Online HVAC Optimization under Comfort Constraints via Reinforcement Learning

Christian Stippel[a], Rafael Sterzinger[a], David Sengl[c]
Aleksey Bratukhin[d], Markus Kobelrausch[a], Stefan Wilker[a], Thilo Sauter[a,d]
[a]TU Wien, Institut for Computer Technology, Vienna, Austria, firstname.surname@tuwien.ac.at
[c]UAS Technikum Wien, Institute Renewable Energy Systems, Vienna, Austria, firstname.surname@technikum-wien.at
[d]University for Continuing Education Krems, Department for Integrated Sensor Systems, Wiener Neustadt, Austria

***Abstract*—This paper shows the capabilities of Reinforcement Learning to enhance the efficiency of heating, ventilation, and air conditioning systems within office buildings. Our research applies the precise management of temperature and humidity, fundamental control algorithms, and several other factors to reduce the building's power consumption while improving thermal comfort and air quality. We succeed in developing optimal control policies by employing Proximal Policy Optimization and Advantage Actor Critic. The outcomes of our research indicate that our RL framework substantially outperforms existing baselines in maintaining ideal humidity and temperature levels while achieving a notable reduction in energy consumption by 12% over seven years compared to the current static control logic employed in HVAC systems. The contributions of our research include introducing RL agents trained online for effective and economical HVAC control from day one and an underlying shared state embedding space to effectively understand the dynamics between various rooms. We compare our approach against four baseline control logics. Moreover, we show a novel socket communication protocol to seamlessly interact with TRNSYS18, a simulation environment that enables rapid training and evaluation of our agents.**

***Index Terms*—Online Reinforcement Learning, Advantage Actor-Critic, Proximal Policy Optimization, HVAC Systems, Building Automation, Power Consumption, Climate Change**

I. Introduction

Heating, Ventilation, and Air Conditioning (HVAC) systems ensure productivity and healthy indoor environments [1]. At the same time, they contribute to approximately 1,950 million tons of CO_2 annually or 3.94% of global greenhouse gas emissions [2]. With climate change and its inherent challenges, there is a pressing need for intelligent HVAC control strategies that optimize both indoor environmental conditions and total energy consumption.

In this work, we address these challenges by applying Reinforcement Learning (RL), a machine learning paradigm where a so-called agent learns to make decisions by interacting with its environment. In our proposed RL framework, our agent learns control strategies online, i.e. independent of historical data. This approach is designed to surpass traditional rule-based algorithms in efficiency and adaptability. Additionally, setting appropriate priors on the action space, i.e. restricting its possible actions within reasonable ranges, eases the deployment of our framework for newly installed HVAC systems, making operation cost-effective from day one.

To validate our approach, we integrate the Transient System Simulation Program (TRNSYS18), a simulation software to model an HVAC system. This integration enables rapid simulation via socket communication, which is essential for RL algorithms that require extensive sampling. Our simulation considers 21 distinct states and uses 9 continuous actions and 1 categorical action to regulate temperature, humidity, and air quality across multiple zones.

The following details our key contributions:

- Introduction of RL agents trained online for practical and cost-effective HVAC control from day one.
- Employment of a shared state embedding space to capture the influence of room onto others.
- Comparison of four baselines and two RL algorithms for continuous action spaces, namely A2C and PPO.
- Development of a fast socket communication protocol for seamless integration with TRNSYS18.

The remainder of this paper is organized as follows: Section II offers an overview of RL applications in HVAC systems and draws parallels. In Section III, we delve into our simulation environment, outlining its foundational elements and design principles, including socket communication, the state and action space, and the reward function. Section IV is dedicated to our RL agents, network design for actors/critics, the encoding of state information, and the restriction of action spaces. Next, we present our results in Section V and conclude our work with Section VI where we summarize our insights and discuss future research.

II. Related Work

Recent advancements in RL have led to its increased application in various domains, including HVAC systems in smart buildings. Along with its rise in popularity is the prominence of algorithms such as Advantage Actor-Critic (A2C) [3] and Proximal Policy Optimization (PPO) [4]. A2C's efficiency in policy gradient methods, achieved by decoupling policy and value estimations, offers a stable learning process, making it suitable for environments with multiple actions. On the other hand, PPO is recognized for its simplicity and effectiveness, utilizing a clipped objective to facilitate stable learning.

The primary objectives of optimizing HVAC control are to reduce energy consumption and improve user comfort, including thermal, air, acoustic, and visual comfort. Various building-specific applications range from single-family homes to school

buildings, universities, and offices within single- or multi-zone spaces. Zhou et al. [5] provide an overview of Machine Learning (ML) in the context of HVAC, including various building types, ML algorithms, and optimization criteria.

According to them, 95.59% of the literature deals with energy savings, 89.71% deals with optimizing thermal comfort, and 85.29% optimizes energy and thermal comfort together. However, typically less attention is paid to air quality in terms of CO_2 saturation. In contrast to this is our work in which we improve the efficiency of a multi-zone office building by stabilizing not only thermal comfort but also indoor air quality while optimizing energy consumption.

Regarding RL techniques for optimizing energy consumption in HVAC systems, Xi Fang et al. [6] employed DQN to adjust the air supply temperature and chilled water temperature setpoints in a variable air volume system, balancing energy consumption and thermal comfort. Similarly, Ruihua Ding et al. [7] applied a DQN-based approach in a data center setting, focusing on energy consumption and cabinet air inlet temperature.

Other studies by Yan Du et al. [8], Zhiang Zhang et al. [9], Marco Biemann et al. [10], Guanyu Gao et al. [11], and Yiqun Pan et al. [12] have explored various HVAC systems, settings, and control actions, employing different learning algorithms such as DDPG and A3C, each with unique optimization objectives. Gao et al. [11] proposed a DRL-based framework for thermal comfort control using DDPG, demonstrating improvements in thermal comfort prediction and HVAC energy consumption reduction. Further, Li et al. [13] introduced a multiagent DRL framework for multizone thermal control, where each zone is represented as an agent, leading to significant energy savings and optimal thermal comfort across different zones. The preference for reinforcement learning, despite the insights provided by approaches such as Model Predictive Control, discussed in Taheri et al. [14], is primarily driven by our target to operate effectively with minimal initial system knowledge, adapting and optimizing policies through continuous interaction with the environment.

The following literature adheres to the same optimization criteria as ours: Valladares et al. [15] utilize an improved DQN-based DRL strategy to optimize energy consumption, thermal comfort, and indoor air quality, while in contrast to us, only consider a single zone in the form of a classroom, significantly simplifying the optimization problem. Yu et al. [16] proposed a multi-actor-attention-critic approach for a commercial building with multiple zones where energy, thermal comfort, and indoor air quality are included in their optimization. In comparison, our approach involves utilizing a single actor-critic method to explore multiple zones, making use of a shared embedding space, reducing the model architecture drastically.

III. Simulation Environment

The HVAC system that we consider in this study is a ventilation unit in an office building comprising a heating/cooling register and a humidifier. It is responsible for providing heating, ventilation, and cooling and, therefore, covers the whole spectrum of climatization.

Modeling and simulating such a system requires sophisticated tools that accurately capture HVAC dynamics. Therefore, we utilized TRNSYS18, a robust, flexible simulation tool well-suited for detailed energy system simulations. This platform allows us to generate a physics-based model of the HVAC system and simulate its interactions with the building and the external environment.

A. Socket Communication

To enable real-time interactions between the RL agent and the TRNSYS18 simulation environment, we implement direct communication via Protocol Buffers over socket communication in Python, utilizing the Type277 component, provided by Farhad Omar [17] for TRNSYS18, which is designed for dynamic data exchange. By establishing a direct socket connection, we circumvent the overhead associated with other approaches such as database-based communication systems [11].

B. State & Action Space

Our state space encompasses a selection of environmental variables, e.g. indoor and outdoor conditions such as temperature, humidity, and CO_2, as well as system variables, e.g. airflow rates and power consumption. A more extensive overview is provided by Table I.

TABLE I
State Space for Multi-Zone HVAC Control

Parameter	Unit	Description
T_Amb	°C	Outdoor Temperature
h_Amb	%	Outdoor Humidity
CO2_ODA	ppm	CO_2-Level of Outdoor Air
T_OP_i	°C	Operational Temperature in Zone i
h_Z_i	%	Relative Humidity in Zone i
CO2_Z_i	ppm	CO_2-Level in Zone i
VFR_Z_i	m³/h	Volume Flow Supply and Extract Air in Zone i
qH_Z_i	W/m²	Specific Heating Demand in Zone i (Radiators)
Q_Z_i	W	Power Consumption in Zone i

Our action space includes control variables such as, setting temperature and humidity setpoints for individual zones. Additionally, the agent can control the systems behaviour by selecting one of three control logics:

Control Logic 1, is a time-based control logic and where the ventilation unit is only operating during work hours, i.e. from 7 AM to 5 PM. Afterward, the fan is switched off and no air is supplied to the room anymore, meaning no thermal energy is added.

Control Logic 2, is temperature-based and mostly used in office buildings without standard working hours. Its goal is to keep the room temperature at a fixed setpoint even if no person is present in the building. This strategy results in a generally higher energy demand than Control Logic 1 since the temperature is kept stable during the night.

Lastly, **Control Logic 3** is a CO_2-based logic. Depending on the CO_2-level, the ventilation system becomes active, thus only

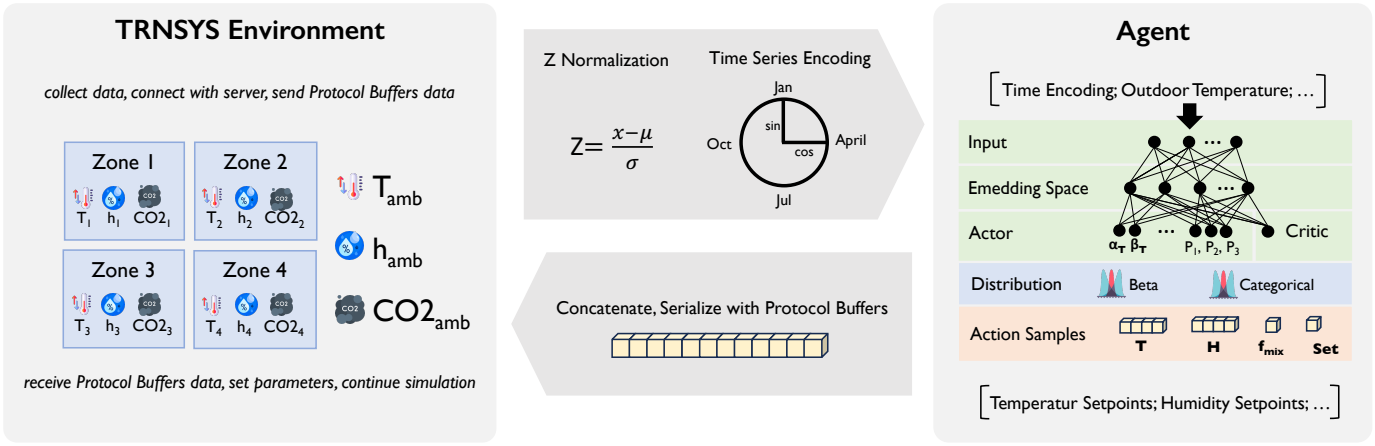

Fig. 1. Overview of our proposed methodology, illustrating the integration of TRNSYS18, data pre- and postprocessing and the interaction with the RL agent.

providing fresh air when the building is occupied. While this is the most energy-efficient control logic, it neglects temperature as long as there is sufficient fresh air, completely ignoring comfort. Additionally, to implement this control logic, CO_2-sensors need to be installed which is accompanied by additional costs.

In Table II, we provide a more extensive list of our actions.

TABLE II
ACTION SPACE FOR MULTI-ZONE HVAC CONTROL

Short	Unit	Description
T_Set_Z_i	°C	Temperature Setpoint for Zone i
Hum_Set_Z_i	%	Relative Humidity Setpoint in Zone i
f_Mix_RCA	%	Fraction of Recirculating Air
Set_Cont	$1, 2, 3$	Control Logic: Schedule, Temperature, CO_2

C. Reward Function

The reward function plays a vital role in our simulation environment as it allows the assessment of different optimization algorithms by quantifying the preferability over different HVAC states. Furthermore, it guides the learning process of our RL agent by punishing/rewarding its actions correspondingly.

In detail, our reward function is designed to balance energy efficiency with comfort which is dependent on several factors, including temperature, humidity, and CO_2-levels (see [1] for more details). Compared to similar works, we include CO_2in our function [5]. Subsequently, we will go further into detail about the different components comprising the reward function.

Firstly, we address energy efficiency by considering the power consumption of the HVAC system. We define the power reward component to be the negative total power consumption normalized by some constant to incentivize energy saving. Ideally, this constant should be aligned with the remaining components for which we found the median to work best. Our power reward component is defined as follows:

$$P(q) = -\frac{q}{q_{\text{MDN}}}, \tag{1}$$

where q_{MDN} is the median value of power consumption observed in the simulation environment, in our case $3,320$ Watt.

Next, we consider two components defining comfort: CO_2-levels and thermal comfortability. Considering comfort within the reward function is key to not solely optimizing for cost savings but also to bettering the overall working environment and consequently increasing productivity. However, there are situations where comfort can be disregarded, i.e. on weekends and before/after working hours. Therefore, we define the following indicator function:

$$T(h, d) = \begin{cases} 1 & \text{if } h \in \mathcal{H} \wedge d \in \mathcal{D}, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where $\mathcal{H} = \{7, \ldots, 16\}$ is the set of working hours, i.e. from 7 AM to 5 PM (exclusive), and $\mathcal{D} = \{1, \ldots, 5\}$ the set of working days, i.e. from Monday to Friday.

Regarding comfort and its dependence on the current amount of CO_2 present (measured in ppm), we suggest the following: as a potential

$$C_{\text{CO}_2}(c) = \begin{cases} 1 & \text{if } 0 \leq c < 1,000, \\ 1 - \frac{c-1,000}{1,000} & \text{if } 1,000 \leq c < 2,000, \\ -1 & \text{otherwise}. \end{cases} \tag{3}$$

Values under 1000 are considered as pleasent, while values between 1000 and 2000 mean stuffy air. An exposure of over 2000 pm can lead to health risks.

Lastly, we employ the definition of ambient comfortability as proposed by [1] and utilized in the subsequent works by [18], for which we introduce the following notation:

$$C_{\text{AMB}}(\text{temp}, \text{hum}) \in \mathbb{B} \tag{4}$$

This indicator function, denoting a person's comfortability depending on ambient factors such as temperature and humidity,

returns 1 if these values are within a predefined region. An illustration of this region is depicted in the works by [18].

Finally, the total reward is a combination of these introduced components, i.e. power efficiency and comfort dependent on CO_2 and ambiance:

$$r_t^i = P(q) + T(h,d)\Big(C_{\text{CO}_2}(c) + C_{\text{AMB}}(\text{temp},\text{hum})\Big) \quad (5)$$

By incorporating temperature, humidity, and CO_2-levels into the reward, the RL agent is encouraged to save energy while maintaining indoor air quality and temperature/humidity comfort. Note that Equation 5 denotes the reward at timestep t for a single zone/room i. An extension to multiple zones/rooms follows naturally.

IV. Reinforcement Learning for HVAC System

RL is a machine learning paradigm where an agent learns to make decisions by performing actions in an environment while optimizing a manually designed reward function. In the context of HVAC systems, the agent's objective is to optimize the performance of the system, considering factors such as energy efficiency, air quality, and thermal comfort. At each time step t, the agent observes a state s_t from the state space $\mathcal{S}$, takes an action a_t from the action space $\mathcal{A}$. After performing action a_t, the environment transitions to a new state s_{t+1} for which it receives a reward r_t. Ideally, the agent should learn a policy π that maximizes the overall cumulative reward.

A. State Preprocessing

In our system, the state space incorporates a cyclical time encoding to effectively capture temporal patterns, alongside z-normalization for other state variables. The cyclical nature of time is encoded using sine and cosine transformations, which represent the periodicity of hours within a day. This encoding is mathematically represented as follows:

$$\sin_h = \sin\left(2\pi\frac{h}{24}\right), \quad (6)$$

where h, represents the current hour of the day. Naturally, this expands to the encoding of weeks and years to capture weekly and yearly seasonal trends.

B. Restricting the Action Space

Our model employs an Actor-Critic architecture, implemented in PyTorch, for managing environmental parameters. The network includes a joint base for the actors and the critic, with ReLU as an activation function. A shared embedding space for each actor enables learning about the interconnected impacts of the environmental zones and actions. The model gains significant computational efficiency and a more cohesive understanding of the environment by employing shared embeddings. The environmental parameters are modeled using one actor head for each action, where each head is responsible for generating parameters for the corresponding action distributions. Opposed to this is the critic, who is responsible for predicting the value of a state.

Our action space is modeled using multiple distributions: We learn to predict α and β parameters for Beta distributions that model temperature, humidity, and the fraction of recycled air, as well as probabilities $p_1, \ldots, p_k$ for a Categorical distribution, modeling the three control logics. We opted for a Beta distribution because its support is between 0 and 1.

To ensure compatibility with the TRNSYS18 simulation environment, we transform the action samples obtained from our Actor-Critic model into appropriate ranges. This transformation restricts the action space, allowing the model to generate only feasible and, most importantly, realistic environmental actions, allowing for real-world cost-efficient applications from day one. The following equations summarize the exact transformation:

$$T_{\text{deg}} = \Big(23 + 20 \cdot (s_T - \frac{1}{2})\Big)_{[18,28]} \quad (7)$$

$$H_{\text{rel}} = 100 \cdot s_H \quad (8)$$

$$R_{\text{air}} = \Big(0.5 + 2 \cdot (s_R - 0.5)\Big)_{[0,1]} \quad (9)$$

$$M_{\text{mode}} = 1 + s_M \quad (10)$$

where

- T_{deg} is the transformed temperature action,
- H_{rel} is the transformed humidity action,
- R_{air} is the transformed recycled air amount,
- M_{mode} represents the mode for schedule, temperature, or CO_2-based approach,
- s_T, s_H, s_R, s_M are the respective samples drawn from the learned distributions of temperature, humidity, recycled air, and mode.

Note that the temperature action is adjusted to lie within the 18°C to 28°C range, while the humidity actions are scaled and clamped to be between 30% and 80% relative humidity. The amount of recycled air is calculated to effectively use the distribution borders by first mapping the sample to a range of $[-\frac{1}{2}, \frac{3}{2}]$ and then clamping back to $[0, 1]$. This approach ensures that the outputs of the model can be cost-effectively applied directly to real-world scenarios and guides the learning process to choose actions to stabilize comfortability.

C. Reinforcement Learning Algorithms

The most commonly used RL algorithms for HVAC systems are A2C, and PPO, because they can deal with continuous action spaces [5]. Both can be used with Actor Critic methods that consists of two components. An actor who updates policy distributions and a critic who estimates value functions.

The core of A2C is capturing the advantage $A(s,a) = Q(s,a) - V(s)$ of taking action a in state s over the average action for that state. This formulation helps balance exploration and exploitation in decision-making processes [3].

A2C is implemented with a unique loss function that balances between policy loss and value loss. Specifically, we use a weighted loss where the policy loss is assigned a weight of 0.125 in the combined loss calculation due to the fact that we learn multiple actions within a single network.

PPO is stable and efficient, primarily due to its objective:

$$L^{CLIP}(\theta) = \mathbb{E}\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

This function clips the policy update to avoid large deviations, ensuring smooth policy evolution and effectively addressing the trade-off between exploration and exploitation [4]. Its advantage lies in its ability to ensure small updates to the policy, avoiding significant policy changes that could lead to a performance collapse. This characteristic makes PPO particularly suitable for our application where stability in environmental control is paramount. During a training session, we perform 250 policy updates within our PPO implementation.

For both A2C and PPO, we conduct training sessions after every week which is equivalent to 7×24 hours of simulation time. We chose this duration specifically to incorporate seasonal trends, especially focusing on weekends, a period when the building is typically unoccupied.

V. Experiments and Results

We compare our system against three baselines, that use Control Logic 1, 2, and 3. We call them: schedule, which operates on a fixed schedule from 7 AM to 5 PM, temperature, which adjusts conditions solely on the temperature to keep a certain temperature level, and CO_2 responds to changes in CO_2 concentration in the environment.

All share the same set points: a temperature of 23°C, a relative humidity level of 40%, and 50% recycled air. Each system's performance is evaluated under various conditions and compared to assess their efficiency and comfortability. Our experiments span seven years of simulation data. Figure 2 presents the cumulative sum of power consumption over the last year. It shows that both PPO and A2C outperform the schedule baseline, the best-performing naive baseline that also respects comfortability.

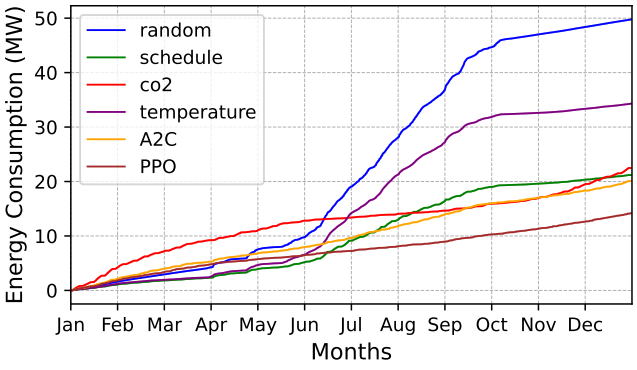

Fig. 2. Cumulative sum of power consumption over the last year.

The total performance gain of PPO over the schedule baseline is 12.32% or 18.1 MWh over 7 years. In the final year, PPO achieves a 33.03% or 7 MWh cost saving, albeit comfortability is slightly reduced. We hypothesize that over the years it learns to operate the HVAC system closer to the comfortability limits.

Figure 3 shows that power consumption is slightly higher for PPO and A2C within the first year. The quick adaptability of PPO and A2C means that these algorithms can efficiently adjust to the HVAC conditions from the very beginning of their deployment. No collection and pretraining on historical data is necessary, which means that the online implementation of PPO showcases its capacity for application in real-world scenarios.

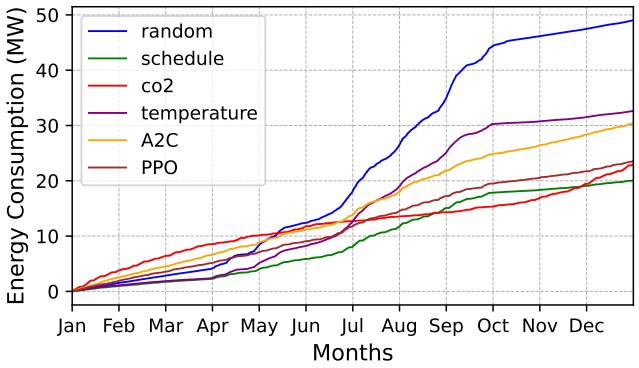

Fig. 3. Cumulative sum of power consumption over the first year. Even in the first year, A2C and PPO can adapt quickly in an online fashion.

Figure 4 shows the cumulative comfort hits, i.e. every time the previously defined comfort window is hit. Both A2C and PPO perform similarly to the schedule baseline while needing less power, with PPO slightly being more comfortable than the schedule baseline.

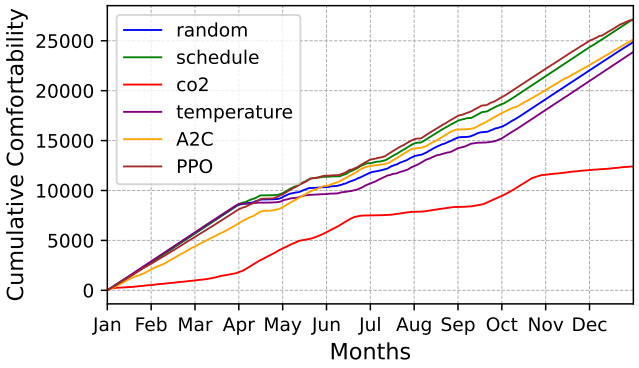

Fig. 4. Cumulative sum of comfort hits during the first year, illustrating the rapid optimization capabilities of A2C and PPO.

The collective insights from the data affirm the superior performance of A2C and PPO over static rule-based control logic, both in terms of energy efficiency and comfort level maintenance w.r.t. CO_2, humidity, and temperature, throughout the whole simulation. The superior performance of PPO over A2C in the context of energy efficiency and comfort level maintenance can be attributed to PPO's clipped objective function, which ensures stable and incremental learning by

preventing drastic policy updates. PPO's efficiency in handling multiple epochs of data per policy update also contributes to its effectiveness, particularly in environments where data collection is costly or limited.

VI. Conclusion and Future Work

Integrating RL into HVAC systems presents a significant potential for optimizing building environments and reducing greenhouse gas emissions. We showed that TRNSYS18, using our proposed socket communication, is a performant simulation environment that can realistically simulate HVAC system interactions for training and evaluating RL agents.

The PPO and A2C algorithms, with the former performing better, demonstrated quick adaptation in real-time HVAC control, effectively optimizing energy consumption while maintaining thermal comfort and acceptable air quality. This approach offers a viable alternative to conventional static methods, showing promising convergence in simulations. The presented approach therefore contributes towards sustainable and adaptive building management systems. For future work, we will focus on real-world deployment to validate our approach and fine-tune the model in a field test.

Acknowledgment

This project is funded by the Klima- und Energiefonds and carried out under the Energy Research Program 2020. We gratefully acknowledge the financial support provided to us by the Klima- und Energiefonds and FFG (Austrian Research Promotion Agency) for the KI4HVACs project within the programme "Energieforschung (e!MISSION)".

References

[1] R. Kosonen and F. Tan, "Assessment of productivity loss in air-conditioned buildings using PMV index," *Energy and buildings*, vol. 36, no. 10, pp. 987–993, 2004.

[2] J. Woods, N. James, E. Kozubal, E. Bonnema, K. Brief, L. Voeller, and J. Rivest, "Humidity's impact on greenhouse gas emissions from air conditioning," *Joule*, vol. 6, no. 4, pp. 726–741, 2022.

[3] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, pp. 1928–1937, PMLR, 2016.

[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[5] S. Zhou, A. Shah, P. Leung, X. Zhu, and Q. Liao, "A comprehensive review of the applications of machine learning for HVAC," *DeCarbon*, vol. 2, p. 100023, Sept. 2023.

[6] X. Fang, G. Gong, G. Li, L. Chun, P. Peng, W. Li, X. Shi, and X. Chen, "Deep reinforcement learning optimal control strategy for temperature setpoint real-time reset in multi-zone building HVAC system," *Applied Thermal Engineering*, vol. 212, p. 118552, 2022.

[7] D. Ruihua, C. Chenggang, and W. Yixuan, "Air conditioning system optimization in data center based on deep reinforcement learning," *Cryogenics & Superconductivity*, vol. 50, no. 09, 2022.

[8] Y. Du, F. Li, J. Munk, K. Kurte, O. Kotevska, K. Amasyali, and H. Zandi, "Multi-task deep reinforcement learning for intelligent multi-zone residential HVAC control," *Electric Power Systems Research*, vol. 192, p. 106959, 2021.

[9] Z. Zhang and K. P. Lam, "Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system," in *Proceedings of the 5th Conference on Systems for Built Environments*, pp. 148–157, 2018.

[10] M. Biemann, F. Scheller, X. Liu, and L. Huang, "Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control," *Applied Energy*, vol. 298, p. 117164, 2021.

[11] G. Gao, J. Li, and Y. Wen, "DeepComfort: Energy-efficient thermal comfort control in buildings via reinforcement learning," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8472–8484, 2020.

[12] X. Yuan, Y. Pan, J. Yang, W. Wang, and Z. Huang, "Study on the application of reinforcement learning in the operation optimization of HVAC system," in *Building Simulation*, vol. 14, pp. 75–87, Springer, 2021.

[13] J. Li, W. Zhang, G. Gao, Y. Wen, G. Jin, and G. Christopoulos, "Toward intelligent multizone thermal control with multiagent deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11150–11162, 2021.

[14] S. Taheri, P. Hosseini, and A. Razban, "Model predictive control of heating, ventilation, and air conditioning (hvac) systems: A state-of-the-art review," *Journal of Building Engineering*, vol. 60, p. 105067, 2022.

[15] W. Valladares, M. Galindo, J. Gutiérrez, W.-C. Wu, K.-K. Liao, J.-C. Liao, K.-C. Lu, and C.-C. Wang, "Energy optimization associated with thermal comfort and indoor air control via a deep reinforcement learning algorithm," *Building and Environment*, vol. 155, pp. 105–117, May 2019.

[16] L. Yu, Y. Sun, Z. Xu, C. Shen, D. Yue, T. Jiang, and X. Guan, "Multi-Agent Deep Reinforcement Learning for HVAC Control in Commercial Buildings," *IEEE Transactions on Smart Grid*, vol. 12, pp. 407–419, Jan. 2021.

[17] F. Omar, *Users Guide to Type277 Loosely-Coupled Integration of TRNSYS with Java*. US Department of Commerce, National Institute of Standards and Technology, 2019.

[18] S. Koçak and L. Pokorádi, "Comparison of defuzzification methods in the case of air conditioning systems," *Műszaki Tudományos Közlemények*, vol. 9, no. 1, pp. 115–118, 2018.