Application of Reinforcement Learning for Control of Heat Pump Systems

Dea Pujić*a, Marko Jelić*a, Marko Batića and Nikola Tomaševića
* School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia
a Institute Mihajlo Pupin,University of Belgrade, Volgina 15, 11060 Belgrade, Serbia
{dea.pujic, marko.jelic, marko.batic, nikola.tomasevic}@pupin.rs

***Abstract*— With the proliferation of heat pump systems for both heating and cooling applications for a wide range of space volumes, from isolated rooms to whole houses and buildings, their efficient operation is paramount to facilitate the transition to a more efficient building stock and reduction of greenhouse gas emissions. Also, phasing out polluting non-renewable fossil fuel-based heating systems in favor of heat pumps contributes notably to the electrification of the thermal domain and allows for a more notable share to be facilitated by clean and renewable generation in the future. Therefore, on top of modeling approaches for these types of systems, adequate control algorithms need to be developed and deployed to ensure the proper utilization of flexibility that these devices offer. This paper presents a set of techniques based on reinforcement learning for heat pump control of room temperature based on varying source and user loop flow rates as control inputs and discusses the implications of a selection of different control strategies on the observed indoor temperature variables.**

I. INTRODUCTION

With the goal of increasing the share of flexible loads in the total demand, heat pumps have been observed as key high efficiency appliances capable of bridging the gap between the thermal and electric domains. In doing so, they provide a means of increasing the adaptability of loads through the introduction of the previously untapped thermal side of the problem into the workflow of energy management solutions that traditionally focus only on electrical appliances. However, in order to ensure efficient operation of heat pump systems, control algorithms need to be deployed such that process parameters are kept in check and operated in the optimal way. With heat pump systems being, due to their intrinsic principle of operation by moving heat to or from the surrounding environment, highly reliant on stochastic meteorological parameters, ensuring that they are constantly operated at their highest possible efficiency is no easy task. Furthermore, programming controller algorithms for highly varying operating conditions presents an additional layer of complexity when striving for the best possible performance metrics.

Various approaches can be found in related literature depicting control algorithms for similar systems [1]. However, with previously mentioned high volatility in mind due to varying meteorological conditions, model-based approaches often need to be tuned independently in a variety of operating conditions as the process parameters change depending on the variable region that is being considered. In an attempt to mitigate this issue, this paper explores and presents a control approach based on reinforcement learning with two different operating rules, adapted to a specific use case involving a numeric heat pump model combined with a building model. Reinforcement learning has been chosen in particular due to its nature which allows for the algorithm to be self-learning, i.e., harnessing the ability to adapt to the system it is being applied to while being given minimum expert input. In essence, it explores the feasible region defined within the given environment and adapts in accordance with a specified simplistic reward/penalty depending on the achieved result(s).

II. RESEARCH QUESTIONS

The heat pump for which the control will be presented in this paper is based on a simplified air-to-water system and transfers a given amount of heat Q, which depends on two inlet flow rates. The first is the air flow rate on the external environment as the energy source side (expressed as a multiple of a nominal value, called the source flow multiplier sfm) and water flow on the consumer side (expressed as a multiple of a nominal value, called user flow multiplier ufm). The mentioned amount of heat Q is transferred to the building and accordingly, the temperature in a room, T_i, is indirectly controlled. Therefore, the aim of this approach is to design a controller that sets the flow rate(s) at the inlet(s) of the heat pump's heat exchangers (evaporator and condenser), which in turn controls the temperature of a particular room within the building. However, by analyzing the results of the model presented in [2], it was concluded that, although it is technically possible to control two flow rates, the impact of varying flow rates on the user side is relatively negligible, and hence it has been removed from consideration in terms of the development of a control algorithm for it. Therefore, it was decided to analyze the control of the internal temperature by adjusting only the air flow rate on the source side, with a constant flow of water set on the user side.

Although a suitable physical heat pump model was developed in order to test the performance and behavior of the controller, no parameter values that were used within this model were utilized in the design of the controller. From the standpoint of the control algorithm, the heat pump system was treated as a so-called black box for which it is only possible to measure the output values at given input parameters. In other words, the developed heat pump model was used as only a simulator, i.e., as an environment for testing its performance, with more details regarding its development available in [2] which discusses parametric heat pump modelling in greater detail.

III. METHODOLOGY

There is a large number of solutions offered in the related literature when it comes to the problem of designing control algorithms. Different solutions can be selected depending on the specific system properties, its key characteristics and the main goal that is desired to be achieved with a given controlled system. Within this paper, it was decided to utilize a common reinforcement learning approach, more precisely neural networks trained using reinforcement learning. Namely, the design of controllers for processes that exbibit frequent internal change or are extremely nonlinear, and consequently require different and specific control strategies, is often extremely challenging. However, as opposed to traditional fixed solutions, reinforcement learning-based controllers, which learn optimal control strategies by repeatedly observing the process through a number of experiments, are usually successful for such system groups.

Concretely, in this paper, the controller was constructed as a neural network with three hidden artificial layers (with 24 neurons each), and a deep Q-network (DQN) agent [3] based on the Q learning method that was used to adjust the parameters. In this context, the neural network is designed to map the current conditions, which include the current internal temperature, the outlet temperature of the heating system, the envelope temperature of the building and the outside ambient temperature, to the required control output. Since the network had as many outputs as there are possible discrete controls, the designed neural network was defined to have four inputs and n (number of control values) outputs.

When considering a conventional supervisory approach to the network training process as opposed to an unsupervised reinforcement learning control algorithm, the corresponding parameters are adjusted to minimize an estimation error on a predefined data set. On the other hand, in the case of a DQN agent, the parameters are determined in such a way as to maximize a certain reward function. There are examples that can be solved in both ways, but there are also some that cannot be solved by using supervised approaches. These include cases in which it is not known in advance what the desired output is. This was exactly the case with heat pump control as described previously. Namely, for a given set of input parameters, there is no historical labeled data that would suggest that it is correct or incorrect to apply a specific control. Accordingly, the supervisory approach is not suitable in this case and so, a DQN agent is used. In a specific example of heat pump control as will be discussed further, the reward for a certain condition could be defined in relation to the deviation of the current temperature in a room which is the object of the control system with a reference from a set point value with details in this regard given in the next section.

Namely, with respect to a desired user comfort specification, a range of acceptable temperatures can be defined bounded by a lower and upper value. Around these comfort constraints, an acceptable band of indoor temperatures is formed such that the control system can receive positive feedback (reward) if it maintains the room temperature within these limits, receive negative feedback (penalty) if the controlled temperature goes outside the bounds, or both. However, the formulation of the reward function (i.e., the way in which it is suggested to the algorithm if the outcome is positive, negative or indifferent) can have a notable impact on the achieved form of both the control signal as well as the controlled variable. Therefore, in the following sections, two different approaches will be discussed in terms of defining the reward function. A simplistic reward will be compared to an arguably more complex one and the output of the neural network controller will be presented.

IV. DISCUSSION

In this section, the obtained results for the heat pump control will be presented. As explained previously, a neural network trained using Q learning has been chosen as the control approach to map current indoor, outdoor and building envelope temperature T_e into one of n discrete control values so that the indoor temperature is maintained at a reference value of 21 °C. For the purpose of model development and corresponding testing, the external conditions were taken from the data set [4] for all tested cases using the first 100 samples of the sequence. The development has been carried out using Python libraries *gym*[1], *tensorflow*, *keras* and *rl*.

A. First use case – Simplistic reward function

Since the crucial step in the process of utilizing neural network-based reinforcement learning is the definition of the reward function, as it directly influences what type of behavior the network should encourage, two different strategies have been tested and will be compared. Since the considered problem was defined as maintaining the indoor temperature at a given constant set point of 21 °C, the idea was to specify such a reward as to have the model learn that the mentioned temperature is required to be maintained in a certain range. This range is defined as an interval $[20, 22]$ °C, as it is expected that a deviation of 1 °C on either side is acceptable and does not significantly affect comfort in the room.

Accordingly, the reward was defined as a pulse that had a positive value of 1 when the interior room was within the interval $[20, 22]$ °C, and -1 otherwise, i.e.

$$r = \begin{cases} 1, & 20 \le T_i \le 22 \\ -1, & \text{otherwise} \end{cases}.$$

Such a simple reward does not penalize in any substantial way large deviations, and also does not provide additional rewards for achieving a temperature of exactly or very close to 21 °C. When it comes to the potential controls that are available for this model, five potential values were selected, namely

$$u = sfm \in [0.00,\ 0.25,\ 0.05,\ 0.75,\ 1.00].$$

Accordingly, the observed network had four inputs and five outputs. In addition to setting the reward and available controls, the total number of steps during the training process was set to 5 000. Mean absolute error (MAE) and

[1] *gymz*. INM-6 & IAS-6, 2020. Accessed: Sep. 25, 2021. [Online]. Available: https://github.com/INM-6/python-gymz

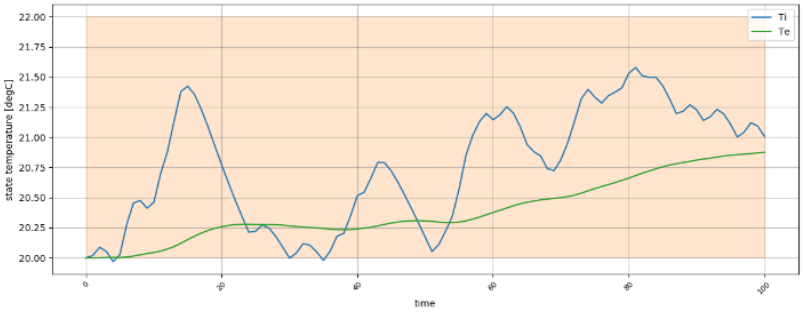

Figure 1 – Timeseries of the observed room and envelope temperature in the first case

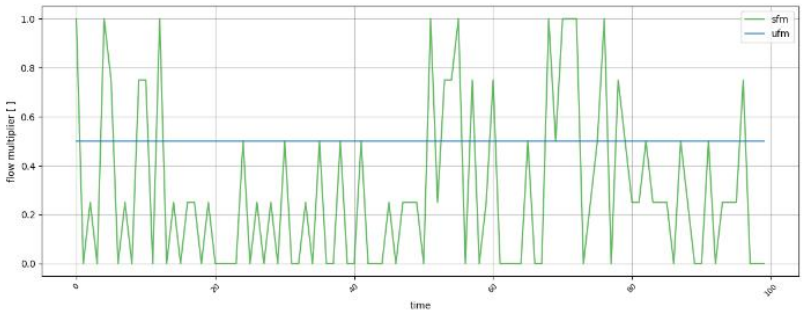

Figure 2 – Performances of the RL controller's control signal with the simpler reward function

the ADAM optimizer [5] with a learning rate of 0.01 were utilized as the objective function for neural network optimization. The same hyperparameters were also utilized for the training of the second neural network model.

The results achieved in terms of the internal temperature and the corresponding flow rate multiple, after the training process, are shown in Figures 1 and 2. The first of them shows the values of the controlled variable (internal room temperature), as well as the envelope temperature (secondary output of the model) of the building. As can be seen, the internal temperature in 98 out of 100 steps remains in the acceptable interval marked in orange $[20, 22]$ °C, which indicates a high quality of control. Namely, although the temperature is not in the immediate vicinity of 21 °C, such an outcome was not necessarily expected, having in mind the way in which the award was defined. Accordingly, the reward set in this way can be equated with the definition of the reward to maintain the temperature in the interval $[20, 22]$ °C, which is almost ideally achieved. The second figure shows the value of the predefined control ufm, as well as the value of the output of the controller – sfm from the source side. What can be immediately noticed is that all discrete control levels are present, which suggests that the model has explored and became aware of all the potential control options that were previously specified.

B. Second use case – Complex reward function

Another controller which will be presented in this paper, differs from the previous one primarily in the reward function that is defined in a slightly more complex way. Namely, the reward for the second reinforcement learning model has been defined as follows:

$$r = \begin{cases} 10, & 20.5 \leq T_i \leq 21.5 \\ 1, & 20 \leq T_i < 20.5 \text{ or } 21.5 < T_i \leq 22 \\ -|T_i - 21|/100, & \text{otherwise} \end{cases}$$

The goal of this reward was to keep the temperature closer to the set point, ideally in the range (21 ± 0.5)°C, but to also accept some minor deviations in the interval of (21 ± 1)°C. The results obtained after applying this control (letting the algorithm explore the allowed domain and training the control output) are shown in Figures 3 and 4. It could be observed that for the whole period of time the temperature is within $[20, 22)$ °C. However, the difference from this perspective is very small, considering that the previous controller also maintains the temperature in the given interval. Even in two samples when this was not the case, the deviations were minimal. However, significantly more notable difference was the time interval in which the temperature enters the range $[20.5, 21.5]$ °C. In the previous case, it was during only 59% of the time, while in this one, with a more complex reward function, it was 78%. This certainly shows an improvement in terms of the controller performance. When analyzing the control sequence, it can be noticed that it uses only three of the five possible outputs. However, since the value of 0.25 and 0 are treated as the same due to a saturation function which was applied, it can be said that only the control of 0.75 was not used. Taking into account all the above, as well as the fact that the duration of the training does not differ, it can be concluded that the second controller is preferable to the first one.

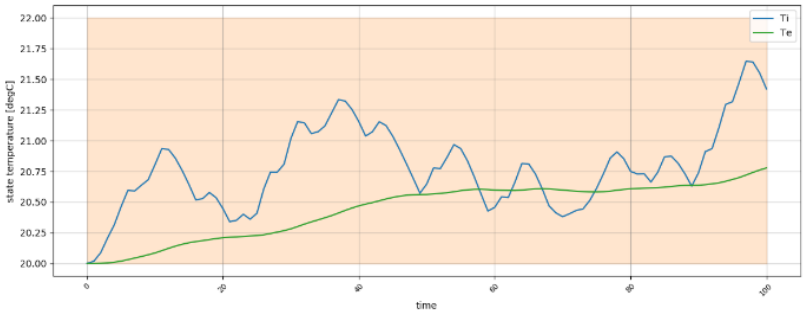

Figure 3 – Timeseries of the observed room and envelope temperature in the second case

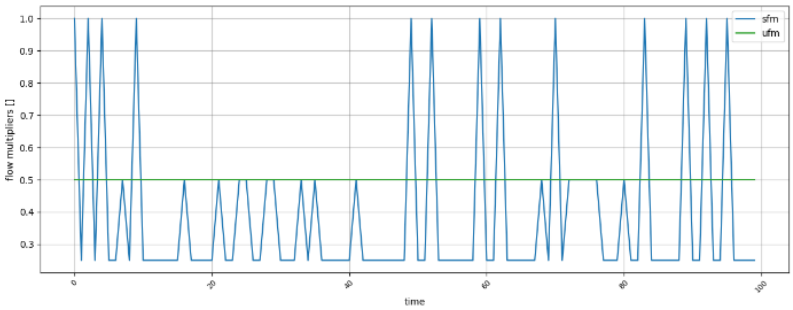

figure 4 – Performances of the RL controller's control signal with the more complex reward function

V. Conclusion and Future Work

The presented results in this paper depict how advanced reinforcement learning approaches for controllers can be applied to a combination of heat pump and building simulators to achieve a desired temperature range. This result is particularly interesting as it was achieved without the controller having any knowledge of the model structure that is being controlled. Thus, it shows that, without specific tuning to different conditions, such approaches may be considered to automatically adapt to varying circumstances.

Acknowledgment

The research presented in this paper is partly financed by the European Union (Horizon 2020 HESTIA project GA #957823 and SINERGY project GA #952140) and partly by the Ministry of Education, Science and Technological Development.

References

[1] S. Noye, R. Mulero Martinez, L. Carnieletto, M. de Carli, and A. Castelruiz Aguirre, "A review of advanced ground source heat pump control: Artificial intelligence for autonomous and adaptive control," *Renewable and Sustainable Energy Reviews*, vol. 153, p. 111685, Jan. 2022, doi: 10.1016/J.RSER.2021.111685.

[2] M. Jelić, D. Pujić, and M. Batić, "Simulation of heat pump performances in buildings," 2022.

[3] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature 2015 518:7540*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.

[4] C. Zagoni, "Data-Driven Thermal Models for Buildings," 2020. https://medium.com/analytics-vidhya/data-driven-thermal-models-for-buildings-15385f744fc5 (accessed Jan. 31, 2022).

[5] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Dec. 2014, doi: 10.48550/arxiv.1412.6980.