Deep Reinforcement Learning for Residential HVAC Control with Consideration of Human Occupancy

Evan McKee, Yan Du, Fangxing Li
The University of Tennessee
Knoxville, Tennessee, USA
{emckee5, ydu15}@vols.utk.edu, fli6@utk.edu

Jeffrey Munk, Travis Johnston,
Kuldeep Kurte, Olivera Kotevska, Kadir Amasyali, Helia
Zandi
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
{munkjd, johnstonjt, kurtekr, kotevskao, amasyalik,
zandih}@ornl.gov

***Abstract*—The Artificial Intelligence (AI) development described herein uses model-free Deep Reinforcement Learning (DRL) to minimize energy cost during residential heating, ventilation, and air conditioning (HVAC) operation. Building cooling loads and HVAC operation are difficult to accurately model due to complexity, lack of measurements and data, and model specific performance, so online machine learning is used to allow for real-time readjustment in performance. Energy costs for the multi-zone cooling unit shown in this work are minimized by scheduling on/off commands around dynamic prices. By taking advantage of precooling events that take place when the price is low, the agent is able to reduce operational cost without violating user comfort. The DRL controller was tested in simulation where the learner achieved a 43.89% cost reduction when compared to traditional, fixed-setpoint operation. The system is now ready for the next phase of testing in a live, real-time home environment.**

***Index Terms*—Automation, demand response, machine learning, transactive control, smart grid.**

I. Introduction

The genesis point of this research is the advent of demand response pricing environments (DRE) throughout the U.S. In DRE's, utilities attempt to use dynamic pricing to influence consumer behavior with the goal of reducing spinning reserve while maintaining frequency stability [1]. In a fixed energy pricing environment, there are no financial incentives to strategic load scheduling. The introduction of DRE's allows for strategic choices that can result in benefits for both homeowners and utilities. For example, a homeowner can use more power when the price of electricity is low, and less at high price. For a single home, an inhabitant could determine the optimal scheduling themselves, but only with a significant commitment of time and calculation.

Automation presents a more favorable alternative. Smart Home Energy Management Systems (SHEMS) allow for the automatic activation and deactivation of devices throughout a home in accordance with some schedule [2]. The Heating, Ventilation, and Air Conditioning (HVAC) appliance is an ideal candidate for automation due to its intermittent use and hands-free operation. Energy consumed by the HVAC system of a home accounts for approximately 50% of total energy usage [3]. Attempts to automate air conditioner use through pre-optimized automation have met with some success – even just passively shifting AC operation to precool a room has resulted in a reduction in electricity bills [4].

HVAC automation for cost and energy savings has been a subject of research for over 20 years [5]. The earliest models used Knowledge-Based Systems (KBS) for predictive control [6] [7]. These were effective in providing cost savings, but required input from an expert in the field. Later developments used Model Predictive Control (MPC), which were trained in simulation [8] [9] [10]. These systems require an accurate model of the relevant building and HVAC system. Researchers have struggled with thermal building modeling because of its nonlinearity and strong specificity of application [5]. As machine learning optimization evolved, the methods of automation evolved as well. The most complex methods used distributed AI systems, and required sensors not readily available to most homes [11] [12]. The majority of developments reported significant cost savings, but almost all were limited to a single application [5].

In recent works, Deep Reinforcement Learning (DRL) methods have been tested which allow for experiential control and online learning. Researchers in [3] simulated an environment using EnergyPlus software and used Deep Q-Learning to manage operation of two air conditioning zones. Their state included the time of day, zone temperature, outdoor temperature, and solar irradiance intensity. Their reward function used two terms: one to account for energy cost, and the other to account for comfort violation. In their environment, the price signal was dynamic, but consistent across different days. The experiment reported 11% energy savings over a rule-based baseline model. Their development was tested exclusively in simulation [3].

Another form of DRL is Deep Deterministic Policy Gradient (DDPG). While Deep Q-Learning methods attempt

Notice: This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

978-1-7281-5508-1/20/$31.00 ©2020 IEEE

to approximate and generalize a Q-Table, policy gradient methods attempt to approximate the policy. In [13], DDPG was used as the learning algorithm, to account for the expanding dimensionality of the action space when more zones are added. The action space was continuous, and included setpoints for humidity as well as temperature. The state included indoor and outdoor temperature and humidity, and the reward function was similar to the two-term reward system examined in [3]. The system reported a lower cost and faster learn rate than using a Deep Q Neural Network [13]. This could be due to the fact that humidity control is more appropriately suited to a continuous action space. As before, their development was only tested in simulation. In a continuation of the work described herein, our DRL controller will be placed in a live home and some of the claims made in literature will be subjected to a field test.

II. Model Formulation

A. *HVAC Modeling Challenges*

When modeling a building in simulation, each room has its own thermal profile that interacts with adjacent rooms and the outside, and air flow between rooms must be addressed [14]. These processes create a tangled web of nonlinear interactions. Within this section, we will use “thermal mass”, a term that broadly refers to the amount that ambient temperature affects indoor home temperatures, as a catch-all for the black box of partially observable home attributes that influence the indoor temperature.

Accounting for the presence of thermal mass during a temperature prediction is the primary challenge of HVAC automation. A perfectly simulated model might yield an optimal solution, but we expect the thermal profile of a home to change over time. An MPC fine-tuned for a home built today could be unusable one year from now. Even the most thorough and accurate simulations have high specificity of application.

Reinforcement Learning (RL) presents an alternative to model-based systems by learning iteratively through interaction with the environment. RL has shown promise in game-based studies as well as in power systems applications [15]. With RL, no foreknowledge of values such as insulation coefficients or internal heat load is necessary. The learner simply senses the current state of its environment, makes decisions, updates its knowledge, and attempts to tune its decision-making strategy to maximize its cumulative reward. Instead of a simulated model which contains every measured variable, those attributes that are cost effective to measure are included in the state, and every other relationship will be accounted for by leveraging Deep Learning. A controller equipped with such an algorithm could be configured to learn indefinitely inside a home and take self-corrective actions until it has approximated the lowest cost HVAC operation.

To expedite the process of algorithm development, a building and HVAC model was developed in Python. This model served as the environment for the RL agent. The model was trained and validated based on data from a highly instrumented unoccupied research house [14]. The data presented herein is 2-zone HVAC system control using this model as the environment.

The overall problem can be modeled as a constrained optimization problem: minimize cost of operation, while maintaining user comfort. Cost of operation is defined as the instantaneous price of energy times power consumption. Comfort is satisfied if two conditions are met:

- The AC never runs if indoor temperature is 0.5 °C below the customer’s lower preference.
- The AC runs continually if indoor temperature is 0.5 °C above the customer’s upper preference.

B. *Reinforcement Learning*

RL is a branch of machine learning that studies the conditioning of a learning agent towards accomplishing some goal through rewards and punishments [16]. At every iteration, the learner (agent) takes an action and is given a positive or negative reward. The agent is not told which action to take, but must discover the maximum long-term reward yielding actions through trial and error. RL works best when a problem can be modeled as a Markov Decision Process (MDP), which brings the problem into the scope of the Bellman Optimality Equation.

The simplified, recursive Bellman Equation is

$$V(s) = \max_{a}[R(s,a) + \gamma V(s')] , \qquad (1)$$

where *s* is the current state, *s’* is the next state having taken action *a*, *V(s)* is the current state value, γ is the discount rate, *R(s,a)* is the reward (having taken action *a* from state *s*), and *V(s’)* is the next state value. This equation links the states of an MDP together and gives the agent a roadmap for deciding the next action from its current state. The state values do not depend solely on instantaneous reward, but on the expected return of the entire trajectory. Solving this equation yields *V*, the value function. In an MDP, the next state is dependent on the current state and action, but independent of all previous state-action pairs [16].

After visiting a state and taking an action, the agent calculates an updated value for that state-action pair in accordance with the chosen algorithm. After a sufficient period of training has elapsed, the agent attempts to converge at an optimal policy that indicates the best action to take from each state for maximum expected return. The time taken to converge is the learning rate of that algorithm. The program may then output a value function *V*, which is a dataset of each state paired with the expected return from that state, a policy function *π* which shows each state and the recommended action for maximum return, or an action value function *Q* which shows each state and the value of taking each possible action from that state.

Three auxiliary variables are common to RL algorithms: 1) the step-size parameter α influences the learning rate by prioritizing recently-learned information over old data; 2) the probability ε guarantees exploration in the commonly used εgreedy approach by granting the agent a probability ε of taking a randomly selected action; 3) a discount rate γ must be applied to the rewards so that their sum will approach a number other than infinity, if the task to be accomplished is continuous [16].

Ultimately, this problem can be modeled as a partially observable MDP because each transition probability is dependent only on the present state, but thermal mass is hidden from the observation. Therefore, RL can be applied to this problem.

III. Solution Architecture

A. Reinforcement Learning Architecture

The architecture of the final RL model is reported here. The state was made up of the inputs shown in Table I. All features were normalized before being recorded as an observation.

TABLE I. STATE USED DURING TWO-ZONE CONTROL

Feature	Title	Function
1	$Z1_T$	$z1_T - Upper$
2	$Z2_T$	$z2_T - Upper$
3	O_T	O_T
4	P_1	$Price(t)$
5	P_2	$Price(t + 5)$
6	P_3	$Price(t + 15)$
7	P_4	$Price(t + 30)$

The two indoor zone temperatures are represented, as well as outdoor temperature and four price values. P_1 measures instantaneous price, while P_2, P_3, and P_4 are forecasted price values in \$/kWh for the next 5, 15, and 30 minutes, respectively. Zone temperatures are recorded as the thermostat temperature minus the upper temperature preference of the customer.

The action space is all binary combinations of “Off” and “On” for each zone. For two zones, the space [0, 1, 2, 3] corresponds to [“Off/Off”, “Off/On”, “On/Off”, “On/On”]. Calibration and parameterization were used to realize the modified two-term reward system

$$R_t = -\lambda P_t C_t - V_t, \tag{2}$$

where P_t is the energy price over time t, C_t is the consumption over time t, and V_t is the amount any indoor zone temperature violated that comfort zone, in degrees Celsius. Since only cooling was tested, the system only counted a comfort violation above the upper preference point. λ is used as a weighting factor that prioritizes cost over comfort. A λ of 100 was used for the final system.

The system is able to save money over a naïve, fixed-setpoint baseline because it practices precooling. The price forecasting features warn the DRL controller of an upcoming price increase. The controller runs the HVAC while the price is low, then remains off after a price increase.

B. Neural Network Architecture

The learning algorithm used in this work is a Deep Q Neural Network (DQN). This algorithm uses an evaluation neural network to estimate the values of a Q Table with each iteration. A second neural network runs in parallel to give the first network a convergence target. The system employs one evaluation and one target network running in parallel. DQN provides a way to execute RL control objectives while leveraging the optimization power of deep learning. Minh et al. were able to show that the use of two networks could offer stability and reduce potential oscillations during training [17]. Table II shows some of the algorithm parameters used.

TABLE II. DQN PARAMETERS

NN Learn Rate	0.01
Input Layers	1x7, one input per feature
Hidden Layers	2x10 (2 layers with 10 neurons each)
Output Layers	1x4, one output per action
Reward Decay (γ)	0.9
Epsilon (ε)	0.1
Memory Size (Experience Replay Memory)	20,000
Batch Size	32
Initial Iterations	200
Δt_c	300
Optimizer	AdamOptimizer

Fig. 1 shows the pseudocode for operation.

```
Reserve and initialize replay memory $MB$
Initialize evaluation network $Q$ with random weights $\theta$
Initialize target network $\hat{Q}$ with random weights $\hat{\theta}$
$N$ = Maximum number of episodes
$TS_{max}$= Maximum number of time steps
$k$ = Minimum cycle time

for episode = 1 to $N$:
    Reset building environment $env$
    $S_{pre}$ = GetInitialObservation($env$)
    $a$ = GetInitialAction($env$)
    for $ts$ = 1 to $TS_{max}$:
        if $ts$ % $k$ == 0:
            $j = ts / k$
            $S_{curr}$ = GetCurrentObservation($env, ts$)
            $r$ = GetReward($env, S_{pre}, a, S_{curr}$)
            $MB \leftarrow$ StoreTransition([$S_{pre}, a, r, S_{curr}$])
            Sample random mini-batch($s_j, a_j, r_j, s_{j+1}$) from $MB$
            if episode terminates at $j$+1:
                $y_j = r_j$
            else:
                $y_j$ = Q-Learning update using $\hat{Q}$ and $\hat{\theta}$
            end if
            $L$ = MSE loss between $y_j$ and $Q$ over all $j$
            $\theta = \theta - \alpha\ (dL/d\theta)$
            Every $\Delta t_c$ time steps, $\hat{\theta} = \theta$
            $S_{pre} = S_{curr}$
        end if
        env = UpdateEnvironment($env, ts, a$)
    end for
end for
```

Figure 1. DQN Algorithm pseudocode for evaluation and target networks [18].

IV. Case Study

Since the eventual goal is the incorporation of the DRL controller into a residential home, some calibration is necessary to facilitate operation outside of a simulated environment. The following section describes enhancements to the DRL controller made to consider accommodation of a human occupant, the customer.

A. Setpoint Governance

In our model, the system is controlled by translating "On/Off" commands into setpoints above or below the indoor temperature. However, the controller is prohibited from submitting a setpoint which is outside customer's comfort preference. Instead of the DRL controller being responsible for comfort regulation, it is allowed to focus on price and monetary savings. There are several benefits to this constraint:

- The model under setpoint governance produces far lower cost operation, up to 40% less per month.
- There are fewer states to learn, increasing learning rate.
- The amount of comfort violations drops to zero above the customer's upper limit.
- The customer never observes a thermostat setpoint outside of preference.
- In case of equipment failure or loss of communications, the setpoint left behind is always within comfort.

B. Relative Temperature Recording

In the upper half of Fig. 2, the indoor zone temperatures have been recorded in the state as empirical values, but the customer's temperature preferences have changed four times over the course of a month. When only empirical temperatures are considered, the DRL controller's learning rate suffers and the indoor temperature strays outside the customer's comfort boundaries. To better demonstrate this anomaly, setpoint governance has been disengaged.

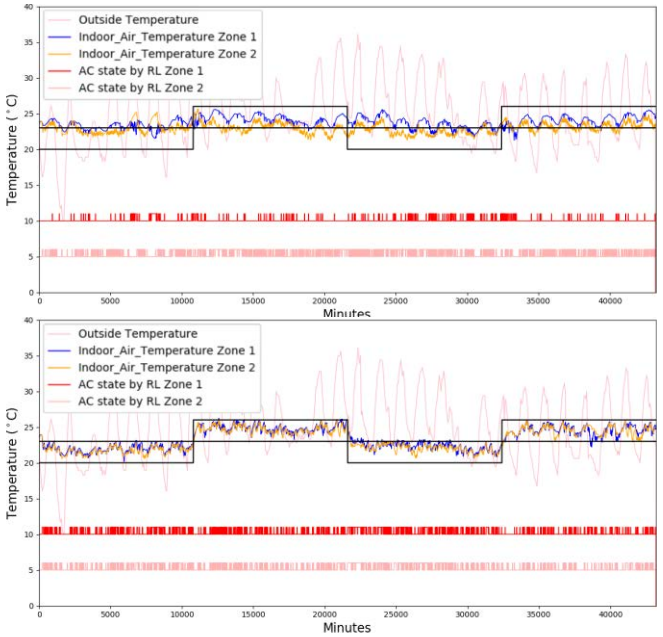

Figure 2. Absolute (top) vs. relative (bottom) temperature recording.

When the temperatures are instead recorded as zone temperature minus the customer's upper limit, the system is better able to stay within comfort, as the lower half of Fig. 2 shows. Accommodation of sudden and frequent customer preference changes are necessary during live occupancy. Even when setpoint governance is enforced, the observation made by the DRL controller must include information about both the indoor temperature and the customer's upper preference in order to properly function.

C. Results

For the following validation experiment, our DRL controller was trained for one month using outdoor temperatures from June 2018, then validated on July. During this month, it outperformed the fixed-setpoint baseline by 43.89%. Parameters and results are shown in Table III.

TABLE III. Validation Results

Comfort Range	21 – 24 ° C
Minimum Cycle Time	15 minutes
Price Range	\$0.05-\$0.25/kWh
Alternating Price Period	6 hours
Apparent Convergence	10 days
Monthly Cost	\$26.68
July Baseline Cost	\$47.55
% Savings	43.89

V. Conclusions

In this work, a flexible model-free RL optimization is developed for creating the optimal schedule for an HVAC system. The goal is to minimize energy cost, while maximizing the comfort of the residents. The development was able to save 43.89% in a simulated home environment over a fixed setpoint baseline, while maintaining comfort throughout. Due to the consistent money-saving performance even during cold months, and since setpoint governance acts as a failsafe for preserving comfort, this model could be implemented into homes as it is today with confidence that customers could save money and maintain comfort. In future tests, heating will be incorporated, as well as the management of other loads. The development is ready for testing in a live home environment.

ACKNOWLEDGMENT

This work was funded by the U.S. Department of Energy, Energy Efficiency, and Renewable Energy, Building Technology Office under contract number DE-AC05-00OR22725.

This work would not have been possible without the continued support and sponsorship of Oak Ridge National Laboratory and its affiliates. CURENT and its affiliates, including NSF, are gratefully acknowledged.

REFERENCES

[1] Shi, Q., Li, F. and Cui, H., “Analytical method to aggregate multi-machine SFR model with applications in power system dynamic studies,” IEEE Transactions on Power Systems, 33(6), pp.6355-6367, 2018.

[2] [1] H. Zandi, T. Kuruganti, E. Vineyard, and D. Fugate, “Home Energy Management Systems: An Overview,” in 9th International Conference on Energy Efficiency in Domestic Appliances and Lighting (EEDAL2017), 2017, no. 2018, pp. 605–614.

[3] T. Wei, Y. Wang and Q. Zhu, "Deep reinforcement learning for building hvac control," Proceedings of the 54th Annual Design Automation Conference, p. 22, 2017.

[4] A. Aribali, M. Ghofrani, M. Etezadi-Amoli, M. S. Fadali and Y. Baghzouz, "Genetic-algorithm-based optimization approach for energy management," IEEE Transactions on Power Delivery, vol. 28, no. 1, pp. 162-170, 2012.

[5] C. Cheng and D. Lee, "Artificial intelligence-assisted heating ventilation and air conditioning and the unmet demand for sensors," Department of Energy and Refrigerating Air-Conditioning Engineering, National Taipei University of Technology, Taipei, 2019.

[6] G. Clark and P. Mehta, "Artificial intelligence and networking in integrated building management systems.," Automation in Construction, Vols. 6(5-6), pp. 481-498, 1997.

[7] F. Lara-Rosano and N. K. Valverde, "Knowledge-based systems for energy conservation programs.," Expert Systems with Applications, Vols. 1-2, no. 14, pp. 25-35, 1998.

[8] R. Godina, E. Rodrigues, E. Pouresmaeil, J. Matias and J. Catalao, "Model predictive control home energy management and optimization strategy with demand response," Applied Sciences, vol. 8, no. 3, p. 408, 2018.

[9] G. Serale, M. Fiorentini, A. Capozzoli, D. Bernardini and A. Bemporad, "Model predictive control (MPC) for enhancing building and HVAC system energy efficiency: Problem formulation, applications and opportunities.," Energies, vol. 11, no. 3, p. 631, 2018.

[10] G. Huang, S. Wang and X. Xu, "A robust model predictive control strategy for improving the control performance of air-conditioning systems," Energy Conversion and Management, vol. 50, no. 10, pp. 2650-2658, 2009.

[11] I. Petri, H. Li, Y. Rezgui, Y. Chunfeng, B. Yuce and B. Jayan, "A modular optimisation model for reducing energy consumption in large scale building facilities," Renewable and Sustainable Energy Reviews, vol. 38, pp. 990-1002, 2014.

[12] A. Gonzalez-Briones, J. Prieto, F. De La Prieta, E. Herrera-Viedma and J. Corchado, "Energy optimization using a case-based reasoning strategy," Sensors, vol. 18, no. 3, p. 865, 2018.

[13] G. Gao, J. Li and Y. Wen, "Energy-efficient thermal comfort control in smart buildings via deep reinforcement learning," CoRR, vol. 1901.04963, 2019.

[14] B. Cui, J. Munk, R. Jackson, D. Fugate and M. Starke, "Building thermal model development of typical house in u.s. for virtual storage control of aggregated building loads based on limited available information," 2017 30th International Conference on Efficiency, Cost, Optimization, Stabilization, and Environmental Impact of Energy Systems, 2017.

[15] F. Li and Y. Du, “From AlphaGo to power system AI: What engineers can learn from solving the most complex board game,” IEEE Power and Energy Magazine, vol. 16, issue 2, pp. 76-84, March-April 2018.

[16] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, MIT press, 2018.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare and S. Petersen, "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, p. 529, 2015.

[18] Kotevska, O., Kurte, K., Munk, J., Johnston, T., McKee, E., Permulla, K., & Zandi, H.. "RL-HEMS: Reinforcement Learning-based Home Energy Management System for HVAC Energy Optimization". ASHRAE (2020).