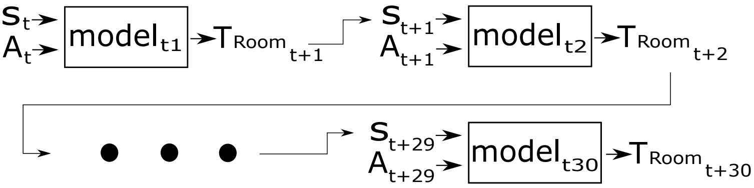


Figure 7: Illustration of model architecture for supervised learning learning. This architecture will compensate for the prediction error that occurs in every time step. A new model is made for all predictions meaning that for 30 time steps, 30 models are made.

From Figure 7 it can be seen that the problem is a regression problem, where a model is predicting the room temperature from the state of the system and the control action performed. The states of the system and the control actions can be seen in Table 2

Table 2: Table showing states and actions used in the model described above for a single zone UFH system.

States	Actions
Room temperature $[t, \cdot\cdot, t_{-6}]$	Supply temperature
Ambient temperature $[t, \cdot\cdot, t_{-6}]$	Valve position
Sun $[t, \cdot\cdot, t_{-6}]$	
Ambient Temperature forecast $[t, \cdot\cdot, t_{-6}]$	
Sun forecast $[t, \cdot\cdot, t_{-6}]$	
Time of day $[t, \cdot\cdot, t_{-6}]$	

As it can be seen in Table 2 is the state-space for the model, not only the current state at time t, but also 6 time steps back. This has to do with the slow and delayed responses of the UFH system that was explained in section 3.

To reduce the complexity of the model it is assumed that the different zones have no hydraulic or thermodynamic effect on each other, this means one model can be made for each zone and then the four models can be combined into the UFH environment that is being simulated. These assumptions are not true for the actual simulation model, or a real-world application. However, the goal of the black box model is not necessarily to converge 100%, but rather to be as data efficient as possible, and therefore this tradeoff between accuracy and complexity is sensible.

4.1. Test of Black box models

The test data is only presented for a single temperature zone. Because of the limitation presented in the section above this is sufficient. Two algorithms will be tested, one with an LSTM layer as presented in section 2 and one with a MLP network. The data foundation is 280 days, equivalent to one heating season of data. The data is split into training and testing data, 60 days is used for training and 220 days is used for testing. Normally, a 70/30 % split would be used, where most of the data is used for training. However, in this paper we want to show that we can perform well with smaller amounts of data, hence the reason for splitting the data so we only training on 20% and validating on 80%.

The hyper-parameters for the two algorithms are shown in Table 3. These have been found by empirical tests.

Table 3: Hyperparameters for the LSTM model and MLP model.

	LSTM model	MLP
Optimizer	Adam	Adam
Activation functions	ReLU	ReLU
Learning rate	0.0005	0.0005
Hidden Layers	1	1
Hidden neurans	64	64
Input layer	8x6	48
Output layer	1	1

Four tests are carried out. The prediction error for each model is presented in Figure 8.

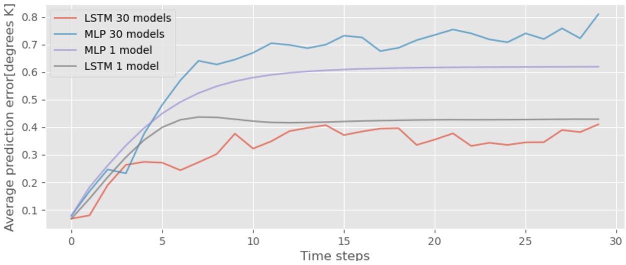


Figure 8: Plot of the average prediction error for each time step. 4 plots can be seen, one for the 1 model LSTM, one for the 30 models LSTM and the same for the MLP test. The data foundation is 60 days for the training of the model and 220 days for the evaluation.

Table 4: The average error pr. prediction, 30 time steps into the future, under the traditional control policy.($\frac{\sum error}{30}$)

	LSTM model	MLP
1 model	0.3894	0.6946
30 models	0.3246	0.6166

From Figure 8 and Table 4 it can be seen that the prediction error is 45% lower for the LSTM based model then the MLP model. Furthermore, it can be seen that by making a model for each time step and thereby compensating for prediction error the models perform 19% better on average.

The framework for the interaction between the RL algorithm, the real-world environment and the black box environment is illustrated in Figure 1. The pseudo code for this framework can be seen in Algorithm 1.

Algorithm 1 RL/Black box framework

```
if Scenario A == True then
    s=Initialize environment
    for I iterations do
        calculate actions based on a Traditional Control Policy.
        Perform calculated actions in the Real-world environment.
        Store states and actions $(s_t^n, a_t^n)$ in buffers $\mathcal{D}$.
    end for
else if  Scenario B == True then
    Store available data in buffer $\mathcal{D}$
end if
Build Black box models from available data in buffer $(\mathcal{D})$
for N Iterations do
    Calculate actions based on a MARL Control Policy.
    Perform calculated actions in the black box model environment
    Update RL Control Policy
end for
for Inf Interactions do
    Calculate actions based on a MARL Control Policy.
    Perform calculated actions in Real-World environment
    Update RL Control Policy
end for
```

The MARL algorithm referred to in the pseudo code above is developed and described in detail in [17]. The theory supporting the MARL algorithm is elaborated on in section 2. The hyper-parameters, input values, and the reward functions used in the MARL are the same as used in [17]. However, these are repeated for the convenience of the reader in the following section.

4.2. MARL dependencies and sub functions

In the following we outline the Reward functions and hyperparameters used in the MARL algorithm. The reward function for the valve agents can be seen in Eq. (6), with the two sub-functions in Eq. (7) and Eq. (8).

$$\mathrm{R}(T_z, V, H_c) = \begin{cases} 2 - (T_z - T_{ref})\cdot & \text{if } 21.6 < T_z < 22 \\ -(T_z - T_{ref}) & \text{if } 21.6 > T_z \text{ or } T_z > 22 \\ -H_c & \text{if } SC = active \end{cases}, \tag{6}$$

$$\mathrm{SC}(T_z, V) = \begin{cases} \text{not active} & \text{if } 21 < T_z > 23 \\ \text{active} & \text{if } T_z < 21 \text{ and } VP = 0 \\ \text{active} & \text{if } T_z < 23 \text{ and } VP = 1 \end{cases}, \tag{7}$$

$$H_c\ (\mathrm{SC}) = \begin{cases} 1 + H_C & \text{if SC = active} \\ 5 & \text{if SC = not active} \end{cases} \tag{8}$$

The abbreviations in the equations above are the following: R = Reward SC = Safety controller, T_z = Zone temperature, VP = Valve position, and H_c = Hard constraint.

The two sub-functions (7) and (8) are parts of the safety controller and ensure a robust behavior, incorporating a safety controller for this type of control task is supported in [39]. In [39] it is found that by incorporating a safety controller is robust behavior ensured, and a reduced convergence time is archived by reducing the action/state space to what is known to be feasible.

Similar to the reward function for the valve agents can the reward function for the supply be seen in Eq. (9) with similar sub-functions Eq. (10) and Eq. (11).

$$R(T_z, V, H_c) = \begin{cases} 2 - (T_z - T_{ref}) - P & \text{if } 21.6 < T_z < 22 \text{ and } VP = 1 \\ -(T_z - T_{ref}) - P & \text{if } 21.6 > T_z \text{ or } T_z > 22 \end{cases} \tag{9}$$

$$SC(T_z, V) = \begin{cases} \text{not active} & \text{if } T_z, > 20.5 \\ \text{active} & \text{if } T_z, < 20.5 \text{ and VP = 1} \end{cases} \tag{10}$$

$$H_c(SC) = \begin{cases} 1 + H_C & \text{if SC = active} \\ 5 & \text{if SC = not active} \end{cases} \tag{11}$$

The hyperparameters for both the supply agent and the valve agent can be seen in Table 5. From Table 5 it can be seen that it is the same hyperparameters used in the supply agent and valve agents.

Table 5: Hyperparameters used for training the agents.

	Supply Agent	Valve Agent
Learning rate	0.01	0.01
Epsilon decay	0.0005	0.0005
Epsilon max	1	1
Epsilon min	0.1	0.1
batch size	432	432
N_steps	45	45
gamma	0.9	0.9
ANN	60 × 60 × 60	60 × 60 × 60
Target update rate	540	540

The following section, present the results of the framework.

5. Simulation Results

This section presents four simulations, two simulations with the RL/black box framework, one simulation only with the RL algorithm, and one simulation with a traditional controller. The four simulations are outlined below.

- Simulation 1: without RL/black box framework but with RL control. This simulation will serve as benchmark for how the RL performs without training in the black box model environment.
- Simulation 2: with a traditional control policy, this will serve as a benchmark to estimate the RL algorithms capability to reduce heating costs while maintaining or increasing the comfort level.
- Simulation 3: with RL/black box framework, in scenario A.
- Simulation 4: with RL/black box framework, in scenario B with one heating season of data(280 days).

In Figure 9 the reward plot for simulation 1, Simulation 2 and Simulation 3 is shown.

From Figure 9 it can be seen that when using the RL/black box framework the performance is improved or equal to the normal MARL controller. Especially during the first 60 days the performance is better. The reason for the improvement in this phase is that the RL/black box framework follows the traditional control policy. After approximately 580 days the MARL and RL/black box framework converge to approximately the same control policy.

In Figure 10 the results of simulation 1, Simulation 3 and Simulation 4 are shown.

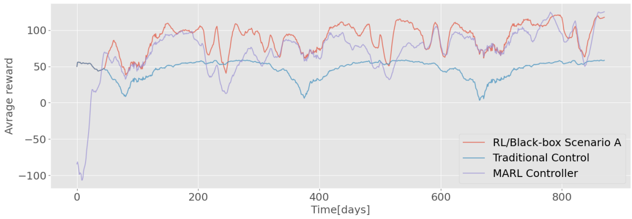


Figure 9: Reward plot over 880 days for simulation 1, Simulation 2 and Simulation 3, where 1 is; MARL control, without the RL/black box framework, and 2 is; A traditional control policy. and 3 is; the RL/black box frame work in Scenario A.

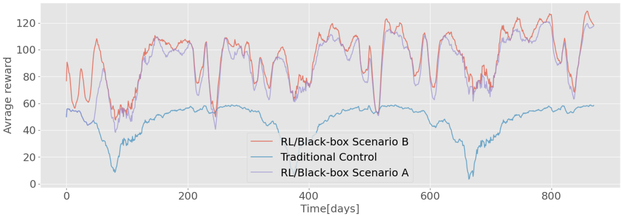


Figure 10: Reward plot over 880 days for simulation 1, Simulation 3 and Simulation 4, where 1 is; MARL control, without the RL/black box framework, and 3 is; A traditional control policy. and 4 is; the RL/black box frame work in Scenario B.

From Figure 10 it can be seen that the RL/black box framework does perform better when more data is available. During the first period of 60 days are the performance of scenario B notably better. After this period is the increase in performance only a marginal. The reason for only a marginal increase, is that the generated black box models do not become much better with the additional data. This is discussed further in section 6.

To assess if the RL algorithms are performing better than a traditional controller, the cost and comfort level are investigated. To analysis the comfort a box-plot of the temperature distribution is made for each of the four zones in the simulation.

From Figure 11 it can be seen that the variation in temperature is smaller or similar when comparing MARL/Scenario A with traditional control. In the box plots it can be seen that the median is approximately 0.2 C°lower in zone 2 and zone 4 and 0.1 C°lower in zone 1 and 3. This deviation from the reference

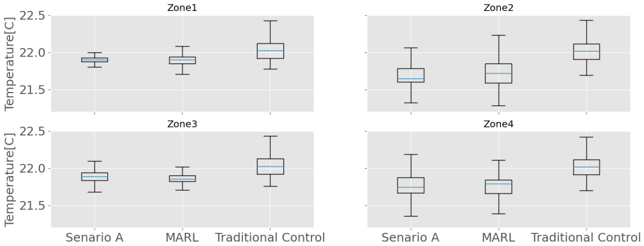


Figure 11: Plots for each of the four temperature zones. In each plot the temperature distribution for Scenario A, MARL and Traditional Control are plotted. The data foundation is the entire simulation period of 880 days.

temperature of 22 C°is according to the reward functions negligible. When comparing MARL to Scenario A it can be seen that the performance is similar. This is to be expected since they converge towards the same control policy. In Figure 12 the temperature distribution for the first 100 days can be seen. From this it can be seen that the variation is higher for the MARL agents without the offline training framework.

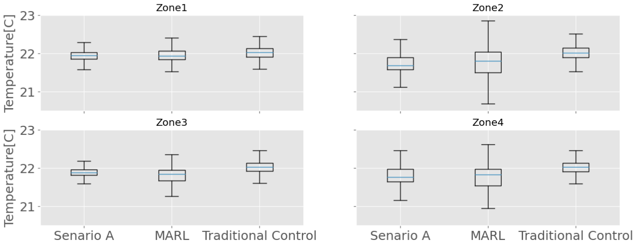


Figure 12: Plots for each of the four temperature zones. In each plot the temperature distribution for Scenario A, MARL and Traditional Control is plotted. The data foundation is the first 100 days of the simulations.

Lastly the energy consumption for the 4 simulations is evaluated. The results can be seen in Table 6.

From Table 6 it can be seen that each of the four simulations uses approximately the same amount of heat energy. However, when evaluating the electric energy consumption it can be seen that the RL-based controllers are performing

Table 6: Cost of heating for each of the four simulations over the entire simulation period of 880 days. Additional to the cost can the consumed Heat energy and the electric energy, the average Coefficient of performance(COP) and the average partial load factor(PLF) be seen.

Test	Heat energy	Electric Energy	Avg. COP	Avg. PLF	Cost	Savings
TC	43.1 MWh	15.7 MWh	2.81	0.82	21854 DKK	0.0%
MARL	42.8 MWh	12.2 MWh	3.54	0.94	18139 DKK	17.1%
Scenario A	42.9 MWh	11.9 MWh	3.61	0.95	17943 DKK	17.9%
Scenario B	42.7 MWh	11.7 MWh	3.68	0.98	17615 DKK	19.4%

significantly better. Scenario B is saving 19.4 % when comparing to traditional control. Scenario B performs better then both Scenario A and MARL. It has, however, been established that they all converge to the same control policy. Therefore, this better performance will over time also become smaller. Over a 30-year lifespan this will most likely become close to zero.

6. Discussion

The similar performance of Scenario A and Scenario B is not given. We did expect the performance of scenario B to be significantly better than scenario A. However, after examining the distribution of the data of which the black box model was made this makes sense.

In Figure 13 histograms of the data distribution for the black box models for scenario A and scenario B can be seen.

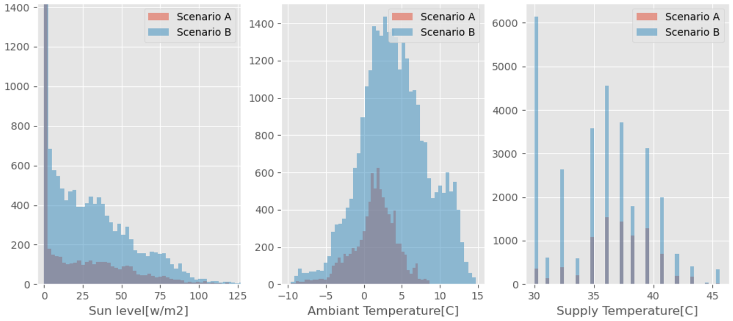


Figure 13: Data foundation for black box models in Scenario A and Scenario B. Starting from the right is the Sun level, then the ambient temperature and lastly the supply temperature.

From Figure 13 it can be seen that even though there is more data in the black box models of scenario B it is close to the same distribution. This is the reason for the similar performance.

All the simulations are initiated on January 1st. One can argue that this is a good time for collecting data, and that 60 days of data therefore might not be enough if the data is collected during the Spring and Summer months. Further research will establish if it is possible to estimate if a black box model will be good or not based on the data distribution rather than the amount of data.

7. Conclusion

This paper presents a novel framework for offline RL with online fine tuning for HVAC systems. The contribution of this paper is that by doing offline RL poor behavior during early training can be eliminated. The online fine tuning will allow the agent to converge better because all dynamics can not be model in a black box model environment. It is additionally showed that this framework can be used in retrofit situations where existing data from a building management system can be used.

It is shown in a simulation environment that poor behavior can be eliminated completely in both recommissioning task and in new commissioning. Furthermore is it shown that a cost reduction of 17.9 and 19.4% for new building and old buildings with data is achieved in this simulation environment.

The black box model generation that is made in this paper is done with LSTM networks. The performance of the LSTM networks is compared to MLP networks and it was found that LSTM improve performance by 50% when comparing to MLP networks. Additionally is different types of architectures tested, it is found that by creating a model for each time-step into the future can the average prediction error be reduced by 17%.

8. Future Work

This paper present a method for doing RL based control of HVAC system where poor behavior during early training is limited to the current state of art controllers. However we note two things that still needs validation.

- A real world test where it is validated that this algorithm is able to compensate for building dynamics and weather disturbances.
- A large simulation study which include occupant disturbance, to verify that these disturbances do not cause the algorithm to fail. This should be followed by a field test demonstrating the algorithm in houses with occupants.

Another matter that has not been addressed is the computational capacity required, to make control schemes like the one described in this paper. It can give cause for concern that a framework like the one described above is from a computational point of view magnitudes more complex then current state of art solutions. A internet connection and cloud computing may be a solution to this otherwise, steps must be take to reduce the complexity of framework presented in this paper.

References

[1] L. Pérez-Lombard, J. Ortiz, C. Pout, A review on buildings energy consumption information, Energy and Buildings 40 (3) (2008) 394–398. doi:https://doi.org/10.1016/j.enbuild.2007.03.007.
URL https://www.sciencedirect.com/science/article/pii/S0378778807001016

[2] M. Akmal, B. Fox, Modelling and simulation of underfloor heating system supplied from heat pump, in: 2016 UKSim-AMSS 18th International Conference on Computer Modelling and Simulation (UKSim), 2016, pp. 246–251. doi:10.1109/UKSim.2016.13.

[3] S. Prívara, J. Široký, L. Ferkl, J. Cigler, Model predictive control of a building heating system: The first experience, Energy and Buildings 43 (2) (2011) 564–572. doi:https://doi.org/10.1016/j.enbuild.2010.10.022.
URL https://www.sciencedirect.com/science/article/pii/S0378778810003749

[4] H. Huang, L. Chen, E. Hu, A new model predictive control scheme for energy and cost savings in commercial buildings: An airport terminal building case study, Building and environment 89 (2015) 203–216.

[5] A. Afram, F. Janabi-Sharifi, Theory and applications of hvac control systems – a review of model predictive control (mpc), Building and Environment 72 (2014) 343–355. doi:https://doi.org/10.1016/j.buildenv.2013.11.016.

[6] K. M. Tsui, S. C. Chan, Demand response optimization for smart home scheduling under real-time pricing, IEEE Transactions on Smart Grid 3 (4) (2012) 1812–1821. doi:10.1109/TSG.2012.2218835.

[7] L. Yu, T. Jiang, Y. Zou, Online energy management for a sustainable smart home with an hvac load and random occupancy, IEEE Transactions on Smart Grid 10 (2) (2019) 1646–1659. doi:10.1109/TSG.2017.2775209.

[8] E. Mills, N. Bourassa, M. Piette, H. Friedman, T. Haasl, T. Powell, D. Claridge, The cost-effectiveness of commissioning new and existing commercial buildings: Lessons from 224 buildings, HPAC Engineering (11 2005).

[9] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[10] 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, D. Hassabis, Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.
URL http://dx.doi.org/10.1038/nature14236

[11] E. Barrett, S. P. Linder, Autonomous hvac control, a reinforcement learning approach, in: Joint European conference on machine learning and knowledge discovery in databases, Vol. 9286, 2015. doi:10.1007/978-3-319-23461-8_1.

[12] A. Overgaard, B. K. Nielsen, C. S. Kallesøe, J. D. Bendtsen, Reinforcement learning for mixing loop control with flow variable eligibility trace, in: 2019 IEEE Conference on Control Technology and Applications (CCTA), 2019, pp. 1043–1048. doi:10.1109/CCTA.2019.8920398.

[13] T. Wei, Y. Wang, Q. Zhu, Deep reinforcement learning for building hvac control, in: 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), 2017, pp. 1–6. doi:10.1145/3061639.3062224.

[14] C. Blad, S. Koch, S. Ganeswarathas, C. Kallesøe, S. Bøgh, Control of hvac-systems with slow thermodynamic using reinforcement learning, Procedia Manufacturing 38 (2019) 1308–1315, 29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2019), June 24-28, 2019, Limerick, Ireland, Beyond Industry 4.0: Industrial Advances, Engineering Education and Intelligent Manufacturing. doi:https://doi.org/10.1016/j.promfg.2020.01.159.
URL https://www.sciencedirect.com/science/article/pii/S2351978920301608

[15] H. Kazmi, J. Suykens, A. Balint, J. Driesen, Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads, Applied Energy 238 (2019) 1022–1035. doi:https://doi.org/10.1016/j.apenergy.2019.01.140.
URL https://www.sciencedirect.com/science/article/pii/S0306261919301564

[16] L. Yu, Y. Sun, Z. Xu, C. Shen, D. Yue, T. Jiang, X. Guan, Multi-agent deep reinforcement learning for hvac control in commercial buildings, IEEE Transactions on Smart Grid PP (2020) 1–1. doi:10.1109/TSG.2020.3011739.

[17] C. Blad, S. Bøgh, C. Kallesøe, A multi-agent reinforcement learning approach to price and comfort optimization in hvac-systems, Energies 14 (22) (2021). doi:10.3390/en14227491.

[18] T. Wei, Y. Wang, Q. Zhu, Deep reinforcement learning for building hvac control, in: 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), 2017, pp. 1–6. doi:10.1145/3061639.3062224.

[19] S. Levine, Decisions from data: How offline reinforcement learning will change how we use machine learning, [https://medium.com/@sergey.levine/decisions-from-data-how-offline-reinforcement-learning-will-change-how-we-use-ml-24d98cb069b0] (Sebtember 2020).

[20] C. Zhang, S. R. Kuppannagari, R. Kannan, V. K. Prasanna, Building hvac scheduling using reinforcement learning via neural network based model approximation, in: Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, BuildSys '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 287–296. doi:10.1145/3360322.3360861.

[21] R. Bellman, Dynamic programming, Science 153 (3731) (1966) 34–37.

[22] L. Busoniu, R. Babuska, B. De Schutter, Multi-agent Reinforcement Learning: An Overview, Vol. 310, 2010, pp. 183–221. doi:10.1007/978-3-642-14435-6_7.

[23] D. Bertsekas, Multiagent reinforcement learning: Rollout and policy iteration, IEEE/CAA Journal of Automatica Sinica 8 (2) (2021) 249–272. doi:10.1109/JAS.2021.1003814.

[24] S. Levine, A. Kumar, G. Tucker, J. Fu, Offline reinforcement learning: Tutorial, review, and perspectives on open problems (2020). arXiv:2005.01643.

[25] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, S. Levine, Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation (2018). arXiv:1806.10293.

[26] R. Venugopalan, R. Ideker, Chapter ii.5.10 - bioelectrodes, in: B. D. Ratner, A. S. Hoffman, F. J. Schoen, J. E. Lemons (Eds.), Biomaterials Science (Third Edition), third edition Edition, Academic Press, 2013, pp. 957–966. doi:https://doi.org/10.1016/B978-0-08-087780-8.00082-6.
URL https://www.sciencedirect.com/science/article/pii/B9780080877808000826

[27] A. Afram, F. Janabi-Sharifi, Review of modeling methods for hvac systems, Applied Thermal Engineering 67 (1) (2014) 507–519. doi:https://doi.org/10.1016/j.applthermaleng.2014.03.055.

[28] A. Afram, F. Janabi-Sharifi, Gray-box modeling and validation of residential hvac system for control system design, Applied Energy 137 (2015) 134–150. doi:https://doi.org/10.1016/j.apenergy.2014.10.026.

[29] A. Afram, F. Janabi-Sharifi, Black-box modeling of residential hvac system and comparison of gray-box and black-box modeling methods, Energy and Buildings 94 (2015) 121–149. doi:https://doi.org/10.1016/j.enbuild.2015.02.045.

[30] A. Kusiak, G. Xu, Z. Zhang, Minimization of energy consumption in hvac systems with data-driven models and an interior-point method, Energy Conversion and Management 85 (2014) 146–153. doi:https://doi.org/10.1016/j.enconman.2014.05.053.

[31] Y. Yu, X. Si, C. Hu, J. Zhang, A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures, Neural Computation 31 (7) (2019) 1235–1270. doi:10.1162/neco_a_01199.

[32] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–80. doi:10.1162/neco.1997.9.8.1735.

[33] H. Shi, M. Xu, R. Li, Deep learning for household load forecasting—a novel pooling deep rnn, IEEE Transactions on Smart Grid 9 (5) (2018) 5271–5280. doi:10.1109/TSG.2017.2686012.

[34] F. A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with lstm, Neural Computation 12 (10) (2000) 2451–2471. doi:10.1162/089976600300015015.

[35] D. Brück, H. Elmqvist, S. Mattsson, H. Olsson, Dymola for multi-engineering modeling and simulation, Deutsches Zentrum fur Luft- und Raumfahrt e.V. (DLR) (2002) 1–8.

[36] M. Wetter, W. Zuo, T. Nouidui, X. Pang, Modelica buildings library, Journal of Building Performance Simulation 7 (07 2014). doi:10.1080/19401493.2013.765506.

[37] J. Nie, Z. Li, X. Kong, D. Li, Analysis and comparison study on different hfc refrigerants for space heating air source heat pump in rural residential buildings of north china, Procedia Engineering 205 (2017) 1201–1206, 10th International Symposium on Heating, Ventilation and Air Conditioning, ISHVAC2017, 19-22 October 2017, Jinan, China. doi:https://doi.org/10.1016/j.proeng.2017.10.354.
URL https://www.sciencedirect.com/science/article/pii/S1877705817350294

[38] K. Piechurski, M. Szulgowska-Zgrzywa, J. Danielewicz, The impact of the work under partial load on the energy efficiency of an air-to-water heat pump, E3S Web of Conferences 17 (2017) 00072. doi:10.1051/e3sconf/20171700072.

[39] C. Blad, C. S. Kallesøe, S. Bøgh, Control of hvac-systems using reinforcement learning with hysteresis and tolerance control, in: 2020 IEEE/SICE International Symposium on System Integration (SII), 2020, pp. 938–942. doi:10.1109/SII46433.2020.9026189.

Paper E

PaperE

Pre-Print
A Field test of Offline Multi Agent Reinforcement Learning for HVAC-Systems

C. Blad[a], S. Bøgh[b], C. Kallesøe[c], Paul Raftery[d]

[a]*Grundfos A/S, Poul Due Jensens Vej 7,Bjerringbro, DK-8850 Denmark and Robotics & Automation Group, Dept. of Materials and Production, Aalborg University, Fibigerstraede 16, Aalborg Øst, DK-9220, Denmark*[1]

[b]*Robotics & Automation Group, Dept. of Materials and Production, Aalborg University, Fibigerstraede 16, Aalborg Øst, DK-9220, Denmark*

[c]*Grundfos A/S, Poul Due Jensens Vej 7,Bjerringbro, DK-8850 Denmark and Dept. of Electronic systems, Aalborg Unicersity, Fredrik Bajersvej 7, Aalborg Øst, DK-9220, Denmark*

[d]*The Center for the Built Environment, Berkeley University of California, 390 Wurster Hall Berkeley, USA*

Abstract

This paper presents a field study of offline Reinforcement Learning (RL) control of Heating Ventilation and Air-Conditioning (HVAC) system. More specifically is the field test a UFH system, consisting of two temperature zones located in Denmark. The algorithm tested is presented in previous work, and is summarized in this paper. Firstly, is a benchmarking test presented, this test has been conducted during spring 2021 and winter 2021/2022. This data is used in the offline RL framework to train and deploy the RL policy. The RL policy is tested during winter 2021/2022 and spring 2022. An analysis of the data shows that the RL policy showed predictive control-like behavior, and reduced the oscillations of the system by a minimum of 40%. Additionally, it is argued that the RL policy is minimum 10% more cost effective than the traditional control policy used in the benchmarking test.

1. Introduction

Heating Ventilation and Air-conditions(HVAC) systems are today consuming about 40% of the annual energy consumption in the US, and this is assumed to be true for much of the western world[1]. There are multiple ways of making these systems more efficient, one of them being improving the control algorithms. Traditionally, control systems for HVAC systems are event based controllers, typically based on; the temperature of the zone (hysteresis control), the

[1] Corresponding author.
Email address: cblad@m-tech.aau.dk

ambient temperature(outside compensated supply temperature) and the time of day(scheduling)[2].

Event based controllers do not allow for any predictive control, and because of the delayed and slow responses associated with HVAC and especially radiant heating or cooling this is not optimal, furthermore is the cost of energy and the efficiency not constant, for compressor systems the efficiency depends on the ambient temperature, the part load factor, and the energy prices. Hence is the price of heating highly dependent on what happens in the future[2].

A common method to do predictive control is Model Predictive Control(MPC) and there has been made quite a few papers on MPC's for HVAC systems [3, 4, 5]. When doing MPC, a model is required, and not 2 buildings are alike. Furthermore, the dynamic of a building can also change over its lifetime. For these reasons MPC controllers for buildings are both expensive to make and can also be expensive to maintain. Other smart controllers are scheduling energy usage according to energy prices [6, 7], these controllers naturally also need a model to predictive energy usage, and are therefore like the MPC controller expensive to commission.

An expensive commission phase gives cause for concern. A study of 150 existing commercial buildings showed that a recommissioning could reduce the energy consummation by 15% on average[8]. Model free Reinforcement Learning is as the name suggests a model free method to do predictive control [9], and hence do not require commissioning, or at least very little. Numerous papers on using RL in HVAC systems have been made [10, 11, 12, 13], these papers shows that an RL algorithm compared to a traditional control policy can reduce cost between 5.5% and 15%. From the papers, it can also be seen that the problem with using RL is that it requires time/data to converge towards an optimal solution.

When using model free RL, is there always a training period, this period can as mentioned be long. But regardless of the length of the training period is it important to ensure some sort of robustness/safety. In [14] is safety for RL in general surveyed. In [15] is a framework for robust RL for HVAC system developed. This framework is elaborated on in Section 2.

To overcome slow convergence Multi Agent Reinforcement Learning for HVAC systems has been proposed in [16, 17, 18]. In MARL the environment is formulated as a Markov Game which reduces the complexity of the action space. In [18] are additional steps taken to reduce the complexity of the action state space, and hence reduce the convergence time.

This paper proposes a model-free offline MARL algorithm to the problem of poor behavior during early training of the RL agent under an assumption that a traditional controller is accessible, that can perform acceptably.

Offline model-based RL/MARL has been applied on HVAC systems in [19]. However, is this based on an extensive model, and hence as expensive to commission as an MPC controller. Offline RL for HVAC systems based on available data has been proposed in [20], but the idea has not been developed. In [21] are model based RL used in an online fashion to control airflow. The model in [21] is a grey-box model, so based on an actual model but the parameters are approximated by an artificial neural network (ANN), and hence more generic. However, can it only model the dynamics of the model, which is a limitation when talking user behavior ect.

2. Methodology

As described in Section 1 is this a field study of the RL algorithm developed in [15, 18]. The methodology described in this section is therefore supported by the above mentioned papers.

The 3 Methods that are validated in this field test are the following:

- A robustness framework that ensures robust behavior regardless of the RL agent's current policy.
- A Multi Agent RL framework that reduces convergence time.
- An offline RL framework that allows for data efficient offline training.

Before describing the method is a short introduction to RL based control for HVAC made.

RL based control of HVAC systems

RL is as described in Section 1 a model-free learning method. RL can be divided into value based, policy based and actor-critic based learning. This paper only focuses on value based learning, more specifically Q learning.

Q-learning is based on the Bellman equation seen in Eq. 1 [9].

$$Q^*(s,a) = \mathbb{E}[R_{t+1} + \gamma \max_{a^{'}} Q^*(s^{'}, a^{'})] \tag{1}$$

The bellman equation simply states that if the further states for all actions is known, the optimal policy is to choose the action that results in the highest Q value (Q*(s,a)). The Q function can also be referred to as the action-value function, because it maps the reward value into the action state space. The Reward function is for HVAC systems typically a trade-off between thermal comfort and price, see eq (2)

$$R = \alpha \cdot Comfort + \beta \cdot Price \tag{2}$$

We of cause never know Q* therefore we iteratively update the estimate of Q* by interacticing with the environment. This is illustrated in Figure 1

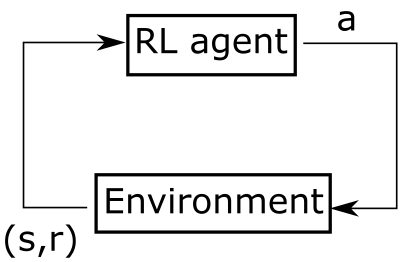


Figure 1: .

From the description above can it be derived that a RL controller can perform unwanted control actions, and therefore is robustness an important feature in online RL based control of HVAC. Additionally, will the training time increase as the complexity of the action state space, therefore is a method for reducing this also an important feature. Lastly, will offline training make it possible to deploy near optimal control policies which again will make RL for HVAC systems an appealing technology.

Robustness framework

This method is developed in [15]. In that paper it is shown that by limiting the action state space to what is known it feasible can the training time be reduced, and robustness can be ensured.

Robustness is accomplished by designing a framework where a primary controller (the RL control) is active if the system is in a predetermined part of the action state space, if the system however enters another part of the action state space, the secondary controller takes over. The secondary controller is a standard industrial control policy that ensures robust behavior. An illustration of this can be seen in Figure Figure 2.

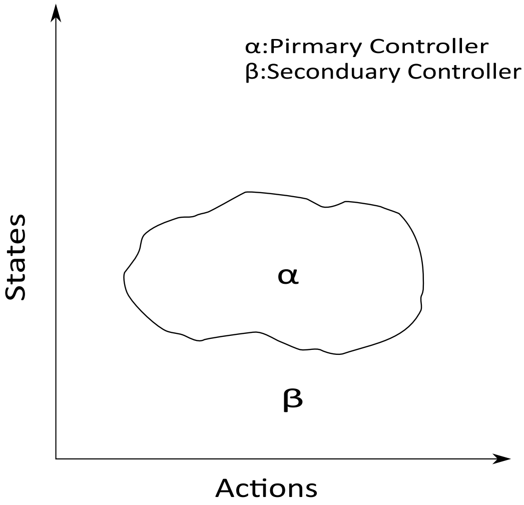


Figure 2: Illustration of robustness framework for RL. Were α represent the part of action state space where the RL control policy is in control and β represent the rest of the action sate space where the traditional control policy is in control.

Regardless of the configuration, this method ensures robustness. However, to ensure convergence for the primary control so the secondary controller over time becomes redundant is it necessary to implement some soft constraints in the objective function of the primary controller. In [15] is it shown that implementing linear increasing negative rewards yield the best results.

The results presented in [15] shows that by implementing linear increasing negative rewards are the convergence time reduced by 75% when comparing to a RL controller without the robustness framework.

MARL framework

The framework used in this field test is presented in [18]. In that paper is a MARL framework for HVAC systems presented. The purpose of using a MARL framework is to mitigate the long convergence time that RL usually is associated with.

The framework contains two types of agents. 1) a supply agent that controls the supply temperature. 2) valve agents that control the flow through each temperature zone. As described in Section 1 and further explained in Section section 3 is valve position and supply temperature the two actuator signals needed for a UFH system.

This Framework is based on Q-learning, which as explained is a value base RL learning method. The Q functions for the valve and supply agents can be seen in Eq. (3) and Eq. (4).

$$Q^{st}_{t+1}(s_{st}, a^{st}, a^{v_1..m}) = \mathbb{E}_{s,a,r,s'}$$
$$\left[(1-\alpha)Q^{i}_{t}((a^{st}, a^{v_1..m}) + \alpha[r^{i}_{t} + \beta \max_{a^{st}} Q^{st}_{t}(s^{'st}, a^{st}, a^{v_1..m})]\right] \tag{3}$$

$$Q^{v_m}_{t+1}(s^{v_m}, a^{st}, a^{v_m}) =$$
$$\mathbb{E}_{s,a,r,s'}\left[(1-\alpha)Q^{i}_{t}((s^{v_m}, a^{st}, a^{v_m}) + \alpha[r^{i}_{t} + \beta \max_{v_m} Q^{v_m}_{t}(s^{'v_m}, a^{st}, a^{v_m})]\right]. \tag{4}$$

From Eq. (3) and Eq. (4) can it be seen that all agents receive partial state spaces and that the valve agents are not aware of the action of the mixing agent. The communication structure is setup in this manner to avoid causality issues and to reduce the state space of the different agent. An illustration of how the communication is set up in the MARL framework can be seen in Figure Figure 3.

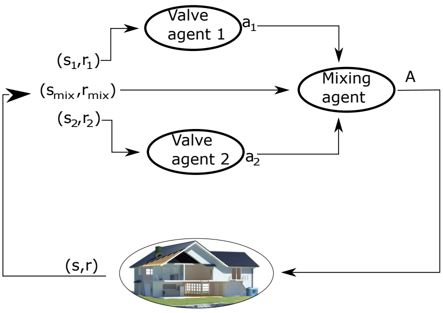


Figure 3: Illustration of two zone MARL control structure. In the illustration can it be seen that the valve action is passed to the mixing agent.

In [18] is it shown that this framework reduces the convergence time by 70% when comparing to a Single Agent RL (SARL) formulation. Additionally, is it shown that a SARL/MARL policy reduces the cost of heating by 17% when comparing to an industrial SOA control policy while maintaining or improving the thermal comfort. This framework is also supported by the robustness framework.

Offline RL framework

This framework is presented in [Paper D]. this work builds on top of the MARL framework presented above. The purpose of this framework is to make it possible to train the MARL algorithm offline from stored data. By training offline can a near optimal control policy be deployed in buildings that already have data available. If data is not available can data be collected online with an industrial SOA control policy and after a period can a near optimal control policy be deployed. An illustration of how this framework works can be seen in Figure 4.

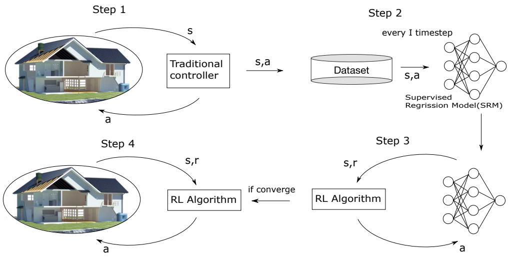


Figure 4: [Paper D]The traditional controller interacts with the environment T time steps, after each time step the action and state transition is saved in the data buffer. The data set is passed to the SRM ones trained, the SRM is used as a artificial environment for the RL agent to train until convergence. The trained agent is then deployed in the real environment, the agent can still do limited exploration for fine tuning. Step 2, 3 and 4 will be repeated until the SRM converges..

From Figure 4 can the scenario where no building data is available be seen. Step 1 is not required if data already is available. In Step 2 can the Supervised Regression Model be seen. This model is used in Step 3 as a simulation environment for the MARL algorithm to learn a near optimal policy in Step 4 is online fine tuning carried out in the real world environment, this is when the optimal policy is found.

From a technical aspect is the SRM model the most challenging part of this framework. Because this work strives to be commission free, is the SRM model a black-box model. This means that unlike white box or grey box models does this model not require any commissioning when applying it to different types of HVAC systems.

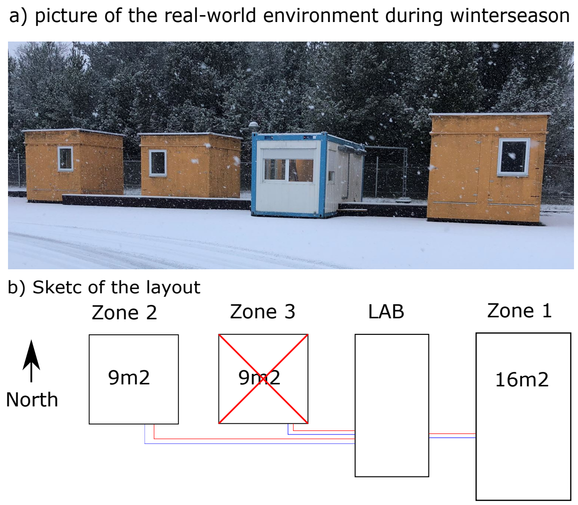
In [Paper D] is Artificial Neural Networks used as function approximators for the black-box model. It is shown in [Paper D] that the following model structure with Long Short Term Memory cells reduces the prediction error on average over 30 time-steps by 53.3% when compared to more traditional Multilayer preceptons and a model structure of only one model.

With the tree frameworks that will be validated in this paper presented can the test environment be presented.

3. Real World Test Environment

The real-world test environment is located at Marinus Svendsensvej 2, 8850 Bjerringbro, Denmark. Here we have set up 3 heating zones that are, from a thermal point of view, independent. However, they are all supplied by the same heating source, so from a hydraulic point of view they are connected.

In Figure 5 a) the real-world environment can be seen during the winter season. In the picture, the 4 buildings this environment consists of can be seen. Tree heating zones and one laboratory. This can also clearly be seen from the layout sketch in Figure 5 b. Unfortunately, due to installation complications have Zone 3 not been included in the experiments.



a) picture of the real-world environment during winterseason

b) Sketc of the layout

Figure 5: In Figure a) can a picture of the test environment during winter be seen. 4 small buildings can be seen in the picture. In Figure b) can the layout of the buildings be seen and how they are hydraulic connected.

In Figure 6 a sketch of the hydraulic installation is seen.

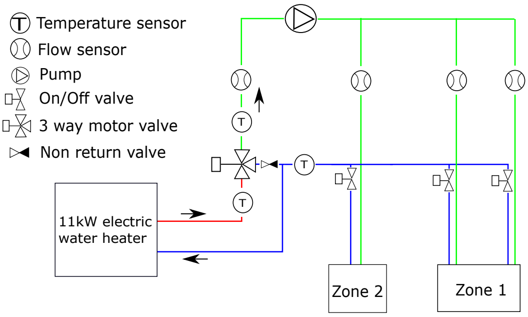


Figure 6: Illustration of the hydraulic setup in the test environment. the arrows indicate the flow direction, red is the hot supply water green is the mixed water and blue is the cooled return water. the position of the sensors can also be seen in the illustration.

From the figure it can be seen where the temperature and flow sensors are installed in the system.

3.0.1. Data collection and hardware

The real-world test environment consists of three data collection devices, one low-level control device, one cloud controller, and a local server. An illustration of this data collection setup can be seen in Figure 7.

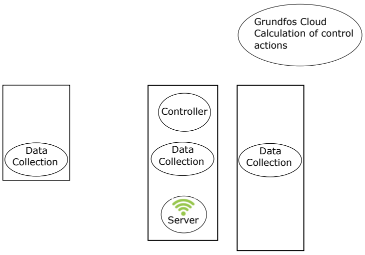


Figure 7: In this Figure is the hardware setup for the real-world experiment illustrated.

The two of the tree data collection devices is configured alike and are placed in each temperature zone. The sensor input collected by these devices can be seen below:

- Room Temperature
- Supply Temperature

- Return Temperature
- Floor Temperature
- Lumen Sensor

The third data collection device is located in the lab. This device collects the following sensor inputs:

- Flow
- Supply Temperature
- Return Temperature
- Ambient Temperature

The server device is a Raspberry Pi 4b. This device creates a local Wi-Fi signal that the data collection devises connects to and sends their sensor signals to the server. The sampling frequency for the data-gathering devices is 5 seconds. The server has Grafana installed and is connected to the internet, making the data easily accessible through Grafana or an API. The code for the data collection devices, the server, and the API can be found in (GIT).

The low-level control device is also a Raspberry Pi 4b. This controller can perform the control actions the controller located in the Grundfos Cloud calculates. The low-level controller can perform the following control actions:

- Control of Supply Temperature with PI controller.
- open/close Valve 1
- open/close Valve 2
- open/close Valve 3

For the supply temperature is a PI controller designed with sensible PI parameters the updating frequency if this is 5 seconds like the sampling rate. The supply temperature setpoint and the position of the valves are calculated in the Grundfos cloud and transmitted to the low-level controller over an API, new control actions are calculated every 800 seconds. Both the low-level controller and the cloud controller have access to the sensor data over the Grafana API.

3.0.2. Bench-marking Test

This section presents the benchmarking test that is performed in the real-world environment, this test is performed with traditional controllers during spring 2021 and winter 2021/2022. The period of the test is from 1th of March to 21th of April and again from 1th of December to 4 January. In total, this is equivalent to 85 days of benchmarking data.

To conduct this benchmark test has two outdoor compensated supply temperatures, and hysteresis controllers been used. The functions for the outdoor compensated supply temperatures can be seen in (5) and (6). The hysteresis ban is set for $0.01C^{\circ}$ and the reference temperature is 22 C° for zone 1 and zone 2.

$$T_{supply} = -0.6 \cdot T_{ambient} + 45. \tag{5}$$

$$T_{supply} = -0.6 \cdot T_{ambient} + 38. \tag{6}$$

The reason two supply temperatures are used is that the first seen in Eq. (5) is used during the spring 2021 and the second seen in Eq. (6) is used during winter 2021/2022.

In the following is 5 sensor signals presented. In Figure 8 is the room temperature for each of the 2 zones presented. In Figure 9 is the Lux measurement for each zone presented and in Figure 10 is the ambient temperature plotted.

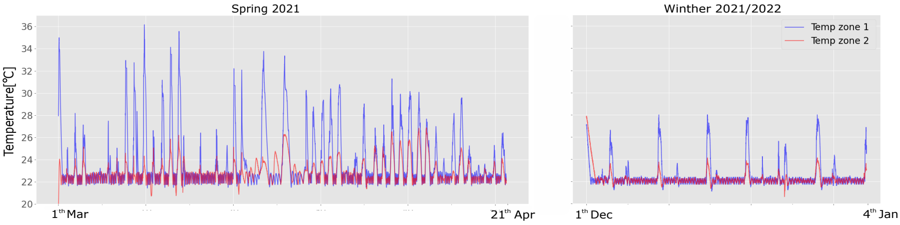


Figure 8: Room temperature in each of the 2 temperature zones during the test period.

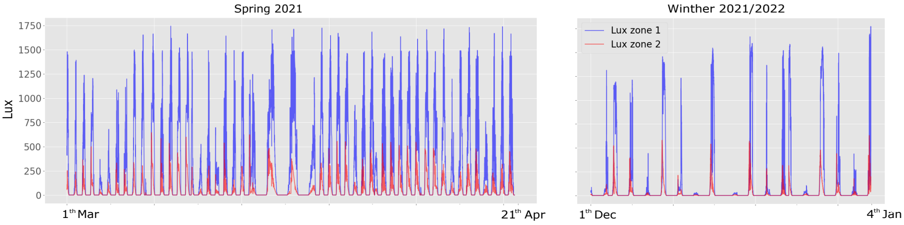


Figure 9: Lux measurement in each of the 2 temperature zones during the test period.

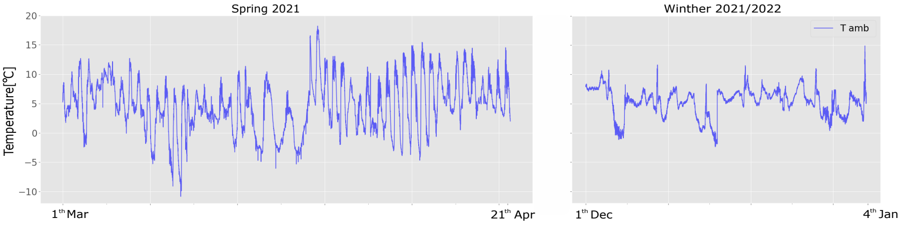


Figure 10: The ambient temperature during the bench marking period.

4. Training and Deployment of Offline RL Policy

Before training and deploying the offline RL framework is the reward/objective function elaborated on. Even though the real-world test setup does not include a heat pump is the reward function written for such an application. The reward functions presented here are developed in [18].

The reward function for the supply agent and the associated sub functions can be seen in Eq. (12) Eq. (13) and Eq. (14) respectively.

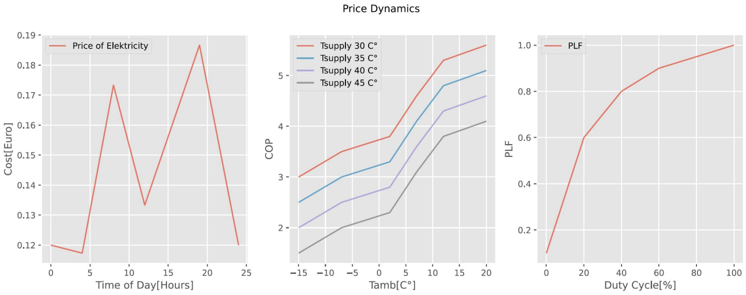
$$R(T_z, V, H_c) = \begin{cases} 2 - (T_z - T_{ref}) - P & \text{if } 21.6 < T_z < 22 \text{ and } VP = 1 \\ -(T_z - T_{ref}) - P & \text{if } 21.6 > T_z \text{ or } T_z > 22 \end{cases} \tag{7}$$

$$SC(T_z, V) = \begin{cases} \text{not active} & \text{if } T_z, > 20.5 \\ \text{active} & \text{if } T_z, < 20.5 \text{ and VP} = 1 \end{cases} \tag{8}$$

$$H_c(SC) = \begin{cases} 1 + H_C & \text{if SC} = \text{active} \\ 5 & \text{if SC} = \text{not active} \end{cases} \tag{9}$$

In the reward function in Eq. (12) it can be seen that the reward is dependent on the room temperature(t_z), the valve position(VP), the price of the energy used (P), and the reference temperature(T_{ref}. The safety controller (SC) is described in Eq. (13). From Eq. (13) can it be seen that if any valve is open and the room temperature is below 20.5 $°C$, then the safety controller takes over. The H_c function describes the numerical value of the negative reward if the SC function is active.

The price of heating with an air-to-water heat pump needs to be emulated due to the lack of the physical hardware. The model seen below considers the COP, partial load factor, and the cost of electricity. In Figure 11 the cost of electricity as a function of time, the COP as a function of the ambient temperature, the supply temperature[22] and lastly the partial load factor as a function of the duty cycle can be seen[23].



(**a**) (**b**) (**c**)

Figure 11: [Paper D]Dynamics of a heat pump: (**a**) Shows the average electricity prices, including taxes in Denmark as a function of the time of day (tod). (**b**) Shows the Coefficient of Performance(COP) as a function of the ambient temperature, for four different supply temperatures. (**c**) Shows the Partial Load Factor(PLF) as a function of the duty cycle (D).

lastly to describe the cost is it necessary to calculate the energy consumption of the system. This is done with the flow measurement and the temperature sensors. The energy consumption is calculated with Eq. (10)

$$\Delta E = Q \cdot (T_{supply} - T_{return}) \tag{10}$$

With the Cost of Electricity (CE), the COP and the PLF described and the power consumption of the system (ΔE), the cost of heating with a heat pump can be simulated with Eq. (11):

$$price = \frac{\Delta E}{COP(T_{amb}, T_{supply}) \cdot PLF(D)} \cdot CE(tod). \tag{11}$$

Much like for the supply agent can the reward function and the associated sub functions for the valve agents be seen in Eq. (12), Eq. 13 and Eq. 14.

$$\mathrm{R}(T_z, V, H_c) = \begin{cases} 2 - (T_z - T_{ref}) \cdot & \text{if } 21.6 < T_z < 22 \\ -(T_z - T_{ref}) & \text{if } 21.6 > T_z \text{ or } T_z > 22 \\ -H_c & \text{if } SC = active \end{cases}, \tag{12}$$

$$\mathrm{SC}(T_z, V) = \begin{cases} \text{not active} & \text{if } 21 < T_z > 23 \\ \text{active} & \text{if } T_z < 21 \text{ and } VP = 0 \\ \text{active} & \text{if } T_z < 23 \text{ and } VP = 1 \end{cases}, \tag{13}$$

$$H_c\ (\mathrm{SC}) = \begin{cases} 1 + H_C & \text{if SC} = \text{active} \\ 5 & \text{if SC} = \text{not active} \end{cases} \quad (14)$$

With the above described reward function and the data from the bench marking can the offline RL method described in Section 2 be train and deployed. For the online deployment is the exploration rate set at 3% this means that 3% of the actions chosen by the agents are not greedy and will therefore be used for exploration of the action state space.

5. Results

In this section is the results of the test with the RL algorithm presented. additionally, are these results compared to the results of the bench marking test. This section is split up into a presentation of the data. Following this is the analysis of the data and a comparison to the bench marking data.

Presentation of RL Data

In Figure 12 the room temperature in the two zones can be seen.

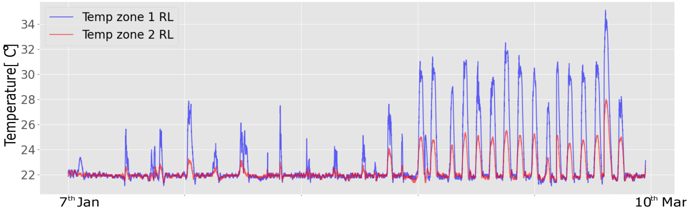


Figure 12: Room temperature in each of the 2 temperature zones during the deployment of the RL algorithm.

From the room temperature plot can it be seen that the temperature is steady at 22 $°C$ except for periods where the radiation causes the temperature to rise.

In Figure 13 is the Lux measurements from the two zones presented.

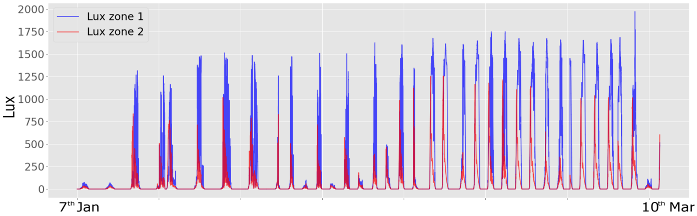


Figure 13: Lux measurement in each of the 2 temperature zones during the deployment of the RL algorithm.

In Figure 13 it can be seen that the Lux measurements and the rise in temperature in Figure 12 are correlated.

In Figure 14 is the ambient temperature over the testing period shown

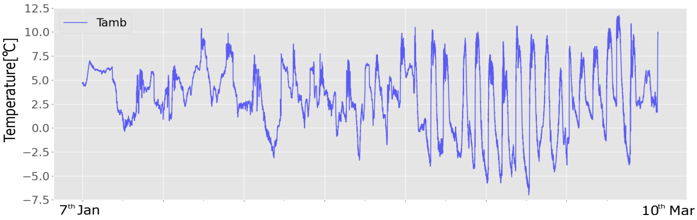


Figure 14: The ambient temperatures during the deployment of the RL algorithm.

If comparing the ambient temperature from the RL deployment with the ambient temperature in the benchmarking in Figure 10 test it can be seen that some of the same tendencies can be seen.

Analysis and Comparison of Data

To asses if the RL control policy found with offline training is performing better than the policy used in the benchmarking test is the following investigated:

- Is the RL controller exhibiting predictive control-like behavior. this is investigated by looking at shorter time series of 24 hours.
- Compare the oscillations of the room temperatures in the benchmark data and the RL data during periods where the system is not saturated.
- Compare duty cycle and supply temperature to estimate if the RL control policy is reducing cost.

For this comparison is only benchmarking data from the winter period 2021 used. The reason for this is that two control policies are used in the benchmarking test. The control policy used during the winter is the best.

In Figure 15 is two 24 hours' time series shown. The time series are taken from the RL policy and benchmarking test, and periods have similar sun and ambient disturbances.

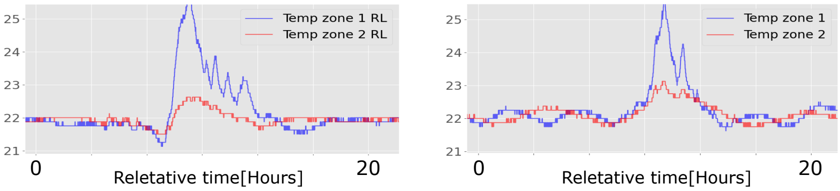


Figure 15: 24 hour time series analysis. In this figure can 24 hours of room temperature data for two similar days for the RL control policy and the traditional control policy be seen.

In Figure 15 can it be seen that the RL policy reduces the temperature in the zones before the sun heats up the rooms, additional can it be seen that the oscillations are smaller. This indicates that the RL control policy exhibits predictive control-like behavior.

To verify that the RL control policy is performing better than the traditional policy is temperature distribution visualized in box plots seen in Figure 16.

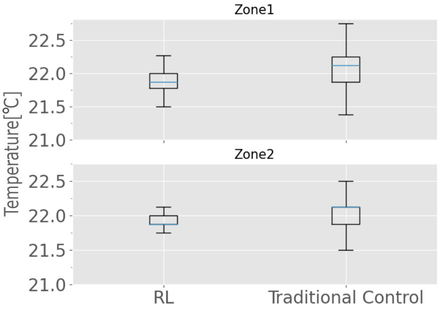


Figure 16: Histograms showing the distribution of the room temperature for the RL control policy and the traditional control policy. The data used in these histograms are only data for periods where the system is not saturated. The system is estimated to be saturated when the temperature is above 23 $^{\circ}C$ and the valves are closed.

The data visualized in Figure 16 is only from periods where the system is not

Table 1: Table showing the duty cycle over the periods of the benchmark test and the test of the RL policy. Additional is the duty cycle over when the system is not saturated shown.

	RL	RL not satuarted	TC	TC not saturated
Zone 1	63	88	42	65
Zone 2	67	83	43	63

saturated. In Figure 16 it can be seen that the distribution for both temperature zones is reduced. In zone 1 is the temperature distribution 43% lower and in zone 2 is 63% lower.

Lastly, to estimate if the RL control policy is more optimal is the duty cycle investigated. The duty cycle is shown in Table 1.

In Table 1 it can be seen that the duty cycle for the RL policy is roughly 20% higher, this is naturally because a lower supply temperature is chosen by the RL controller.

Estimating the energy savings by comparing the price calculated in Eq.(11) cannot be done directly since the benchmarking test and the RL policy test have not been subject to the same disturbances. However, we can compare the duty cycle of the two tests. The relationship between the partial load factor (PLF) and the duty cycle can be seen in Figure 11, where the heat pump dynamics are elaborated on [23].

In Figure 11 can it be seen that the partial load factor is roughly 0.1 lower for the benchmarking test. The PLF has a linear relationship with the efficiency of the system[23], this is also seen in Eq. (11). Because of this, it can be concluded that the RL policy is 10% more energy efficient. If assuming that the price of electricity is the same for the two tests is the RL policy at least 10% more cost-efficient.

6. Conclusion

In this paper is a real-world test environment used to test the RL control framework presented in [15, 18] and Paper D. Firstly, is a resume of the important features and functions of the control framework presented, following this is a description of the real-world environment.

In this paper is benchmarking data presented. The data is gathered during the spring 2021 and winter 2021/2022. Two different control policies are used where the best of the two are used for the comparison with the RL policy data.

The benchmarking data is used is the offline RL framework for offline training where after the RL policy is deployed on the 7th of January 2022. The data gathered from the 7th of January to the 10th of March shows that 1) from looking at the data can it be seen that the overall performance is satisfying, the RL controller does not perform completely unwanted control actions 2) the RL control policy is performing predictive control like action. 3) the oscillation in not saturated periods is reduced by 43% in zone 1 and 63% in zone 2. and 4) the cost is reduced by a minimum of 10% when looking at the duty cycle of the system.

References

[1] L. Pérez-Lombard, J. Ortiz, C. Pout, A review on buildings energy consumption information, Energy and buildings 40 (3) (2008) 394–398.

[2] M. Akmal, B. Fox, Modelling and simulation of underfloor heating system supplied from heat pump, in: 2016 UKSim-AMSS 18th International Conference on Computer Modelling and Simulation (UKSim), 2016, pp. 246–251. doi:10.1109/UKSim.2016.13.

[3] S. Privara, J. Širokỳ, L. Ferkl, J. Cigler, Model predictive control of a building heating system: The first experience, Energy and Buildings 43 (2-3) (2011) 564–572.

[4] H. Huang, L. Chen, E. Hu, A new model predictive control scheme for energy and cost savings in commercial buildings: An airport terminal building case study, Building and environment 89 (2015) 203–216.

[5] A. Afram, F. Janabi-Sharifi, Theory and applications of hvac control systems – a review of model predictive control (mpc), Building and Environment 72 (2014) 343–355. doi:https://doi.org/10.1016/j.buildenv.2013.11.016.
URL https://www.sciencedirect.com/science/article/pii/S0360132313003363

[6] K. M. Tsui, S.-C. Chan, Demand response optimization for smart home scheduling under real-time pricing, IEEE Transactions on Smart Grid 3 (4) (2012) 1812–1821.

[7] L. Yu, T. Jiang, Y. Zou, Online energy management for a sustainable smart home with an hvac load and random occupancy, IEEE Transactions on Smart Grid 10 (2) (2017) 1646–1659.

[8] E. Mills, N. Bourassa, M. Piette, H. Friedman, T. Haasl, T. Powell, D. Claridge, The cost-effectiveness of commissioning new and existing commercial buildings: Lessons from 224 buildings, HPAC Engineering (11 2005).

[9] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[10] E. Barrett, S. Linder, Autonomous hvac control, a reinforcement learning approach, in: Joint European conference on machine learning and knowledge discovery in databases, Springer, 2015, pp. 3–19.

[11] A. Overgaard, B. K. Nielsen, C. S. Kallesøe, J. D. Bendtsen, Reinforcement learning for mixing loop control with flow variable eligibility trace, in: 2019 IEEE Conference on Control Technology and Applications (CCTA), IEEE, 2019, pp. 1043–1048.

[12] T. Wei, Y. Wang, Q. Zhu, Deep reinforcement learning for building hvac control, in: Proceedings of the 54th annual design automation conference 2017, 2017, pp. 1–6.

[13] C. Blad, S. Koch, S. Ganeswarathas, C. Kallesøe, S. Bøgh, Control of hvac-systems with slow thermodynamic using reinforcement learning, Procedia Manufacturing 38 (2019) 1308–1315.

[14] J. García, Fern, o Fernández, A comprehensive survey on safe reinforcement learning, Journal of Machine Learning Research 16 (42) (2015) 1437–1480.
URL `http://jmlr.org/papers/v16/garcia15a.html`

[15] C. Blad, C. S. Kallesøe, S. Bøgh, Control of hvac-systems using reinforcement learning with hysteresis and tolerance control, in: 2020 IEEE/SICE International Symposium on System Integration (SII), IEEE, 2020, pp. 938–942.

[16] H. Kazmi, J. Suykens, A. Balint, J. Driesen, Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads, Applied energy 238 (2019) 1022–1035.

[17] L. Yu, Y. Sun, Z. Xu, C. Shen, D. Yue, T. Jiang, X. Guan, Multi-agent deep reinforcement learning for hvac control in commercial buildings, IEEE Transactions on Smart Grid 12 (1) (2020) 407–419.

[18] C. Blad, S. Bøgh, C. Kallesøe, A multi-agent reinforcement learning approach to price and comfort optimization in hvac-systems, Energies 14 (22) (2021). `doi:10.3390/en14227491`.
URL `https://www.mdpi.com/1996-1073/14/22/7491`

[19] T. Wei, Y. Wang, Q. Zhu, Deep reinforcement learning for building hvac control, in: 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), 2017, pp. 1–6. `doi:10.1145/3061639.3062224`.

[20] S. Levine, Decisions from data: How offline reinforcement learning will change how we use machine learning, [https://medium.com/@sergey.levine/decisions-from-data-how-offline-reinforcement-learning-will-change-how-we-use-ml-24d98cb069b0] (Sebtember 2020).

[21] C. Zhang, S. R. Kuppannagari, R. Kannan, V. K. Prasanna, Building hvac scheduling using reinforcement learning via neural network based model approximation, in: Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, BuildSys '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 287–296. `doi:10.1145/3360322.3360861`.
URL `https://doi.org/10.1145/3360322.3360861`

[22] J. Nie, Z. Li, X. Kong, D. Li, Analysis and comparison study on different hfc refrigerants for space heating air source heat pump in rural residential buildings of north, Procedia Engineering 205 (2017) 1201–1206.

[23] K. Piechurski, M. Szulgowska-Zgrzywa, J. Danielewicz, The impact of the work under partial load on the energy efficiency of an air-to-water heat pump, E3S Web Conf (2017).

ISSN (online): 2446-1636
ISBN (online): 978-87-7573-898-4

AALBORG UNIVERSITY PRESS