Simulation-based Control of Natural Ventilation with Operable Windows: Transformation from Predictive Control into Reinforcement Learning Control

Wei Zhang*, Ali Malkawi

* Harvard Center for Green Buildings and Cities, Harvard University, Cambridge, MA

* Corresponding email: wei_zhang@g.harvard.edu

Abstract

Natural ventilation is a promising passive technology to improve building energy performance and indoor air quality. However, the control of natural ventilation is a challenge in building technology and is often missing in building system advanced control design. This paper proposes the application of Model Predictive Control (MPC) and Reinforcement Learning (RL) control for winter natural ventilation control and evaluates both through on-site control experiments. Furthermore, this paper suggests the internal connection between MPC and RL control as simulation-based control, by transforming the MPC design into RL control design. This paper also highlights the RL control as a potential solver-free solution in deployment for building systems.

Introduction

Natural ventilation is a promising sustainable technology that improves indoor air quality and reduces energy consumption. However, the control of winter natural ventilation is still a challenge for engineers and architects: the cold draughts and regulation of CO₂ level (Mumovic et al. 2009; Vassella et al. 2021). With modern control strategies design, the motorized operable window could potentially resolve these issues with manually operable windows.

The model predictive control (MPC) and reinforcement learning (RL) control are two optional methods for winter natural ventilation control in this study. As a dominant advanced control strategy in process control industries for decades (Borrelli, Bemporad, and Morari 2017), MPC is becoming the research interest for building control in recent studies. However, it is rarely applied in natural ventilation control in real buildings (Drgoňa et al. 2020). RL is also closely related to control theory and engineering; the research has been conducted to apply RL in building energy efficiency in Google (Gamble; and Gao 2018) and other simulation-based research in mechanical ventilation and buildings systems (Azuatalam et al. 2020). At the Harvard Center for Green Building and Cities (CGBC), the researchers focused on the application of advanced control strategy in natural ventilation in recent years: Chen et al. (Chen et al. 2018; Chen et al. 2020; Chen et al. 2019; Chen 2019) investigated MPC, heuristic control, and the application of Reinforcement Learning and Transfer learning in EnergyPlus simulation for building energy studies. They found that MPC and reinforcement learning control have performance superior to heuristic control through building simulation. Their research was conducted through simulation-based study and needed further development work in real-world conditions. On the other hand, in order to investigate the natural ventilation control in real buildings, a data-driven model was proposed for natural ventilation control purposes (Zhang et al. 2020; Zhang, Wu, and Malkawi 2020). In this experimental research with machine learning algorithms, a control-oriented system dynamics of natural ventilation were obtained and serves as the starting point for this paper.

This paper firstly studies an MPC model based on the identified system dynamics of room air temperature and CO₂ level (Zhang et al. 2020). Secondly, an RL control model is built as a transformation of the MPC model with statistical simulation. Furthermore, both MPC and RL control models were evaluated with a self-developed Python API in a real building.

Reinforcement Learning overview

Reinforcement learning (RL) is a branch of machine learning where the interaction between a learning agent and its environment is defined in terms of states, actions, and rewards in the formal framework of the Markov Decision Process (MDP) (Barto 2018), as indicated in Figure 1. The learning agent takes an action in its environment and receives its performance indicator as a reward from its environment. In the training phase, the
learning agent adjusts its action based on quantified reward; in the application phase, the learning agent takes agent in order to gain maximal reward.

![Diagaram](image1)

Figure 1: learning agent and its environment interaction

The PEAS (Performance, Environment, Actuators, Sensors) (Norvig and Russell 2009) method is used to model the learning agent in the context of this study. The motorized operable window is modeled as a learning agent. It performs through the opening or closing of the window as actions; it receives the feedback information as rewards through the sensors in its environment.

<table>
<thead>
<tr>
<th>Table 1: PEAS description for operable window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent Type</td>
</tr>
<tr>
<td>Motorized operable window</td>
</tr>
</tbody>
</table>

Experiment

Building and Infrastructure

The naturally ventilated building in this study is in Cambridge, Massachusetts, USA. The south-facing office on its 3rd floor is set up for advanced control experiments, with motorized windows. This office has a dimension of 3.85 m in depth and 2.6 m in width. The volume is about 30 m³. The sensors deployed for this study include temperature, CO₂, wind speed, and direction sensors, as in Table 2; the actuators are motorized single-sided operable windows with two openings. A specified Python API is developed to augment the existing building control system and provide a programmable interface to operable windows for MPC and RL control experiments. Occupants were present in the lab around the reading desk as shown in

![Figure 2](image2)

Figure 2: building section and the south office as the lab

![Figure 3](image3)

Figure 3: the representation of the lab

<table>
<thead>
<tr>
<th>Table 2: Sensor network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor</td>
</tr>
<tr>
<td>Outdoor temperature sensor</td>
</tr>
<tr>
<td>Wind sensors</td>
</tr>
<tr>
<td>Room air sensors type 1</td>
</tr>
<tr>
<td>Room air sensor type 2</td>
</tr>
</tbody>
</table>

Winter Natural Ventilation Requirement

The winter natural ventilation requirement in this study is simplified from ASHRAE standards (ASHRAE
control design. Given the available information from the built environment through the sensor network, the first step is to find the relationship between room air temperature/CO$_2$ as dependent variables and the independent variables related to the built environment information (wind, outdoor temperature). With linearization assumption, the system dynamics for room air temperature and CO$_2$ level are identified through a data-driven method as follows.

$$
T_s(k + 1) = a_1 \cdot T_s(k) + f_1 \cdot U(k) \quad (1)
$$
$$
C_{CO_2}(k + 1) = a_2 \cdot C_{CO_2}(k) + f_2 \cdot U(k) \quad (2)
$$

$$
f_1 = \left( b_{11} \cdot \Delta T + b_{12} \cdot \sqrt{\Delta T} + b_{13} \cdot v \cdot \cos \theta + b_{14} \cdot v \cdot \sin \theta + b_{15} \cdot v \cdot |\cos \theta| + b_{16} \cdot v \right) \quad (3)
$$

$$
f_2 = \left( b_{21} \cdot \Delta T + b_{22} \cdot \sqrt{\Delta T} + b_{23} \cdot v \cdot \cos \theta + b_{24} \cdot v \cdot \sin \theta + b_{25} \cdot v \cdot |\cos \theta| + b_{26} \cdot v \right) \quad (4)
$$

Where $a_i, b_i$ are the coefficients that are identified in system identification. $T_s(k)$ is the room air temperature at $k^{th}$ moment ($^\circ$C). $C_{CO_2}(k)$ is the room CO$_2$ level (ppm). $U(k)$ is the operable window opening (0-100%) at the $k^{th}$ moment. $\Delta T$ is the temperature difference between indoor and outdoor temperature ($^\circ$C). $\theta$ is the wind incident angle. $v$ is the wind speed (m/s). The values of these coefficients are in Table 3.

The system dynamics are obtained through an iterative experimental method with a combination of data science and control theory: step 1) the natural ventilation features are proposed and evaluated in regression analysis, and step 2) the system dynamics are identified with selected features with system identification algorithm. The method of the first step was proposed initially with manually operable windows (Zhang et al. 2020) and further applied in motorized operable windows (Zhang et al. 2020). The method of the second step was proposed for the same motorized operable window in this study with the identified system dynamics as result (Zhang et al. 2020). The data in all these studies was collected through the same sensor network as detailed in Table 2; the collected data included wind speed, wind direction, outdoor temperature, and room air temperature/CO$_2$ level. During the data collection, one or two occupants presented in the lab room. The solar radiation was also collected as reference information, but not used to identify the system dynamics. More information about the identified system dynamics could be found in the related paper (Zhang et al. 2020). A more theoretical discussion around the modeling could be found in the related dissertation (Zhang 2021), which will be publicly accessible in July 2022.

© 2022 ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE’s prior written permission.
MPC design and experiment

MPC optimizes the operation of the operable window based on room air temperature and CO₂ level. Its cost function:

\[
\min c_1 \sum u(k)^2 + c_2 \sum (u(k) - u(k-1))^2
\]

where \( c_1 \) are the weight parameters

With the identified system dynamics in equations 1-4, the necessary soft constraints are added to form the complete predictive control framework.

<table>
<thead>
<tr>
<th>Temperature ( T ) dynamic coefficients value</th>
<th>CO₂ ( CO_2 ) dynamic coefficients value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1 ) 1.004</td>
<td>( a_2 ) 1.002</td>
</tr>
<tr>
<td>( b_{11} ) 0</td>
<td>( b_{24} ) 10.09</td>
</tr>
<tr>
<td>( b_{12} ) -0.191</td>
<td>( b_{22} ) -55.33</td>
</tr>
<tr>
<td>( b_{13} ) 0</td>
<td>( b_{23} ) -2.558</td>
</tr>
<tr>
<td>( b_{14} ) 0.1609</td>
<td>( b_{24} ) 0</td>
</tr>
<tr>
<td>( b_{15} ) 0.119</td>
<td>( b_{25} ) 0</td>
</tr>
<tr>
<td>( b_{16} ) -0.1227</td>
<td>( b_{26} ) -0.3529</td>
</tr>
</tbody>
</table>

Where equation (6) defines the higher limit of room CO₂ level. Equation (7) defines the room air temperature range. Equation (8) limits the change of room air temperature during a time interval. Finally, the equation defines the window opening range.

\[
C_{CO_2} \leq 1000 \text{ ppm} \quad (6)
\]

\[
20^\circ C \leq T_2 \leq 24^\circ C \quad (7)
\]

\[
|T_2(k+1) - T_2(k)| \leq 1^\circ C \quad (8)
\]

\[
0 \leq U(k) \leq 100\% \quad (9)
\]

The cost function (5), system dynamics (1 - 4), and constraints (6 - 9) form together with the MPC formulation. The implementation of MPC formulation is realized through Python CVXPY (Diamond and Boyd 2016) with a mathematical solver GUROBI (Gurobi Optimization 2021) for the deployment in the lab. Additionally, the MPC formulation is validated under MATLAB with a CPLEX solver.

RL control design and experiment

In the RL framework described in Figure 1 and the operable window learning agent (hereafter: agent) described in Table 1, the agent’s actions include the window opening and closing. When the window is closed, the natural and indoor environments have neglected mass transfer without natural ventilation. Once the window opens, the impact on the indoor environment from the natural environment through natural ventilation is evaluated quantitatively by a reward function. The control of this impact with reward value is the primary mechanism of RL control. The modeling of the environment and the design of reward function are the core of RL control design. Inspired by the MPC design in the previous section, the system dynamics in equations (1 - 4) could be used as a model of the environment, that combines room environment and natural environment. The performance measure of the agent in Table 1 could be realized through the cost function of MPC in equation (5) and the constraints of MPC in equations (6 - 9). The reward function is therefore built on the performance measure:

\[
R = R_1 + R_2 + R_3 \quad (10)
\]

\[
R_1 = \begin{cases} 
-d_1 \cdot |T_2(k+1) - T_2(k)|, & \text{if } |T_2(k+1) - T_2(k)| < 1^\circ C \\
-d_1 \cdot |T_2(k+1) - T_2(k)|, & \text{if } |T_2(k+1) - T_2(k)| \geq 1^\circ C 
\end{cases} \quad (11)
\]

\[
R_2 = d_4 \cdot |C_{CO_2}(k) - C_{CO_2}(k-1)| \quad (12)
\]

\[
R_3 = -d_4 \cdot U(k) \quad (13)
\]

Where the total reward R contains three rewards: 1) in equation (11), the \( R_1 \) is the temperature change reward, that is transformed from equation (8); 2) in equation (12), \( R_2 \) is the CO₂ level change reward, and 3) in equation (13), \( R_3 \) is the window opening reward. The hyperparameters \( d_i \) are tuned during the experiment to ensure the agent maximizes the CO₂ reduction while minimizing the room temperature change and window opening \( U(k) \). The values of coefficients \( d_i \) are available in Table 4.

<table>
<thead>
<tr>
<th>Reward coefficients value</th>
<th>Reward coefficients value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d_1 ) 100</td>
<td>( d_4 ) 20</td>
</tr>
<tr>
<td>( d_2 ) 500</td>
<td>( d_4 ) 700</td>
</tr>
</tbody>
</table>

The reward function group (10 - 13) and the environment equations (1-4) provide the foundation of RL control in this study. The following sections propose two different RL control strategies: Q learning control and Deep Q learning control.

Q learning RL framework

As one of the RL algorithms, the Q learning algorithm (Watkins 1989) often utilizes a discretized lookup table for Q value, based on the Bellman equation:
Equation (14) reveals that the selection of action $a_i$ in each state $s_i$ should ensure the gain of the max long-term reward $Q$ value. At the computational level, the $Q$ value is calculated in the following iterative equation (Jang et al. 2019):

$$Q(s_i, a_i) = Q(s_i, a_i) + \alpha [R_i + \gamma \max Q(s_{i+1}, a_{i+1}) - Q(s_i, a_i)] \tag{15}$$

The $\alpha$ is the learning rate, and $\gamma$ is the discount rate. In this study, the room CO$_2$ level for each state $s_i$ is discretized for the agent from a high CO$_2$ state (1200 ppm) to a low CO$_2$ state (700 ppm): 11 room states for every 50 ppm. The operable window action space was also discretized in 11 states for each 10% opening from complete closed (0%) to complete opening (100%). After the discretization, the agent's task is as follows: 1) find a sequence of actions to maximize the total rewards from any state to a low CO$_2$ state in the MDP schema.

Deep Q learning RL framework

The main difference between the deep Q learning RL (Riedmiller 2013) control in this study and the Q learning RL control is that the $Q$ value table is replaced by a neural network in deep Q learning RL control. Both RL control strategies utilize the same environment equations and reward equations.

© 2022 ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE’s prior written permission.
Result analysis

The MPC and RL control were evaluated on-site in the real building. These control algorithms were implemented in a cloud-based computational server (Intel Xeon 2.40GHz with 4G memory). During the experiments, one or two occupants generally presented in the lab.

MPC experiment result

MPC was evaluated several days on-site. Two tests are selected to illustrate its performance: 1) On March 13, 2020, the MPC test lasted from 10 am to 5 pm, as shown in Figure 8. The outdoor temperature was around 10°C in the morning and increased to 15°C. The south wind speed was around 2–4 m/s. With the opening of an operable window, the room CO2 level was kept around 1000 ppm. The room air temperature was stable. 2) On November 18, 2020, the MPC test lasted from 2 pm to 5:30 pm, as shown in Figure 9. The outdoor temperature decreased below 0°C most of the time. The north wind lasted the whole afternoon. The window opening increased with the decreasing wind speed from 4 m/s to 2 m/s. The room CO2 was regulated at around 1000-1100 ppm, with a stable room air temperature.

![Figure 8: MPC test on Mar 13, 2020](image)
Q learning RL control experiment result
The Q-learning RL agent control algorithm was evaluated in the building on March 24, April 12, and April 26, 2021. The outdoor temperature these days was either around 10°C or below 10°C. The room CO₂ was regulated around 1000 - 1100 ppm on all three test days with a short period above this range. When the wind speed was lower, as shown in Figure 10, the window opened more frequently than in Figure 11 and Figure 12. The room air temperature was always stable during the control test.

Figure 9: MPC test on Nov 18, 2020
Figure 10: Q learning control test on Mar 24, 2021

Figure 11: Q learning control test on April 12, 2021

© 2022 ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE’s prior written permission.
Figure 12: Q learning control test on April 26, 2021

Figure 13: Deep Q learning control test on June 15, 2021
Deep Q learning RL control experiment result
The Deep Q learning RL control was tested in June 2021. As the outdoor temperature was close to the room air temperature, only the regulation of CO\textsubscript{2} should be observed. The CO\textsubscript{2} level was maintained at around 1300 ppm, higher than the designed 1000-1100 ppm.

Discussion

Natural ventilation control
The design of the natural ventilation control strategy is quite different from the design of conventional HVAC control. The loading calculation of HVAC design guarantees the achievability of reasonable setpoint parameters such as CO\textsubscript{2} level, temperature, and humidity while conditioning incoming air by heating or cooling. The consumption of high-quality energy ensures the desirable condition of the served space. In the design of natural ventilation control, without the help of conditioned air, the natural ventilation control strategy may not be capable to realize the setpoint temperature. For example, in winter, the natural ventilation control cannot increase room air temperature by itself. Instead, it only makes the decision to operate the windows for an optimal result in the given condition, and it weights the CO\textsubscript{2}-temperature trade-off in different scenarios.

Temperature variation
In the MPC and RL control experiments, the room air temperature variation was very limited, based on observation (Figure 8, Figure 9, Figure 10, Figure 11, Figure 12). The following factors contribute to this result.

The operable windows in this study open vertically from the window frame surface, with a maximum angle of 18\degree. The most incoming cold air was either redirected to the roof if the flowrate was large, or slowly released into the window area if the flowrate was small. Two sensors were placed on the windowsill and reading desk; they monitored the room air temperature for the occupant’s thermal comfort. Once entering the room, the cold air was heated directly by the internal thermal mass on the roof internal surface and wall internal surface. The temperature difference was therefore mitigated.

Furthermore, in MPC and RL control design phases, the room air temperature and its variation were configured with a higher priority compared to the room CO\textsubscript{2} level. The control strategy is to keep the room warm and to find an efficient moment to release indoor CO\textsubscript{2}. As a result, the room CO\textsubscript{2} level could raise higher than 1000 - 1100 ppm for some time. Additionally, the permissible exposure limit of CO\textsubscript{2} in the workplace is 5000 ppm by OSHA (Occupational Safety and Health Administration). The higher CO\textsubscript{2} level during a short period in this study will not be harmful to the occupant’s health.

Simulation-based optimization and control
This experimental control study in the operable window for winter natural ventilation reveals the position of simulation in advanced control in buildings.

In MPC, the expression “predictive control” suggests the forecast of the variables, and the expression “model” is supposed to simulate the process (Haber, Bars, and Schmitz 2011). In other words, if the identified system dynamics is trustable as the process model, the simulation with the parameters in a given moment will predict the coming future from that moment. The philosophy behind MPC is the determinism viewpoint, the prevailing methodology in building simulation. In this study, the MPC predicts the possible consequence on the indoor environment (air temperature/CO\textsubscript{2} level) relatively short due to opening an operable window through the simulation of identified system dynamics. Then the MPC selects the optimal window opening based on constraints through the operation of a mathematical solver. Mathematically, MPC transforms a control problem into a simulation-based optimization problem and solves it for the optimal control command.

In RL, the Q learning algorithm family is considered to be “model-free” RL algorithms (Barto 2018). Ideally, the Q learning algorithm needs to learn about winter natural ventilation in the real natural/indoor environment through the trial-and-error method. However, the natural ventilation and the operable window comprise a high stochastic scenario to learn. Instead of the trial-and-error method with the real environment, the environment equations are constructed, and the exploration-exploitation search is deployed in this study. The environment equations represent the natural and indoor environments for the natural ventilation problem. In exploration-exploitation search policy functions as the statistical simulation in this study of Q learning control: in 10\% of the total 5000 – 10000 iterations, the agent was assigned to sample the action space randomly to try the possible actions exhaustively; in the 90\% iterations, the agent would optimize the explored actions. With the statistical simulation, the agent can learn and act in real-time. Therefore, the Q learning control is a kind of simulation-based optimization control, and the simulation is based on a probabilistic viewpoint.
Figure 14: simulation-based control framework: MPC & RL control

The transformation between MPC and RL control
This paper demonstrates a possible transformation from the MPC framework to the RL control framework: 1) the process model in MPC and the environment equations in RL control share the same system dynamics; 2) the cost function and constraint conditions in MPC constitute the reward function in RL control. The successful transformation of control formulations strongly suggests that the two simulation-based controls have some transformable concepts. Theoretically, the MPC solves the control problem by simulating in an infinite time horizon; the RL control solves the control problem by simulating in a probability space in space and action. In this context, the deterministic simulation in time horizon and the statistical simulation in probability space are linked: the control problem in time could be solved through a dual control problem in (probability) space.

MPC and RL control performance
The MPC and Q learning RL control have a similar control performance in CO₂ regulation and avoiding cold draught. The computational times of MPC and Q learning RL control are comparable, as in Table 5. In deep Q learning RL, the computational time is much longer; consequently, the control decision was made with a considerable delay. Environmental factors such as wind may not remain as before. This study reveals that the RL control would perform as well as MPC in natural ventilation control if the training of the learning agent would be finished in the second level.

Table 5: simulation-based control performance

<table>
<thead>
<tr>
<th></th>
<th>MPC</th>
<th>RL control</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO₂ regulation</td>
<td>Performed as designed</td>
<td>Performed as designed (Q learning RL control);</td>
</tr>
<tr>
<td>Room air temperature variation</td>
<td>No dramatic temperature change</td>
<td>No dramatic temperature change (Q learning RL control);</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No observation (Deep Q RL control)</td>
</tr>
<tr>
<td>Computational time</td>
<td>Less than 2 secs</td>
<td>Less than 2 secs (Q learning RL control);</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Around 2 mins (deep RL control)</td>
</tr>
</tbody>
</table>

Limitation of Deep Q learning RL control
A neural network significantly increases the training time with a limited hardware resource, as indicated in Table 5. Several possible methods could be tried for future research: 1) separating neural network training from control decision-making, such as training a neural network with all environmental parameters in advance. The advantage of this method is that the neural network could be prepared and tuned for control, but this neural network model would be more complicated than that used in this study due to more input parameters. 2) setting a dedicated processor (GPU or CPU) for neural network training. The advantage of this method is that the original algorithm is kept but at the cost of powerful hardware.

Mathematical solver free control
The Q learning RL control is shown to be an alternative to MPC in natural ventilation control through the experimental study, which could be extended to other building systems with the same control design philosophy. The advantage of RL control in this study is that the mathematical solver is not required as in MPC; the optimal value of window opening is found through statistical simulation. During the MPC experiments, the performance of open-source mathematical solvers in natural ventilation control was not satisfactory. The requirement of a high-quality solver would be a financial burden for deploying the MPC algorithm for each operable window in buildings.

Conclusion
This experimental control study proposed and evaluated the application of two advanced control in winter natural ventilation initially: MPC and RL control. Both control strategies showed satisfactory performance in room CO₂ regulation and avoiding the cold draught in a real...
building. Furthermore, the MPC and RL control modeling in this study reveals their internal connection as two sorts of simulation-based control strategies. Finally, the Q learning RL control is discussed as a mathematical solver-free control strategy compared to the MPC in natural ventilation control. The limitation of this study also exists: the control design is customized for a specific building. In the future, more experiments will be conducted to generalize the natural ventilation control design for more regions with different climate conditions. The RL control would also be investigated with a powerful computational resource.

Acknowledgments
The authors would like to thank Professor Les Norford from MIT for the valuable discussion.

References
Chen, Yujiao. 2019. 'Natural Ventilation Control Strategies and Their Effectiveness in Different Climates.', Harvard University.
Occupational Safety and Health Administration. 'Permissible Exposure Limits – Annotated Tables', Accessed 04/10/2022. osha.gov/annotated-pels.


