Data-driven Offline Reinforcement Learning for HVAC-systems

Article *in* Energy · August 2022

DOI: 10.1016/j.energy.2022.125290

3 authors:

Christian Blad
Aalborg University
7 PUBLICATIONS **50** CITATIONS
SEE PROFILE

Simon Bøgh
Aalborg University
75 PUBLICATIONS **1,884** CITATIONS
SEE PROFILE

Carsten Skovmose Kallesøe
Grundfos Holding and Aalborg University
97 PUBLICATIONS **746** CITATIONS
SEE PROFILE

Energy 261 (2022) 125290

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy

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

ARTICLE INFO

Keywords:
Reinforcement learning
Energy optimization
Black-box models
HVAC-systems
Optimal control

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 presents a method to do pre-training in a black box model environment, where the black box models are built 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 compared with a MLP model. Additionally, different network architectures are tested. It is found that by creating a new model for each time step, 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 Management System (BMS) is available, an RL agent can be deployed which performs close to optimally from the first day of deployment. An optimal control policy reduces the cost of heating by 19.4 % when compared to a traditional control policy in the simulation presented in this paper.

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 type of control has previously been described in the literature in relation to HVAC systems [3–5]. MPC requires a model. However, not two buildings are alike and the dynamic of a building can also change over its lifetime. This means that with MPC a new model is required 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

* Correspondence to: Fibigerstræde 16, 9220 Aalborg East, Denmark.
E-mail addresses: cblad@m-tech.aau.dk (C. Blad), sb@mp.aau.dk (S. Bøgh), csk@es.aau.dk (C.S. Kallesøe).

https://doi.org/10.1016/j.energy.2022.125290
Received 6 January 2022; Received in revised form 14 July 2022; Accepted 24 August 2022
Available online 29 August 2022

Nomenclature

Abbreviations

ANN	Artificial Neural Network
BMS	Building Managements System
GRU	Gated Recurrent Unit
$HVAC$	Heating Ventilation and Air-Conditioning
$LSTM$	Long Short Term Memory
$MARL$	Multi Agent RL
MDP	Markov Decision Process
MLP	Multilayer Perceptron
MPC	Model Predictive Control
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SRM	Supervised Regression Model

Parameters

γ	Discount Factor
Φ	Heat Flux
θ	Weights of ANN
H_c	Hard Constraint
T_{amb}	Ambient Temperature
T_{ref}	Reference Temperature
T_z	Zone Temperature
a'	Future Action
a	Action
CE	Cost of Electricity
COP	Coefficient of Performance
PLF	Partial Load Factor
r	Reward
s'	Future State
s	State

RL in HVAC systems have been published [11–14]. These papers show that RL algorithms compared to traditional event-based controllers can reduce costs between 5.5% and 15%. The papers also 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 [15–17]. 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 gray-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, gray-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. The black box model is developed on data collected from the considered system with the old control activated. The RL/MARL is then trained on the black box model, meaning that the RL/MARL is able to pre-train with control inputs different from the old control inputs much faster than in real time. This idea is a novel solution that solves the problem of poor behavior during early training without using first principle building models. The benefits with the proposed control is exemplified via numerical studies. 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 Fig. 1 scenario A can be seen. Scenario B can be derived from Fig. 1 by removing step 1 where the "traditional controller" is interacting with the environment.

The goals of this paper are verified in a Dymola Simulation, showing 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

One contribution of this paper is simulation tests that show the convergence of the RL using pre-training. Previous studies has shown that convergence is better with MARL that with RL [17]. Therefore, the simulation tests and the training is done with the MARL approach described in the following. Model-free RL is a en 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 Fig. 2.

As seen in the illustration in Fig. 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

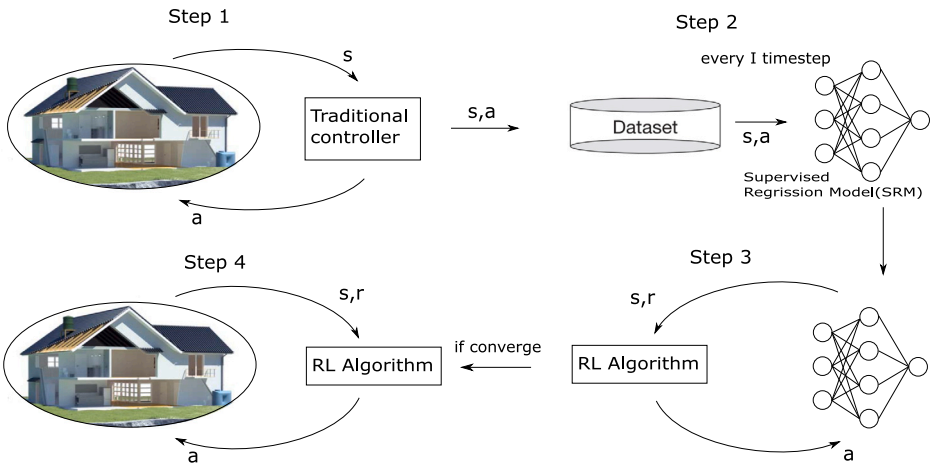

Fig. 1. Illustration of scenario A: The traditional controller interacts with the environment T time steps, after each time step the action and state transition is saved in the data buffer. The data set is passed to the SRM ones trained, the SRM is used as an artificial environment for the RL agent to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

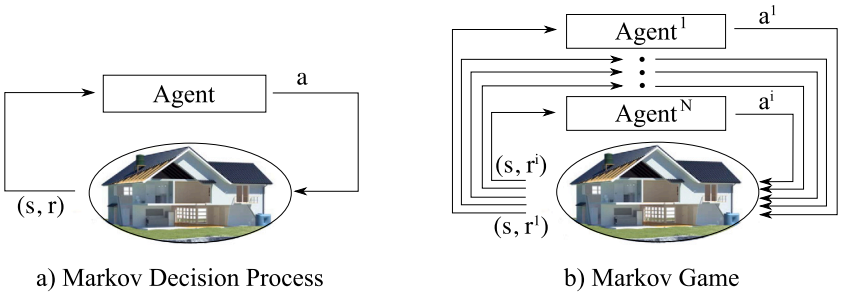

Fig. 2. Illustration of difference between a Markov decision process and a Markov game. In Fig. 1a can one agent and one environment be seen. The agent interacts with the environment by sending a tubule of actions and receiving one reward and the states of the environment. In Fig. 1b can multiple agents 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.

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 typically known, the entire reason for doing value based RL is to learn Q*. this can be done efficiently with Artificial Neural Networks (ANNs). 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 backpropagation through 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 an MG and uses MARL. A Q-learning algorithm for an MG can be formulated with Eq. (2) [22].

$$Q^{*,m}(s,a_1,\dots,a_m) = \mathbb{E}[R_{t+1} + \gamma \max_{a'_m} Q^*(s',a'_1,\dots,a'_m) \mid s,a_1,\dots,a_m] \tag{2}$$

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 $a_1,\dots,a_m$. 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 Eqs. (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.

$$Q^{st}(s^{st},a^{st},a^{v_1..m}) = \mathbb{E}\left[r^{st} + \gamma \max_{a^{st}} Q^{st}(s'^{st},a^{st},a^{v_1..m})\right] \tag{3}$$

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

In is Q^{st} the Q function 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 Fig. 3. Additional can a illustration of a UFH system, with the valve and supply temperature, be seen in Fig. 5 in Section 3.

From Fig. 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.

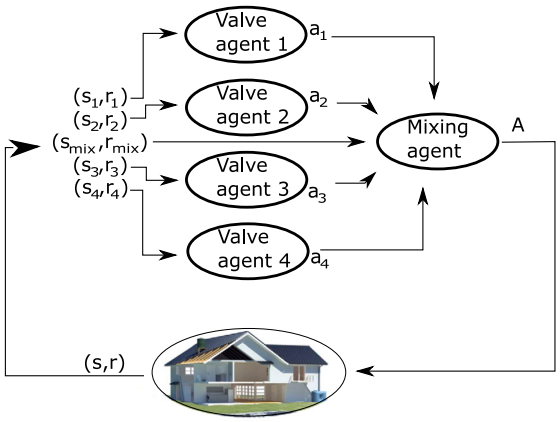

Fig. 3. Illustration of how the agents interacts with each other and the environment. In the figure four valve agents, one mixing agent and a four-zone UFH system are be seen. The sequence of interactions is as follows; all valve agents choose an action based on the state of the environment, these actions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

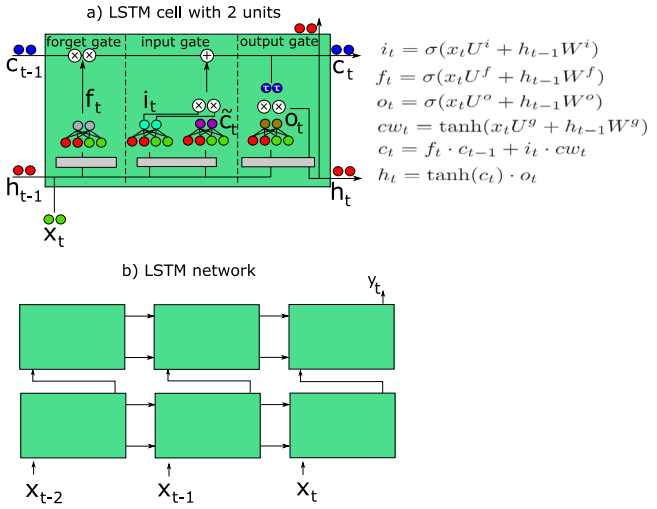

Fig. 4. (a) illustration of LSTM cell: h is the hidden cell state, c is the cell state, x is the state from the environment, f, i cw and o are function dependencies and σ and tanh are activation functions. The mathematical expression of a LSTM cell can be seen to the right. (b) Illustration of supervised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

2.2. Offline training

The main contribution of this paper is the use of offline train of the RL algorithm based on a black box model obtained from historical data. Offline training of an RL algorithm requires a model of the real environment or historical 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 gray-box methods [26]. In gray-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 modeling 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 gray-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 gray-box model will be more data efficient and better at generalizing from a small amount of data, but for the reason stated above a gray-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. In this paper long short term memory (LSTM) is investigated for handling this time-series in the black box model later used for pre-training of the RL. A 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 Fig. 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.

In Fig. 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.

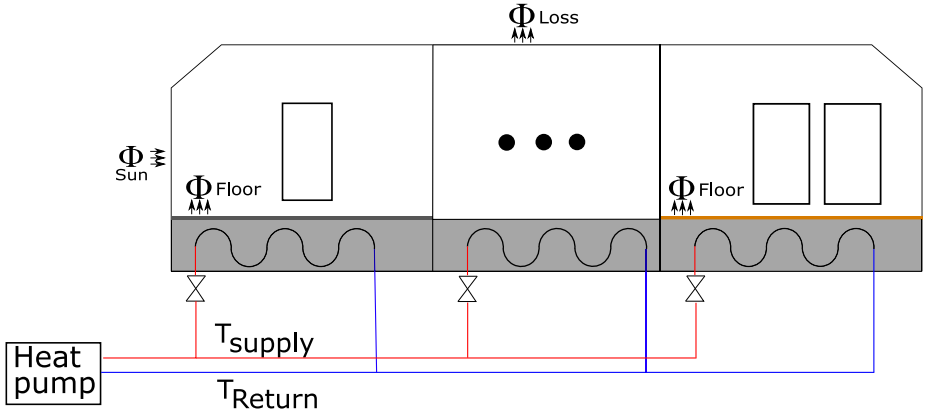

Fig. 5. Illustration of a n zone underfloor heating system. From the illustration can it be seen that the heat supply is a air to water heat pump, and that flow to the individual zones are controlled by on/off valves. additionally can the heat fluxes, Φ_{Sun}, Φ_{Floor} and Φ_{Loss} be seen.

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. The use of a black box model for training the RL agents before deployed in the real environment and its utilization in a HVAC systems is to the authors knowledge new and is the main contribution of this paper. It solves the problem with poor behavior during early training of the RL algorithm. When not pre-trained, this early RL training phase can last for several months, meaning that the training phase cannot be neglected in practical systems. 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 limitations 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. A validation of this model is made in [17] where results from the simulation is compared to building statistics from Denmark.

Fig. 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.

From Fig. 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.

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

Parameter	Zone 1	Zone 2	Zone 3	Zone 4
Length of pipe	56 m	105 m	42 m	70 m
Window area	12 m^2	25 m^2	12 m^2	24 m^2
Wall area	36 m^2	40 m^2	12 m^2	30 m^2
Zone area	16 m^2	30 m^2	12 m^2	20 m^2
Zone volume	48 m^3	90 m^3	36 m^3	60 m^3

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 a 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 1.5 L/h pr meter of pipe in each zone, hence the flow through a zone with a 100 m of pipe is 150 L/h (see Table 1).

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 based on [37,38]. This model is used together with the Dymola simulation environment to calculate the cost of heating. This model take into account the Coefficient of Performance (COP) [37], Partial Load Factor (PLF) [38] and the Cost

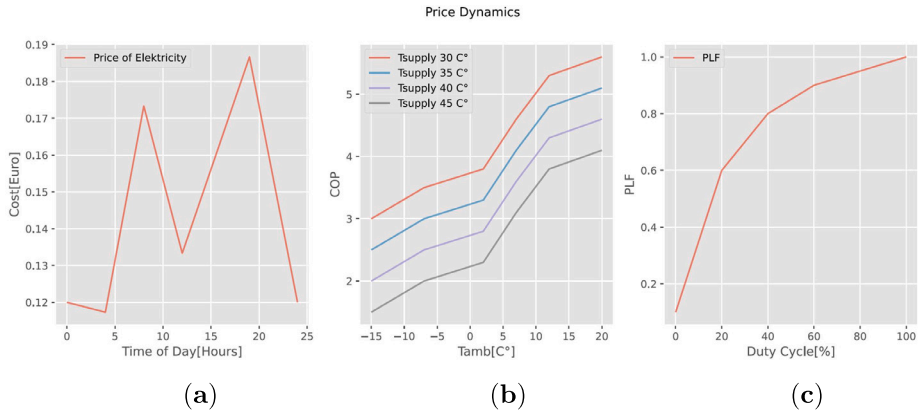

Fig. 6. Dynamics of a heat pump: (a) Shows the average electricity prices, including taxes in Denmark as a function of the time of day (tod). (b) Shows the Coefficient of Performance (COP) as a function of the ambient temperature, for four different supply temperatures. (c) Shows the Partial Load Factor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

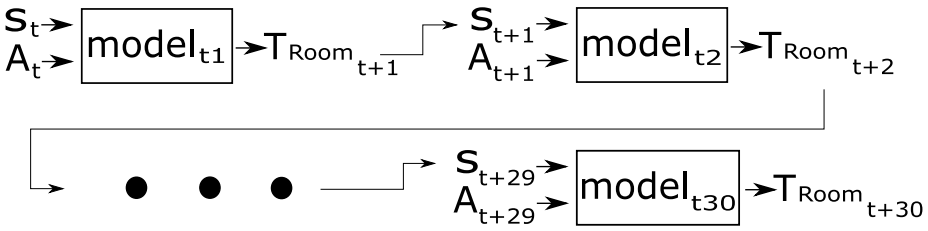

Fig. 7. Illustration of model architecture for supervised learning. This architecture will compensate for the prediction error that occurs in every time step. A new model is made for all predictions meaning that for 30 time steps, 30 models are made.

of Electricity(CE). In Fig. 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.

With the 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 h, 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 Fig. 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 Fig. 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.

From Fig. 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 $[t, \dots, t_{-6}]$	Supply temperature
Ambient temperature $[t, \dots, t_{-6}]$	Valve position
Sun $[t, \dots, t_{-6}]$	
Ambient temperature forecast $[t, \dots, t_{-6}]$	
Sun forecast $[t, \dots, t_{-6}]$	
Time of day $[t, \dots, t_{-6}]$	

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

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 neurons	64	64
Input layer	8 × 6	48
Output layer	1	1

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.

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

From Fig. 8 and Table 4 it can be seen that the prediction error is 45% lower for the 1 step LSTM based model when comparing to the 1 step MLP model. Furthermore, it can be seen that by making a model for each time step and thereby compensating for prediction error the 30 step LSTM model perform 19% better on average than the 1 step LSTM model.

The framework for the interaction between the RL algorithm, the real-world environment and the black box environment is illustrated in Fig. 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 D.
    end for
else if Scenario B == True then
    Store available data in buffer D
end if
Build Black box models from available data in buffer (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 Eqs. (7) and (8).

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

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

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

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 (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 Eqs. (10) and (11).

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

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

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

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.

The following section, present the results of the framework.

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

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

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.

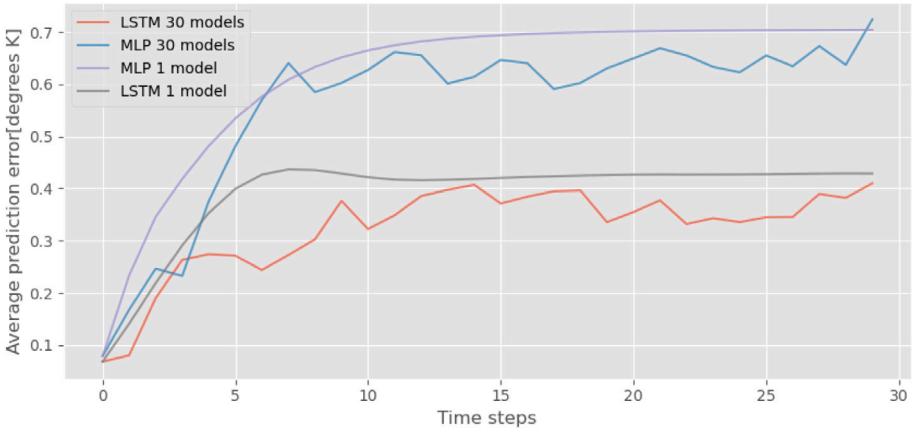

Fig. 8. Plot of the average prediction error for each time step. 4 plots can be seen in the figure; one plot for the 1 model LSTM, one plot for the 30 models LSTM and the same plots for the MLP tests. The data foundation is 60 days for the training of the model and 220 days for the evaluation.

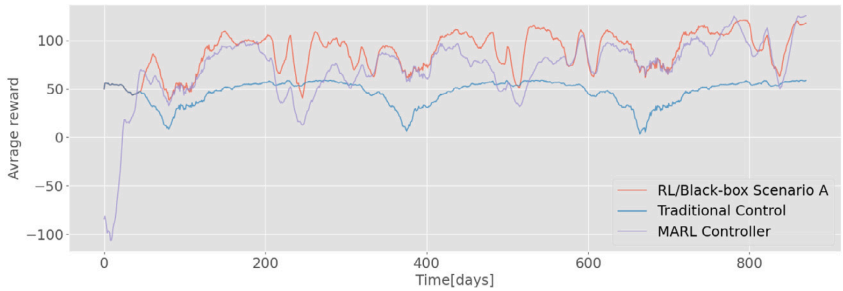

Fig. 9. Reward plot over 880 days for simulation 1, Simulation 2 and Simulation 3, where 1 is; MARL control, without the RL/black box framework, and 2 is; A traditional control policy. and 3 is; the RL/black box frame work in Scenario A.

- 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 Fig. 9 the reward plot for simulation 1, Simulation 2 and Simulation 3 is shown.

From Fig. 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 period is that the RL/black box framework follows the traditional control policy. After 60 days, the RL/black box framework performs better then the controller without the framework. Lastly after approximately 580 days the MARL and RL/black box framework converge to approximately the same control policy.

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

From Fig. 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 Fig. 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 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 Fig. 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.

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

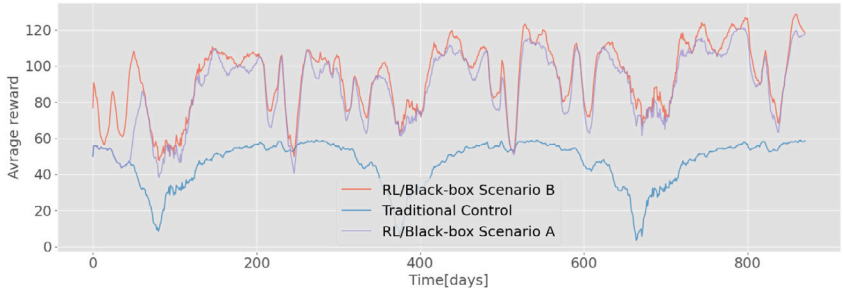

Fig. 10. Reward plot over 880 days for simulation 1, Simulation 3 and Simulation 4, where 1 is; MARL control, without the RL/black box framework, and 3 is; A traditional control policy. and 4 is; the RL/black box frame work in Scenario B.

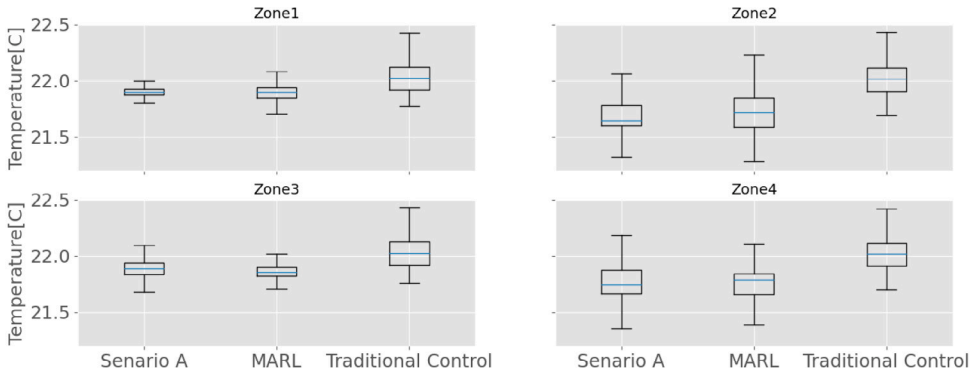

Fig. 11. Plots for each of the four temperature zones. In each plot the temperature distribution for Scenario A, MARL and Traditional Control are plotted. The data foundation is the entire simulation period of 880 days.

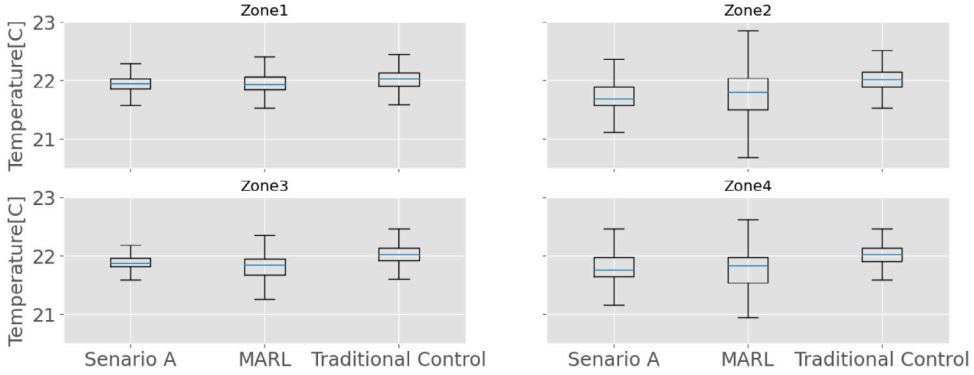

Fig. 12. Plots for each of the four temperature zones. In each plot the temperature distribution for Scenario A, MARL and Traditional Control is plotted. The data foundation is the first 100 days of the simulations.

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	21 854 DKK	0.0%
MARL	42.8 MWh	12.2 MWh	3.54	0.94	18 139 DKK	17.1%
Scenario A	42.9 MWh	11.9 MWh	3.61	0.95	17 943 DKK	17.9%
Scenario B	42.7 MWh	11.7 MWh	3.68	0.98	17 615 DKK	19.4%

From Fig. 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.

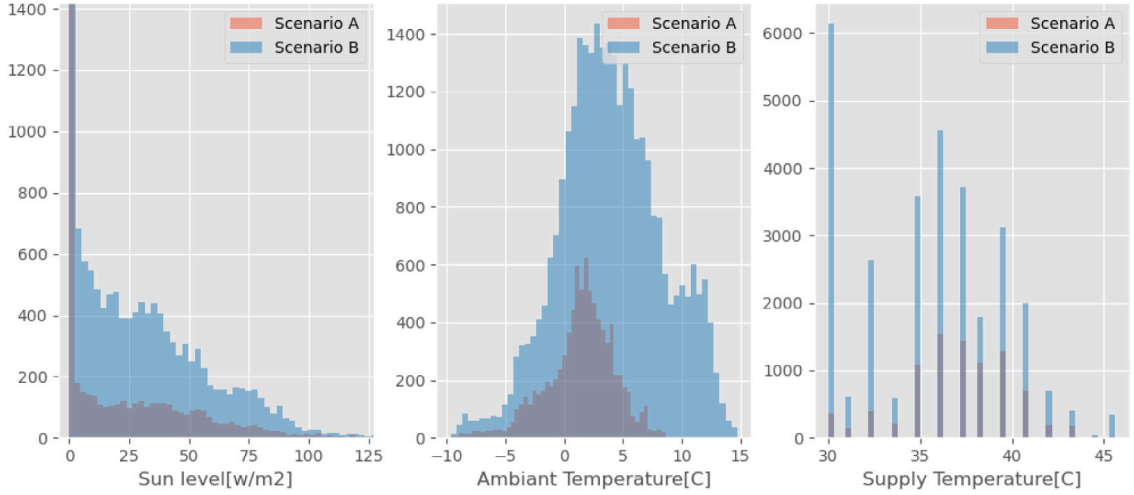

Fig. 13. Data foundation for black box models in Scenario A and Scenario B. Starting from the right is the Sun level, then the ambient temperature and lastly the supply temperature.

The online fine tuning will allow the agent to converge better because all dynamics cannot 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.

In the simulation environment presented in the paper is it shown that poor behavior can be eliminated completely in both a recommissioning task and in a new commissioning. Furthermore, it is shown, in the same simulation environment, that cost of heating can be reduced. For the recommissioning task a 19.4% cost reduction is achieved in the simulated case. And in the new commissioning a 17.9% reduction is achieved.

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 improves performance by 50% when compared to MLP networks. Additionally, are 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 presents a method for doing RL based control of HVAC systems where poor behavior during early training is limited to the current state of art controllers. However, we note two things that still need 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 that includes 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 than current state of art solutions. An internet connection and cloud computing may be a solution to this otherwise, steps must be taken to reduce the complexity of the framework presented in this paper.

CRediT authorship contribution statement

Christian Blad: Formulated the objective of the paper, Made the state of art and stimulation work, Writing – original draft, Writing – review & editing. **Simon Bøgh:** Formulated the objective of the paper, Supervision and comments to improve the manuscript. **Carsten Skovmose Kallesøe:** Formulated the objective of the paper, Supervision and comments to improve the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Pérez-Lombard L, Ortiz J, Pout C. A review on buildings energy consumption information. Energy Build 2008;40(3):394–8. http://dx.doi.org/10.1016/j.enbuild.2007.03.007, URL https://www.sciencedirect.com/science/article/pii/S0378778807001016.

[2] Akmal M, Fox B. 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, p. 246–51. http://dx.doi.org/10.1109/UKSim.2016.13.

[3] Prívara S, Široký J, Ferkl L, Cigler J. Model predictive control of a building heating system: The first experience. Energy Build 2011;43(2):564–72. http://dx.doi.org/10.1016/j.enbuild.2010.10.022, URL https://www.sciencedirect.com/science/article/pii/S0378778810003749.

[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;89:203–16.

[5] Afram A, Janabi-Sharifi F. Theory and applications of HVAC control systems – A review of model predictive control (MPC). Build Environ 2014;72:343–55. http://dx.doi.org/10.1016/j.buildenv.2013.11.016.

[6] Tsui KM, Chan SC. Demand response optimization for smart home scheduling under real-time pricing. IEEE Trans Smart Grid 2012;3(4):1812–21. http://dx.doi.org/10.1109/TSG.2012.2218835.

[7] 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 2019;10(2):1646–59. http://dx.doi.org/10.1109/TSG.2017.2775209.

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

[9] Sutton RS, Barto AG. Reinforcement learning: an introduction. MIT Press; 2018.
[10] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D. Human-level control through deep reinforcement learning. Nature 2015;518(7540):529–33, URL http://dx.doi.org/10.1038/nature14236.
[11] Barrett E, Linder SP. Autonomous HVAC control, a reinforcement learning approach. In: Joint European conference on machine learning and knowledge discovery in databases, Vol. 9286. 2015, http://dx.doi.org/10.1007/978-3-319-23461-8_1.
[12] Overgaard A, Nielsen BK, Skovmose KC, Bendtsen JD. Reinforcement learning for mixing loop control with flow variable eligibility trace. In: 2019 IEEE conference on control technology and applications (CCTA). 2019, p. 1043–8. http://dx.doi.org/10.1109/CCTA.2019.8920398.
[13] Wei T, Wang Y, Zhu Q. Deep reinforcement learning for building HVAC control. In: 2017 54th ACM/EDAC/IEEE design automation conference (DAC). 2017, p. 1–6. http://dx.doi.org/10.1145/3061639.3062224.
[14] Blad C, Koch S, Ganeswarathas S, Kallesøe C, Bøgh S. Control of HVAC-systems with slow thermodynamic using reinforcement learning. Proc Manuf 2019;38:1308–15. http://dx.doi.org/10.1016/j.promfg.2020.01.159, URL https://www.sciencedirect.com/science/article/pii/S2351978920301608, 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.
[15] Kazmi H, Suykens J, Balint A, Driesen J. Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads. Appl Energy 2019;238:1022–35. http://dx.doi.org/10.1016/j.apenergy.2019.01.140, URL https://www.sciencedirect.com/science/article/pii/S0306261919301564.
[16] 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;PP:1. http://dx.doi.org/10.1109/TSG.2020.3011739.
[17] Blad C, Bøgh S, Kallesøe C. A multi-agent reinforcement learning approach to price and comfort optimization in HVAC-systems. Energies 2021;14(22). http://dx.doi.org/10.3390/en14227491.
[18] Wei T, Wang Y, Zhu Q. Deep reinforcement learning for building HVAC control. In: 2017 54th ACM/EDAC/IEEE design automation conference (DAC). 2017, p. 1–6. http://dx.doi.org/10.1145/3061639.3062224.
[19] Levine S. Decisions from data: How offline reinforcement learning will change how we use machine learning. 2020, https://medium.com/@sergey.levine/decisions-from-data-how-offline-reinforcement-learning-will-change-how-we-use-ml-24d98cb069b0.
[20] Zhang C, Kuppannagari SR, Kannan R, Prasanna VK. 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, New York, NY, USA: Association for Computing Machinery; 2019, p. 287–96. http://dx.doi.org/10.1145/3360322.3360861.
[21] Bellman R. Dynamic programming. Science 1966;153(3731):34–7.
[22] Busoniu L, Babuska R, De Schutter B. Multi-agent reinforcement learning: An overview. 310, 2010, p. 183–221. http://dx.doi.org/10.1007/978-3-642-14435-6_7,
[23] Bertsekas D. Multiagent reinforcement learning: Rollout and policy iteration. IEEE/CAA J Autom Sin 2021;8(2):249–72. http://dx.doi.org/10.1109/JAS.2021.1003814.
[24] Levine S, Kumar A, Tucker G, Fu J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. 2020, arXiv:2005.01643.
[25] Kalashnikov D, Irpan A, Pastor P, Ibarz J, Herzog A, Jang E, Quillen D, Holly E, Kalakrishnan M, Vanhoucke V, Levine S. QT-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. 2018, arXiv:1806.10293.
[26] Venugopalan R, Ideker R. Chapter ii.5.10 - bioelectrodes. In: Ratner BD, Hoffman AS, Schoen FJ, Lemons JE, editors. Biomaterials science (3rd edition). 3rd ed.. Academic Press; 2013, p. 957–66. http://dx.doi.org/10.1016/B978-0-08-087780-8.00082-6, URL https://www.sciencedirect.com/science/article/pii/B9780080877808000826.
[27] Afram A, Janabi-Sharifi F. Review of modeling methods for HVAC systems. Appl Therm Eng 2014;67(1):507–19. http://dx.doi.org/10.1016/j.applthermaleng.2014.03.055.
[28] Afram A, Janabi-Sharifi F. Gray-box modeling and validation of residential HVAC system for control system design. Appl Energy 2015;137:134–50. http://dx.doi.org/10.1016/j.apenergy.2014.10.026.
[29] Afram A, Janabi-Sharifi F. Black-box modeling of residential HVAC system and comparison of gray-box and black-box modeling methods. Energy Build 2015;94:121–49. http://dx.doi.org/10.1016/j.enbuild.2015.02.045.
[30] Kusiak A, Xu G, Zhang Z. Minimization of energy consumption in HVAC systems with data-driven models and an interior-point method. Energy Convers Manage 2014;85:146–53. http://dx.doi.org/10.1016/j.enconman.2014.05.053.
[31] Yu Y, Si X, Hu C, Zhang J. A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput 2019;31(7):1235–70. http://dx.doi.org/10.1162/neco_a_01199.
[32] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9:1735–80. http://dx.doi.org/10.1162/neco.1997.9.8.1735.
[33] Shi H, Xu M, Li R. Deep learning for household load forecasting—A novel pooling deep RNN. IEEE Trans Smart Grid 2018;9(5):5271–80. http://dx.doi.org/10.1109/TSG.2017.2686012.
[34] Gers FA, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM. Neural Comput 2000;12(10):2451–71. http://dx.doi.org/10.1162/089976600300015015.
[35] Brück D, Elmqvist H, Mattsson S, Olsson H. Dymola for multi-engineering modeling and simulation. Dtsch Zent Luft- Raumfahrt EV (DLR) 2002;1–8.
[36] Wetter M, Zuo W, Nouidui T, Pang X. Modelica buildings library. J Build Perform Simul 2014;7. http://dx.doi.org/10.1080/19401493.2013.765506.
[37] 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 China. Procedia Eng 2017;205:1201–6. http://dx.doi.org/10.1016/j.proeng.2017.10.354, URL https://www.sciencedirect.com/science/article/pii/S1877705817350294, 10th International Symposium on Heating, Ventilation and Air Conditioning, ISHVAC2017, 19-22 October 2017, Jinan, China.
[38] 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. http://dx.doi.org/10.1051/e3sconf/20171700072.
[39] Blad C, Kallesøe CS, Bøgh S. Control of HVAC-systems using reinforcement learning with hysteresis and tolerance control. In: 2020 IEEE/SICE international symposium on system integration (SII). 2020, p. 938–42. http://dx.doi.org/10.1109/SII46433.2020.9026189.