Towards plug&play Smart Thermostats inspired by Reinforcement Learning

Charalampos Marantos
School of ECE, National Technical University of Athens, Greece
hmarantos@microlab.ntua.gr

Christos P. Lamprakos
School of ECE, National Technical University of Athens, Greece
clabrakos@gmail.com

Vasileios Tsoutsouras
School of ECE, National Technical University of Athens, Greece
billtsou@microlab.ntua.gr

Kostas Siozios
Department of Physics, Aristotle University of Thessaloniki, Greece
ksiop@auth.gr

Dimitrios Soudris
School of ECE, National Technical University of Athens, Greece
dsoudris@microlab.ntua.gr

ABSTRACT

Buildings are immensely energy-demanding and this fact is enhanced by the expectation of even more increment of energy consumption in the future. In order to mitigate this problem, a low-cost, flexible and high-quality Decision-Making Mechanism for supporting the tasks of a Smart Thermostat is proposed. Energy efficiency and thermal comfort are the two primary quantities regarding control performance of a building's HVAC system. Apart from demonstrating a conflicting relationship, they depend not only on the building's dynamics, but also on the surrounding climate and weather, thus rendering the problem of finding a long-term control scheme hard, and of stochastic nature. The introduced mechanism is inspired by Reinforcement Learning techniques and aims at satisfying both occupants' thermal comfort and limiting energy consumption. In contrast to to existing methods, this approach focuses on a plug&play solution, that does not require detailed building models and is applicable to a wide variety of buildings as it learns the dynamics using gathered information from the environment. The proposed control mechanisms were evaluated via a well-known building simulation framework and implemented on ARM-based, low-cost embedded devices.

KEYWORDS

HVAC control, Intelligent agents, Energy efficiency, Learning systems, Decision making, Embedded software

ACM Reference Format:
Charalampos Marantos, Christos P. Lamprakos, Vasileios Tsoutsouras, Kostas Siozios, and Dimitrios Soudris. 2018. Towards plug&play Smart Thermostats inspired by Reinforcement Learning. In *INTelligent Embedded Systems Architectures and Applications (INTESA), October 4, 2018, Turin, Italy.*

1 INTRODUCTION

Intelligent computing systems are gradually reshaping the world as we know it, in an effort to optimize every aspect of contemporary activities. Unprecedented monitoring and calculation abilities are at the disposal of system designers, which in turn need to designate novel applications with societal impact. A relative important field is rural development, since buildings are immensely energy-demanding, consuming around 40% of the total European Union's energy [16]. Taking into account that the total consumption increases by 1% per year [15], a balancing mechanism is required abiding to the concept of energy consumption minimization.

There is a plethora of techniques that aim to optimize the financial and ecological cost of buildings. Innovative design methods, new materials and appliances are used during the construction of new buildings including insulation improvements and more energy efficient HVAC systems. While new green building can be optimized for energy savings, maximizing also the usage of renewable sources, an efficient solution for old buildings is crucial, as the existing infrastructure cannot be replaced in cost effective manner.

An energy optimization technique that applies to all buildings, regardless of their age is fine-tuning and control of its heating, ventilation, and air conditioning (HVAC) systems. An online control system of HVAC is frequently referred to as a Smart Thermostat: Computerized embedded platforms that apply advanced control methods on HVAC systems. Smart Thermostats promise to achieve energy reduction and better thermal conditions by proper configuration of the HVAC system at real-time based on environmental parameters, building's state and occupants preferences. Recent reports estimate that the global smart thermostat market is expected to generate a revenue of $1.3 billion by 2019 [1]

Embedding intelligence on a dynamic HVAC configuration has attracted the interest of many researchers over the years resulting in numerous design approaches. This work focuses on a plug&play solution that is applicable in a wide variety of buildings, aiming at a rapid prototyping solution (low design time). The core of the decision making logic of the proposed Smart Thermostat is inspired by Reinforcement Learning augmented with supervised learning techniques in order to effectively adapt to the parameters and dynamics of the controlled environment. A further contribution of this

[1] According to a recent report by Sandler Research

work is an optimal cost formulation, creating an efficient normalization of the competitive efficiency metrics (energy and thermal comfort), in order to be equally taken into account without any prior knowledge. Experimental results, using popular simulation software (EnergyPlus), highlight the effectiveness of this work.

The rest of the manuscript is summarized as follows: Section 2 provides an overview of relevant approaches found in literature, whereas Section 3 provides the technical background that is considered necessary in order for the reader to have a clear view of the aspects that will be discussed afterwards. The proposed framework, as well as its components, is discussed in detail in Section 4, while Section 5 presents the experimental setup. The efficiency of our proposed solution is discussed and quantified under various metrics in Section 6. Finally, Section 7 concludes the manuscript.

2 RELATED WORK

In the context of dynamic HVAC control two major techniques, i.e. on-line decision-making and Model Predictive Control (MPC), dominate the literature with each one being characterized by number of pros and cons. On-line algorithms usually require lower design time, while MPC methods are usually more robust efficient, especially in cases where the control was designed along with the system. Nevertheless, on-line methods are more reactive to real-time conditions, whereas the accuracy of MPC techniques is affected by the precision of the weather forecasting and building dynamics models.

MPC control techniques have been successfully applied in a wide range of similar non-linear applications [22, 24], including HVAC system control [1]. In most cases the design of the controller requires extensive analysis of the system, which leads to high-dimensional mathematical problems [14] requiring high computational power. As a result a great amount of design and customization time for every different type of building (detailed experimental and mathematical analysis) is needed. In general, MPC methods cannot support real time applications but are better for controlling components of HVAC systems that have been modeled at design time and are not affected by the building's behavior.

Fuzzy rules [2, 9, 30] alleviate the necessity of a detailed mathematical model, through a fuzzy approximation scheme. The controller follows a (usually predefined) action plan according to the information received by the environment. Sometimes genetic algorithms are employed to support the fuzzy controller [3, 19].

Supervised machine learning techniques, such as Artificial Neural Networks (ANNs) [7, 21], are recently gaining a lot of attention, due to the fact that they do not require a detailed study of the underlying dynamics of the building. Contrariwise, they can be trained, basing on historical data and learn the behavior of the building's physics. Although these techniques are in alignment with the model-free controller idea, they have a number of limitations. Machine learning models usually need long time to be trained and calibrated and are difficult to implement in practice, especially as a lightweight plug&play solution, while fuzzy rules create fuzzy classes of some parameters and as a result they are not able to learn building's behavior in detail, in order to react on real-time dynamics. Therefore a stage of "pre-training" is performed, based on historical data of a target building that they target to or building modeling tools (e.g. EnergyPlus, Modelica).

Reinforcement Learning (RL) promises to give a solution by continuously learning through the results of different inputs in the system. This is achieved by matching each action to a reward that accrues by the evaluation of the produced output. RL is gaining attention nowadays and a growing use in the field of embedded systems is observed [28]. Several state-of-the-art approaches use RL for HVAC control [5, 11, 33]. Usual criticism to RL is the instability at the initial system period, as well as prolonged learning periods [1].

As far as the use case of Smart Thermostats is concerned, several works focus only on energy consumption minimization regardless of thermal comfort. Some take into account the energy market, trying to satisfy a desired threshold set by users, either by controlling the HVAC system [35] or escaping from the limits of available thermostat choices by choosing a purchase-bidding strategy for the building [25]. The first approach [35] uses simulations to develop a linear regression model that is related only to temperature difference, while the second one [25] assumes a full model for estimating energy consumption, based on modeled building parameters and a computationally demanding Monte Carlo approach. On the other hand, a big number of proposed solutions are attempting to serve occupant's preferences according to their manual modifications on temperature set-points. These approaches attempt to build a schedule and provide energy savings by avoiding unnecessary adjustments, normalizing fluctuations and turning off the HVAC when the zone is not occupied. In order to achieve this, some works ask the user to identify the comfort zone manually [10], [6].

Our proposed method envisions a controller that takes into account both energy and thermal comfort, solving a multi-objective optimization problem. Similarly, [4] proposes a control method that comprises energy with a comfortable lifestyle and provides a solution to the whole Smart Home tasks scheduling, using a detailed model of the building and a predefined thermal comfort model. A low cost and flexible solution to the Smart Thermostat problem is devised in [12], coupling Neural Networks (NN) with Fuzzy control. However, the solution employs a NN that is pre-trained off-line using a thorough design space exploration. The results highlighted that a machine learning technique can be very efficient, leading to near optimal results with low computational complexity.

Regarding RL, an examination its application on Smart Thermostats has been introduced in [5]. In this work, energy cost corresponds to a reward of -1 when the HVAC is on but no actual energy costs are integrated. Additionally, the controller tries to achieve a predefined temperature by occupants. Another approach formulates a reward function that focuses only on minimizing energy cost taking into account a desired range for the temperature [33]. However, this range is occasionally violated and does not consider more realistic thermal comfort values. Finally, [11] is the only work on RL that comprises both energy and thermal comfort in construction of the reward function. However, the comfort exceeds the acceptable limits for numerous periods, while the technique relies on some prior information such as the maximum energy consumption.

3 THEORETICAL BACKGROUND

3.1 Reinforcement Learning - NFQ Algorithm

In alignment to the unknown parameters of the system, a deterministic approach for decision making is limited by approximations

regarding the available system states, able to be reached at run-time. Similarly, when the parameter space is vast, the definition of deterministic transitions from one state to another can be prove to be infeasible. Such design requirements, gave birth to Reinforcement learning (RL) approach, which constantly gaining attention.

A Reinforcement Learning problem, consists of a set S of states, a set A of actions, and a function $r : S \times A \rightarrow R$, called the reward function. At each instance of the problem, an action $a_i \in A$ (we assume that A is finite) has to be chosen, which will lead from $s_i \in S$ to a new state s_{i+1}. The tuple (s_i, a_i, s_{i+1}) is called a *transition*. A real value r_i is assigned to each of the transitions. The agent's goal is a series of transitions $t_1, t_2, \ldots, t_n$ that maximizes the R value (called *the return*). Since maximizing a reward is equal to minimizing a cost, with the said cost defined as the negative of the reward, we will refer to c as the *cost* function for the rest of this manuscript. A real value c_i is assigned to each of the transitions. The mathematical formulation of this problem is given by Eq. 1, while the minimization of total cost instead of maximization of the total reward, is the differentiator compared to conventional Reinforcement Learning problem. The $\gamma \in [0, 1)$ is a discounting factor that controls the importance of future rewards and ensures convergence of the sum in Eq. 1 when $n \rightarrow \infty$.

$$Minimize\ R = \sum_{i=0}^{n} \gamma^i c_i \tag{1}$$

Although state-values suffice to converge to an optimal solution, it is useful to define action-values. Given a state s and an action a, the action-value of the pair (s, a) is is defined as Q and is calculated according to Eq. 2, where R stands for the return associated with first taking action a in state s. Consequently, estimating Q plays a key role on the overall system effectiveness as it quantifies the efficiency of the possible alternative selections that can be performed by the decision-making mechanism.

$$Q(s, a) = \mathbb{E}[R|(s, a)] \tag{2}$$

In this paper, RL is employed via Neural Fitted Q-iteration (NFQ), which has proven successful in real world applications [26][27]. NFQ uses a multilayer perceptron (MLP) in order to approximate the Q function. The agent acts ϵ-greedily on each state encountered based on its current approximation of the Q function. All of the resulting transitions are stored on a growing batch of data on which the MLP is trained, and the estimation of $Q(s, a)$ is renewed.

3.2 Thermal Comfort

Thermal comfort counts the satisfaction of people in a thermal environment. Thermal comfort can not be measured directly and therefore can only be estimated using a number of parameters. A popular index for estimating occupants' thermal comfort is the Predicted Mean Vote (PMV). It was developed by Fanger [17] and produces values in a seven-point scale ($[-3, 3]$). The sign of the PMV denotes feeling colder or warmer than the ideal.

4 PROPOSED CONTROL ALGORITHM

An overview of the proposed controller framework is schematically presented in Figure 1. Each set-point generation cycle starts with retrieval of the current system state. For a feasible approximation of

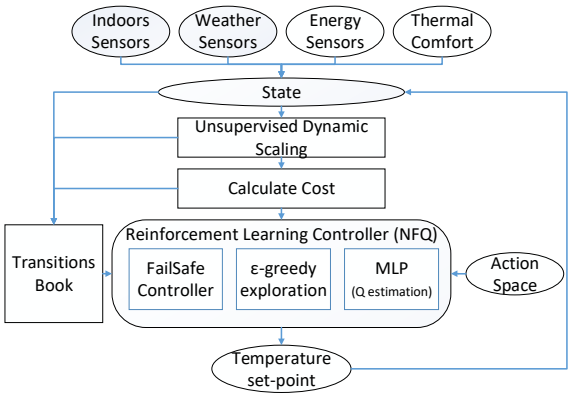

Figure 1: Proposed controller's framework

the Q-function, the system state has to fulfill the Markovian property, i.e. it has to contain all information relevant to the Q function. Additionally, in the proposed design the state vector must be able to effectively capture both energy consumption and thermal comfort. Consequently, in this work the system state is summarized by a vector of Outdoor temperature, Solar radiation, Indoor humidity and Indoor temperature.

The term *action* refers to a set-point designation for the thermostat. The state-action space has to be kept minimum, in order for the agent to learn as fast as possible. The actions in this case are:

- Maintain indoor temperature
- Increase indoor temperature by 1^oC
- Decrease indoor temperature by 1^oC

The agent's knowledge is the history of all encountered states, taken actions and received costs. We will refer to this history as Transitions Book (TB). As NFQ implies, TB is a batch of data in the form of concatenated tuples (s_i, a_i, c_i), one for each transition (s_i, a_i, s_{i+1}). The cost function (Eq. 3) of the proposed decision making mechanism is computed with respect to the energy consumption E of each transition as well as the thermal comfort value in the form of PMV. Moreover, a trade-off tr ($0 \leq tr \leq 1$) is introduced in the cost calculation to allow the user to designate its preference with respect to the importance of Energy and Comfort.

$$c(s, a, s') = \begin{cases} tr \cdot E_{std} + (1 - tr) \cdot |PMV|_{std} & \text{(non-terminal)} \\ terminal\_cost & \text{else} \end{cases} \tag{3}$$

More precisely, E_{std}, $|PMV|_{std}$ are normalized values. In the extreme cases of $tr = 0$ or $tr = 1$ the resulting control is purely comfort-driven or energy-efficiency-driven, respectively. Regarding the PMV value, absolute values are used since our goal is to minimize the distance from the ideal conditions ($PMV = 0$).

A practical obstacle in the calculation of the cost function is the fact that the value range of the two involved quantities may differ significantly. This difference would inevitably affect the cost calculation and thus a mitigation strategy is mandatory.

Adhering to our plug&play design concept, we refrain from using existing knowledge in order to achieve scaling of Energy and thermal comfort values. On the contrary, we adopt a *unsupervised dynamic scaling* [8] technique, where the running average and

standard deviation are dynamically calculated for both Energy consumption and PMV. During run-time, as new data are accumulated, the scaling parameters are re-computed, ensuring that the scaling is up-to-date. The principle of the employed scaling the technique is shown Eq. 4 for an arbitrary function F. The variable μ^k represents the current (k-th time-step) mean value of F, while δ^k its standard deviation. Similarly, Eq. 5 and 6 indicate how these values are iteratively updated in preceding steps.

$$F_{norm} = \frac{F - \mu^k}{\delta^k} \tag{4}$$

$$\mu^k = \mu^{k-1} + \frac{F - \mu^{k-1}}{k} \tag{5}$$

$$\delta^k = \sqrt{\frac{s^k}{k-1}}, \quad s^k = s^{k-1} + (F - \mu^{k-1})(F - \mu^k) \tag{6}$$

A critical aspect of effective RL is to determine the range, within which the system operates in a acceptable way. This is more straightforward in other RL applications, such as autonomous driving where the automobile should stay within the limits of the road at all times [27]. We implement a similar strict zone of accepted system function by considering limiting the Smart Thermostat within the specified thermal comfort (PMV) limit. In other words, exceeding this limit is an unacceptable action by the agent. This zone of function is set to $|\text{PMV}| \leq 0.75$[2]. Consequently, an action is deemed terminal (corresponding to a *terminal cost*) if:

- the action led the PMV out of the zone of function
- the PMV was already out of the zone of function, and the action increased its absolute value

When terminal costs surface, the MLP is retrained and the next set-point is generated by the following *failsafe controller*:

if $PMV > 0$ **then**
  decrease indoor temperature by 1^oC
else
  increase indoor temperature by 1^oC
end if

This choice compensates for the agent's failure and is consistent with the general form of the agent's actions.

Given a state and the available actions, the agent has to produce a set-point which will minimize the expected return (the cumulative future costs in the time horizon determined by γ in Eq. 1). Due to the incremental nature of the thermostat's actions, we desire set-points optimized with a long-term horizon in mind. Consequently, for this case study γ was set equal to 0.98 to maximize performance.

As stated in Section 3, the Q function summarizes the expected benefit from a future action and this function is approximated by an MLP. The MLP weights are updated at the start of each day or whenever the agent has received a terminal cost. The data set used for training the MLP is extracted from TB in the form of (s, a) tuples. Denoting Q_k as the output of the *current* NN, the training targets, as defined by the NFQ framework, are given in Eq. 7.

$$target = c(s, a, s') + \gamma \cdot \min_a Q_k(s', a) \tag{7}$$

An important consideration stemming from the utilized approximations is the possibility of the controller to be trapped in in a sub-optimal solution because the controller's selected actions are based on transitions that were examined in the past. Consequently, a dilemma for each time-step is whether the controller will exploit its current knowledge, or it will seek a possibly better solution. This is the exploration/exploitation dilemma, since the controller cannot know the optimality of a certain action in a certain state if this action is never picked. In this work, we approach this dilemma via *ϵ-greedy action selection*, with ϵ being self-regulated as described in [31]. The "positive outcomes" are counted and then used to regulate ϵ. In our case, such positive outcomes are determined by the validity of the MLP's prediction. This validity is represented by the Temporal Difference (TD) error, defined in Eq.8.

$$TD = c(s, a, s') + \gamma \cdot \min_a Q(s', a) - Q(s, a) \tag{8}$$

A decision of the agent is deemed positive if $|TD| < 0.15$. On the one hand, this bound is tight enough to represent an accurate approximation. On the other hand, we observed that it is elastic enough to allow for iterative learning at early stages where the training error is initially very high. The association of the TD error with the agent's exploration mechanism was inspired by [18].

The set-point generation process is summarized as follows: the last taken action is evaluated. The values used for scaling the energy consumption and thermal comfort are updated, and so is the agent's knowledge (in the form of TB). The TD error is calculated, thus regulating the mechanism's exploration. If the last action was terminal, the MLP is retrained and the failsafe controller takes action. Otherwise, the next set-point is chosen via ϵ-greedily exploiting the current approximation of the Q function.

The control ensemble, illustrated in Figure 1, is repeated in period T, until the end of the schedule. The definition of this period is important as it affects the granularity of control as well as the computational requirements of the Smart Thermostat, thus leading to a trade-off. In the context of this work, T was set equal to 10 minutes so that the agent collects a greater amount of experience every day, which in turn results in faster learning. In addition, this time-step guarantees that if the agent makes a sub-optimal set-point designation (which is expected to happen, especially in the early stages of learning), it will not affect the occupants for too long.

5 EXPERIMENTAL SETUP

The effectiveness of the proposed control logic was evaluated using a well-known simulation and testing testbed provided by [20, 29]. Figure 2 illustrates an overview of the testbed, which has been used in a variety of works [12, 23]. The building dynamics and input sensor data for the controller are produced by the EnergyPlus suite [13]. The controller gathers this data and calculates the set-point through MATLAB. Data exchange is facilitated through the BCVTB (Building Controls Virtual TestBed) [34]. The employed building model corresponds to an actual building located in Crete, Greece[3]: The utilized weather data correspond to publicly available information collected in 2010.

The smart thermostat demonstrated in this paper targets a single, randomly occupied thermal zone of the building, active from 6:00 to 21:00. It is also assumed that, during the daily schedule, the zone is occupied at all times from at least one person.

[2]This threshold is the acceptable limit due to the EN15251 European standard.

[3]Building models were part of the PEBBLE FP7 EU project (grant agreement 248537).

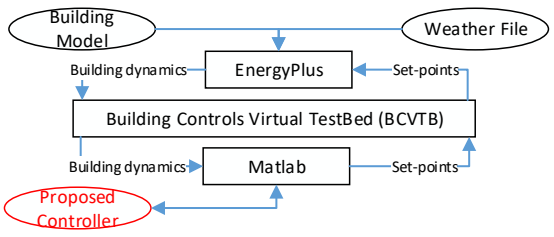

Figure 2: Simulation testbed of proposed controller

To evaluate the thermostat's performance, the resulting energy consumption and thermal comfort are compared with a wide, reasonable array of rule-based control set-points (RBC's). This is a typical function found in all the cooling/heating devices for setting a "static" temperature set-point. For the sake of completeness, we select to provide the performance results achieved with the usage of alternative RBCs as a reference, in order to highlight the enhancement achieved with the proposed solution. Most manual thermostats tend to operate in a single heating set-point in winter, and a respective cooling set-point in summer [32]. Similarly, other smart thermostats also produce set-points in a range deemed reasonable in regard of thermal comfort (in this case, from 20^oC to 27^oC). Fluctuations in these set-points do exist, but still the result of these fluctuations would be a trajectory varying between the RBC set-points. By including as many of them as possible, a meaningful assessment against typical user or smart control is ensured.

6 EXPERIMENTAL EVALUATION

The first experiment, summarized in Figure 3, evaluates the proposed controller's efficiency against typical RBC values, for a typical summer day, concerning the two basic metrics: energy consumption and thermal comfort. The controller actually verges on the ideal comfort level for $tr = 0$ and leads to less consumption for $tr = 1$. Additional results concerning two three-month periods, one in winter (January to March) and one in summer (June to August), are summarized in Table 1. The proposed controller achieves up to 59.2% mean energy savings (for $tr = 1$) and up to 41.8% comfort savings (for $tr = 0$) on average. Regarding the learning performance, it is shown that the worst-case scenario requires on average only $303/90 = 3.37$ training sessions per day.

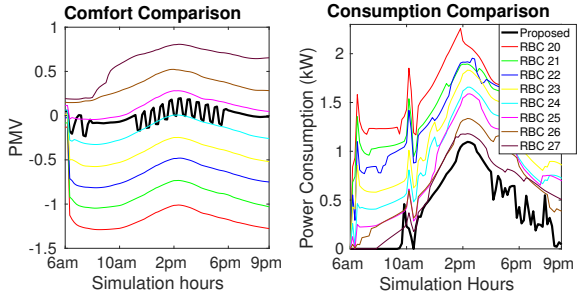

Figure 3: Daily performance vs RBCs (left $tr = 0$, right $tr = 1$)

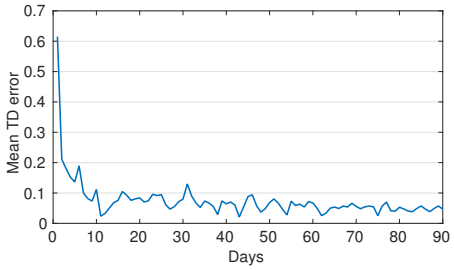

Figure 4: Daily mean MLP TD-error.

The mean TD error (Eq. 8) for each of the first 90 days is plotted in Figure 4. These results confirm previous evidence about improving the controller's efficiency over time, as the machine learning part of the controller leads to lower error values. The majority of these values is less than 0.15, which has been defined in Section 4 as the threshold for considering the model as successful.

Expressing the optimization objective as a weighted sum, enables the designer to designate preference with respect to the importance of each objective. To abide by this functionality, the critical component of our design is its dynamic scaling part, which normalizes the values of the objectives so that the weighted factors dominate the calculated cost. In the experiment illustrated in Figure 5 we study this ability according to different values of the trade-off factor tr. We observe that according to its value the system emphasizes in one of the two metrics. For example in the case of $tr = 0$ the controller leads to best comfort values, while in the case of $tr = 1$ the controller minimizes the energy consumption. The study of Figure 5 highlights that setting $tr = 0.5$, actually leads to results, where both the energy cost and the occupants thermal comfort metrics are of equal importance. It is also apparent that the thermostat's performance improves over time.

Last, it is important to quantify the ability of the proposed framework to support online execution on small-factor, resource constrained embedded device. Towards this direction, the proposed control logic ensemble was evaluated on two well-known, single-core embedded devices, a BeagleBoard xm (ARM37x Cortex-A8@1GHz) and a Raspberry Pi Zero (ARMv6 BCM2835@1GHz). We focus our analysis on the average execution latency for the training and prediction of the utilized MLP Neural Network, since these are the

Learning Performance (winter/summer)		
Trade-off	Retrain sessions	
0 (Optimize Comfort)	180 / 112	
0.5 (Optimize both)	207 / 157	
1 (Optimize Energy)	303 / 223	
Control Performance (winter/summer)		
Trade-off	Mean energy savings (vs RBCs)	Mean comfort savings (vs RBCs)
0 (Optimize Comfort)	-7.8% / 10.8%	**11.9%** / **41.8%**
0.5 (Optimize both)	**28.4%** / **32.4%**	**-3.9%** / **27.4%**
1 (Optimize Energy)	**59.2%** / **48.3%**	-23.3% / 5.8%

Table 1: Evaluation of learning and control performance.

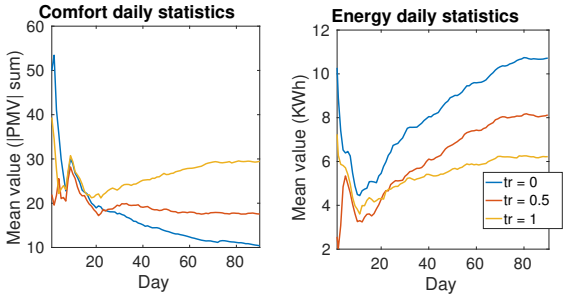

Figure 5: Efficiency of dynamic scaling to satisfy trade-off

most computationally demanding tasks of the proposed controller. The average training latency for a batch of 4000 transitions (around 1.5 months of function) on the two devices was 48.94 and 64.32 seconds, respectively. The corresponding prediction latency was measured at 0.0013 and 0.0025 seconds. These results emphasize the feasibility of the proposed controller's implementation on an embedded platform and are attributed to the simple nature of the utilized Neural Network.

7 CONCLUSIONS

This paper has introduced a model-free, plug&play approach on the problem of an HVAC thermostat's set-point scheduling. Through reinforcement learning, the controller adapts and improves its performance over time. The user can set the preferred balance in energy savings and thermal comfort. The proposed controller is lightweight and can be implemented in low-cost embedded devices. The solution is demonstrated using a well-known simulation and testing framework and is implemented in ARM-based microprocessors.

Acknowledgments Work in this paper has been partially funded by the European Union's Horizon 2020 research and innovation programme, under project SDK4ED, grant agreement No 780572.

REFERENCES

[1] Abdul Afram and Farrokh Janabi-Sharifi. 2014. Theory and applications of HVAC control systems–A review of model predictive control (MPC). *Building and Environment* 72 (2014), 343–355.
[2] Rafael AlcalÃą, Jorge Casillas, Oscar CordÃşn, Antonio GonzÃąlez, and Francisco Herrera. 2005. A genetic rule weighting and selection process for fuzzy control of heating, ventilating and air conditioning systems. *Engineering Applications of Artificial Intelligence* 18, 3 (2005), 279 – 296.
[3] Plamen P Angelov and Richard A Buswell. 2003. Automatic generation of fuzzy rule-based models from data by genetic algorithms. *Information Sciences* 150, 1-2 (2003), 17–31.
[4] Amjad Anvari-Moghaddam, Hassan Monsef, and Ashkan Rahimi-Kian. 2015. Optimal smart home energy management considering energy saving and a comfortable lifestyle. *IEEE Transactions on Smart Grid* 6, 1 (2015), 324–332.
[5] Enda Barrett and Stephen Linder. 2015. Autonomous hvac control, a reinforcement learning approach. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 3–19.
[6] Sahand Behboodi, David P Chassin, Ned Djilali, and Curran Crawford. 2018. Transactive control of fast-acting demand response based on thermostatic loads in real-time retail electricity markets. *Applied Energy* 210 (2018), 1310–1320.
[7] Abdullatif E Ben-Nakhi and Mohamed A Mahmoud. 2002. Energy conservation in buildings through efficient A/C control using neural networks. *Applied Energy* 73, 1 (2002), 5–23.
[8] Danushka Bollegala. 2017. Dynamic feature scaling for online learning of binary classifiers. *Knowledge-Based Systems* 129 (2017), 97–105.
[9] Francesco Calvino, Maria La Gennusa, Gianfranco Rizzo, and Gianluca Scaccianoce. 2004. The control of indoor thermal comfort conditions: introducing a fuzzy adaptive controller. *Energy and buildings* 36, 2 (2004), 97–102.
[10] David P Chassin, Jakob Stoustrup, Panajotis Agathoklis, and Nedjib Djilali. 2015. A new thermostat for real-time price demand response: Cost, comfort and energy impacts of discrete-time control without deadband. *Applied Energy* 155 (2015), 816–825.
[11] Konstantinos Dalamagkidis, Denia Kolokotsa, Konstantinos Kalaitzakis, and George S Stavrakakis. 2007. Reinforcement learning for energy conservation and comfort in buildings. *Building and environment* 42, 7 (2007), 2686–2698.
[12] Panayiotis Danassis, Kostas Siozios, Christos Korkas, Dimitrios Soudris, and Elias Kosmatopoulos. 2017. A low-complexity control mechanism targeting smart thermostats. *Energy and Buildings* 139 (2017), 340–350.
[13] U. S Department of Energy. 2015. EnergyPlus Energy Simulation Software. http://apps1.eere.energy.gov/buildings/energyplus/.
[14] Anastasios I Dounis and Christos Caraiscos. 2009. Advanced control systems engineering for energy and comfort management in a building environmentâĂŤA review. *Renewable and Sustainable Energy Reviews* 13, 6-7 (2009), 1246–1261.
[15] E.U. Commission. 2008. European energy and transport - Trends to 2030 (Update 2007). http://aei.pitt.edu/46140/
[16] Eurostat. 2005. Energy balance sheets. Data 2002-âĂŞ2003, Luxemburg.
[17] Poul O Fanger et al. 1970. Thermal comfort. Analysis and applications in environmental engineering. *Thermal comfort. Analysis and applications in environmental engineering*. (1970).
[18] Clement Gehring and Doina Precup. 2013. Smart exploration in reinforcement learning using absolute temporal difference errors. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1037–1044.
[19] D Kolokotsa, GS Stavrakakis, K Kalaitzakis, and D Agoris. 2002. Genetic algorithms optimized fuzzy controller for the indoor environmental management in buildings implemented using PLC and local operating networks. *Engineering Applications of Artificial Intelligence* 15, 5 (2002), 417–428.
[20] GD Kontes, GI Giannakis, Elias B Kosmatopoulos, and DV Rovas. 2012. Adaptive-fine tuning of building energy management systems using co-simulation. In *Control Applications (CCA), 2012 IEEE International Conference on*. IEEE, 1664–1669.
[21] Rajesh Kumar, R.K. Aggarwal, and J.D. Sharma. 2013. Energy analysis of a building using artificial neural network: A review. *Energy and Buildings* 65 (2013), 352 – 358. https://doi.org/10.1016/j.enbuild.2013.06.007
[22] L Magni, Giuseppe De Nicolao, Lorenza Magnani, and Riccardo Scattolini. 2001. A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica* 37, 9 (2001), 1351–1362.
[23] Charalampos Marantos, Kostas Siozios, and Dimitrios Soudris. 2017. A Flexible Decision-Making Mechanism Targeting Smart Thermostats. *IEEE Embedded Systems Letters* 9, 4 (2017), 105–108.
[24] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. 2000. Constrained model predictive control: Stability and optimality. *Automatica* 36, 6 (2000), 789–814.
[25] Daniele Menniti, Ferdinando Costanzo, Nadia Scordino, and Nicola Sorrentino. 2009. Purchase-bidding strategies of an energy coalition with demand-response capabilities. *IEEE Transactions on Power Systems* 24, 3 (2009), 1241–1255.
[26] Martin Riedmiller. 2005. Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*. Springer, 317–328.
[27] Martin Riedmiller, Mike Montemerlo, and Hendrik Dahlkamp. 2007. Learning to drive a real car in 20 minutes. In *Frontiers in the Convergence of Bioscience and Information Technologies, 2007. FBIT 2007.* IEEE, 645–650.
[28] Luigi Rucco, Andrea Bonarini, Carlo Brandolese, and William Fornaciari. 2013. A bird's eye view on reinforcement learning approaches for power management in WSNs. In *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*. IEEE, 1–8.
[29] Carina Sagerschnig, Dimitrios Gyalistras, Axel Seerig, Samuel Prívara, Jiří Cigler, and Zdenek Vana. 2011. Co-simulation for building controller development: The case study of a modern office building. In *Proc. CISBAT*. 14–16.
[30] Jagdev Singh, Nirmal Singh, and JK Sharma. 2006. Fuzzy modeling and control of HVAC systems–A review. (2006).
[31] Teck-Hou Teng, Ah-Hwee Tan, and Yuan-Sin Tan. 2012. Self-regulating action exploration in reinforcement learning. *Procedia Computer Science* 13 (2012), 18–30.
[32] Bryan Urban and Kurt Roth. 2014. A Data-Driven Framework For Comparing Residential Thermostat Energy Performance.
[33] Tianshu Wei, Yanzhi Wang, and Qi Zhu. 2017. Deep reinforcement learning for building HVAC control. In *Design Automation Conference (DAC), 2017 54th ACM/EDAC/IEEE*. IEEE, 1–6.
[34] Michael Wetter. 2011. Co-simulation of building energy and control systems with the Building Controls Virtual Test Bed. *Journal of Building Performance Simulation* 4, 3 (2011), 185–203.
[35] Ji Hoon Yoon, Ross Baldick, and Atila Novoselac. 2014. Dynamic demand response controller based on real-time retail price for residential buildings. *IEEE Transactions on Smart Grid* 5, 1 (2014), 121–129.