2019 IEEE Conference on Control Technology and Applications (CCTA)
Hong Kong, China, August 19-21, 2019

Reinforcement Learning for Mixing Loop Control with Flow Variable Eligibility Trace

Anders Overgaard[1,2], Brian Kongsgaard Nielsen[1], Carsten Skovmose Kallesøe[1,2] and Jan Dimon Bendtsen[2]

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

I. INTRODUCTION

There is a lot of energy to be saved by improving building heating, ventilation and air-conditioning (HVAC). In the United States 40% of energy consumption is in buildings, with 50% of that being from HVAC systems [1]. It is estimated in [2] that 11-16% can be saved by improving control. Due to this huge savings potential multiple control schemes are being researched, where Model Predictive Control [3] and Multi-Agent Systems [4] are two promising areas.

A major problem is improper or lack of commissioning of building. Only around 5% of buildings get commissioned [5], leaving the remaining buildings without well tuned HVAC controls. By introducing self learning controls the need for commissioning can be diminished. Reinforcement Learning as a self learning controller has been studied for building HVAC in [6], [7] and [8]. Reinforcement learning was combined with deep learning function approximation for HVAC control in [9] and building energy optimization in [10].

In this paper the focus is on Mixing Loops which is part of the hydraulic thermal distribution system in buildings. In [11] a method for taking into account the flow variable delay in prediction of the return temperature was shown to improve the prediction. In this work it is shown that adding flow variable delay into a Reinforcement Learning controller improves the performance leading to cost savings.

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

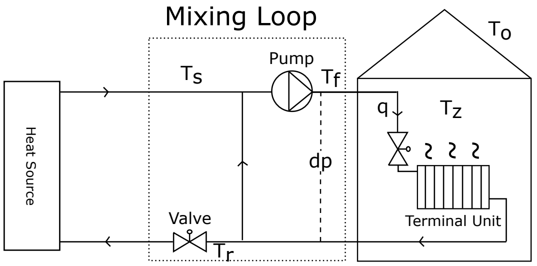

Fig. 1. Schematic of simple mixing loop application

II. BUILDING HEAT SUPPLY VIA MIXING LOOP

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

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

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

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

III. PRELIMINARIES

This work makes use of the Reinforcement Learning controller $Q(\sigma, \lambda)$ introduced in [12] which combines state of the art methods for dealing with temporal difference and eligibility traces in a unified manner. In this section some basic concepts of Reinforcement Learning are briefly

[1] Grundfos Holding A/S, Core Technology - Department of Control Technology, Bjerringbro, Denmark. {anovergaard,ckallesoe,bknielsen}@grundfos.com

[2] Aalborg University, Department of Electronic Systems, section of Automation and Control, Aalborg, Denmark. {ano,csk,dimon}@es.aau.dk

978-1-7281-2767-5/19/$31.00 ©2019 IEEE

summarized. For a deeper look into Reinforcement Learning the reader is referred to [13].

A. Basics

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

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \quad (1)$$

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

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

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

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

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

$$A_t = \arg\max_a \, q_\pi(s,a). \quad (4)$$

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

B. Temporal Difference

Temporal difference is a central concept of Reinforcement Learning, where ideas from both Monte Carlo and Dynamic Programming are used. Where Monte Carlo waits until the episode is finished to update the estimate of the value function, dynamic programming bootstraps using current estimates to form a new estimate. A simplified representation of this is that Monte Carlo uses an estimation of [13]

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

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

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

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

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

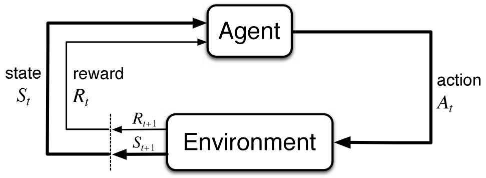

Fig. 2. Reinforcement Learning [13].

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

In [13] the method Q(σ) was first introduced. Here σ is used as a weight between two approaches to temporal difference error

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

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

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

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

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

C. Eligibility Traces

Multi Step Reinforcement Learning learns from the return

$$\begin{aligned} G_{t:t+n} \doteq\ & R_{t+1} + \gamma R_{t+2} + ... \\ & + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n}), 0 \leq t \leq T-n \end{aligned} \quad (10)$$

In [14] the TD(λ) method was introduced where a trace decay of the returns in implemented

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

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

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

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

$$\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha\delta_t\mathbf{z}_t \quad (13)$$

D. The method Q(σ,λ)

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

E. Radial Basis Function Approximation

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

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

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

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

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

IV. Proposed Method

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

A. Flow Dependent Eligibility Trace

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

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

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

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

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

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

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

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

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

B. Flow dependent Q$_\phi(\sigma,\lambda)$

The proposed algorithm for online Q$_\phi(\sigma,\lambda)$ with flow variable λ can be seen in Algorithm 1. For the operation of finding solutions to problems such as $\max_{\mathbf{a}} Q(\mathbf{w}^T\mathbf{x}(\mathbf{s},\mathbf{a}))$ different solvers can be used. In this work a search algorithm was made, which utilizes the knowledge of location of feature points in the radial basis network to make multiple local gradient searches for finding a global maximum. Due to scope of this paper, this solver will not be further introduced.

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

Algorithm 1: Algorithm $Q(\sigma,\phi)$

V. Test

Testing on buildings is not a trivial task. It is very time-consuming due to slow dynamics and there is often a desire to test performance over multiple years. Furthermore benchmarking can be imprecise due to not having an equal comparison due to different load conditions. This can pose

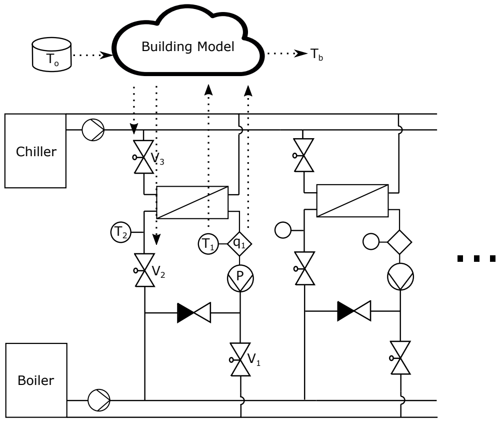

Fig. 3. Hardware in the loop. Four parallel systems installed.

a problem for rapid development. A hardware in the loop approach is used here for faster testing.

The test setup consists of two parts; A hydraulic mixing loop system and a building model. The hydraulic dynamics of the mixing loop react much faster than the thermal dynamics of the building. To increase testing speed, the model of the building is run at accelerated speed in the loop together with the hardware hydraulics. The idea is to run the building model faster, but still slower than the hydraulic dynamics to increase testing speed. This allows for, in the specific test setup, to simulate 12 days in the time of 1 day.

A. Hydraulics

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

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

B. Building Model

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

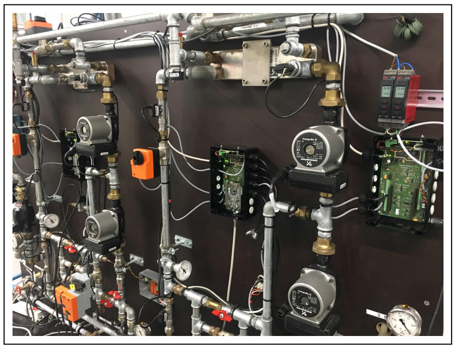

Fig. 4. Picture overlooking part of test setup.

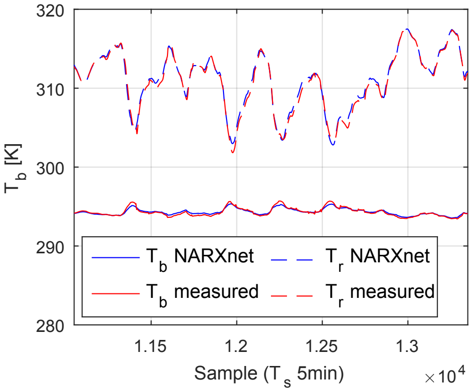

Fig. 5. Validation of building- (T_b) and return temperature (T_r). Data shown is only 8 days example out the full 3 months validation data.

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

C. Controllers

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

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

Here $e(t)$ is the building temperature error that is used when in heating mode, but not in set back mode. $\beta = 0.5$ is a weight between comfort and cost due to the multi objective nature of the reward. $\psi_{heat}(t,\Delta T)$ is the heating power cost

TABLE I

Radial basis function dimension and range.

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

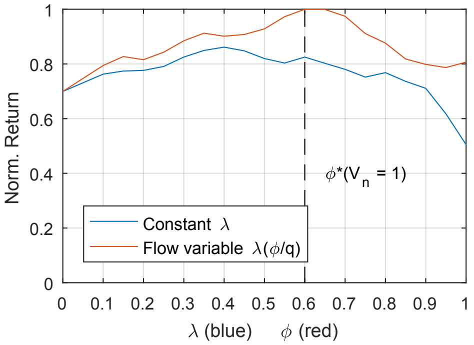

Fig. 6. Norm. yearly return for different values of constant λ and flow variable delay with different α for physical model with $V_n = 20$.

which is determined by the district heating company as a function of ΔT. The lower ΔT the higher price. $\psi_{pump}(t)$ is the pump power cost. See [15] for further description of the chosen reward function. To determine the performance of the proposed algorithm it is compared with an industrial standard controller for mixing loops. The controller that is being compared with is the one installed in the office building from which the model was derived. This controller is developed by a major Building Management System (BMS) supplier that is kept anonymous. The industrial controller also performs night setback in the hours 21 to 5.

VI. Determining ϕ^*

A first principle physical model which was introduced in [15] is first used to compare performance of $Q(\sigma,\lambda)$ with the proposed flow dependent $Q_\phi(\sigma,\lambda)$ for different values of λ and ϕ. The controller was trained for a year and afterwards evaluated running a second year. In Fig. 6 it can be seen that the highest yearly return occurs at $\phi^* = 0.6$. To determine the relation between the lumped scaled volume and ϕ^* a numerical approximation was done by doing multiple test at different lumped volumes. The sampling time was kept constant $t_s = 300s$ such that an approximation for $\phi^* = h_{t_s=300}(V_n)$ was found as seen in Fig. 7. From this relation ϕ^* can be determined for the building given V_n. Different ways can be used to get an estimate of the lumped model, such as knowledge of piping. Here it was determined for the office building in Bjerringbro via data analysis. A method for determining the lumped volume through data analysis using Mutual Information was shown in [11]. The lumped scaled volume (V_n) was for the tested building found to be $1.15m^3$ which via the linear approximation $h_{t_s=300}(V_n)$ leads to $\phi^* = 0.8$.

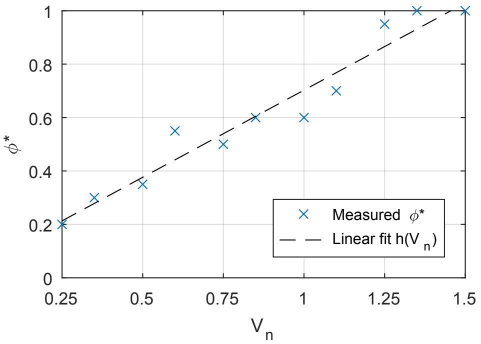

Fig. 7. Relation between α and V_n. Used to determine $C(V_n)$ from physical model.

VII. Results & Discussion

The first results shows how the $Q_\phi(\sigma,\lambda)$ controller performs over the first 5 months compared with the industrial controller. Using the rapid hardware in the loop setup this test took 12.5 days to run. Fig. 8 the mean absolute value of the weights is shown to as a representation of how the 3000 weights converge. If the system is time invariant and the system is fully explored the weights will over time converge to a final value. There is a steep learning curve the first 45 days with following slower convergence. The development of the weights is an indication of stable convergence. Fig. 9 shows the root mean square error (RMSE) of the building temperature and the heating cost with a moving mean covering the last 14 days. Here it can be seen that the RMSE is at first worse than the industrial controller, but over time $Q_\phi(\sigma,\lambda)$ learns to provide conditions for the local controller to achieve low RMSE. In the cost plot it can be seen that after approximately 55 days $Q_\phi(\sigma,\lambda)$ starts saving compared to the industrial controller. In the last plot the return of the rewards over 14 is shown which is the goal $Q_\phi(\sigma,\lambda)$ optimizes towards. Here 55 days seems to be the point where $Q_\phi(\sigma,\lambda)$ overtakes the industrial controller in performance defined by the reward function. Having worse performance for approximately 2 months than a well tuned industrial controller and afterwards improving seems reasonable, especially if some of the training can be done while commissioning of the building is ongoing.

For the next results the different controllers are allowed to train for 5 months. Afterwards the controllers are run for 6 months on a different weather data set than was used for training. Summation of normalized return, RMSE and Cost is done for all the controllers and compared in TABLE II. All the controllers provide conditions for the local temperature controller to achieve similar low RMSE. On return and cost the proposed method with the estimated ϕ^* performs best. Compared to the industrial controller it saves 20.5% in costs over the 6 months period. The controllers with lower and higher ϕ values perform worse, but still better than the industrial controller and when using a constant eligibility trace λ.

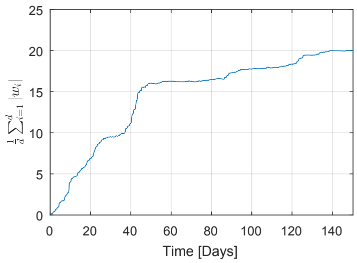

Fig. 8. mean absolute value of weights converging during training.

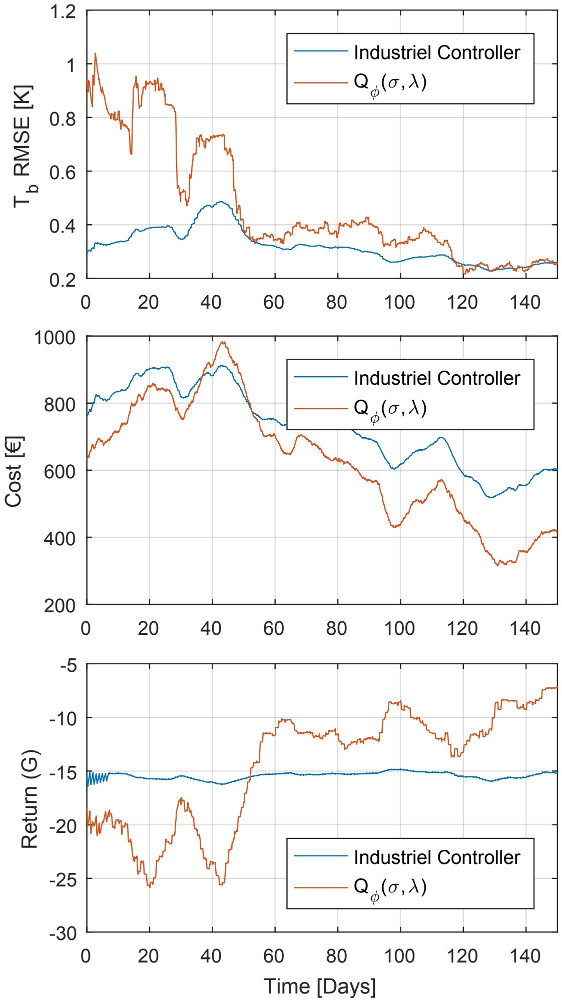

Fig. 9. Temperature error, cost, and return in a running 14 days window.

VIII. Conclusion

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

TABLE II
Comparison of Controllers performance over 6 months after 5 months training.

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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