A Review for Reinforcement Learning and AI Techniques in HVAC Automation System

Course No: M03-056
Credit: 3 PDH

Ahmed Paridie, P.E.

CED engineering.ca CANADA
CPD FOR THE PROFESSIONAL

Continuing Education and Development, Inc.

P: (877) 322-5800
info@cedengineering.ca

Table of Contents

Abstract

Artificial intelligence (AI) has been widely used in 20th century to find optimized solutions for real-time problems in different disciplines. Since buildings consume around 40% of direct energy consumption in United States based on United States Green Building Counsel (USGBC) reports, a review is done on the opportunities where AI, especially reinforcement learning, is utilized to reduce the energy consumption of the heating ventilation and air conditioning (HVAC) system used in building industry.

This discussion starts with a review of the common AI algorithms used in the control sequences of HVAC systems. Since most (not all) AI algorithms need information about the environment being studied, an additional review is done on the methods used to collect simulated information that represent the HVAC environment of new buildings and the methods used to obtain data for existing buildings. Next, the architectures of recent AI algorithms are further discussed, and the methodologies used to interface the AI algorithm with a building HVAC system model are explained for different case studies. Finally, real-time applications where AI is used as an assistive algorithm to enhance energy savings are reviewed and the gaps that prevent AI from being widely used as a stand-alone control system for HVAC systems are discussed.

Review Highlights

- Using artificial intelligence to enhance energy savings for heating, ventilation, and air conditioning systems.
- The evolution of energy modeling software used to model real-time buildings to perform energy analysis and enhance energy savings.

Specifications table

Subject area	Computer Science
More specific subject area	Artificial intelligence and energy-modeling
Name of the reviewed methodology	Utilizing artificial intelligence to enhance energy savings of heating, ventilation and air conditioning system
Keywords **Resource availability**	Artificial intelligence, Building industry, Energy modeling, HVAC N/A
Review question	How artificial intelligence algorithms architectures are modified to accommodate the operation on a heating, ventilation and air conditioning system and perform energy savings? How is a building environment developed and interfaced with an artificial intelligence algorithm?

Nomenclature

BAS	Building Automation System
AI	Artificial Intelligence
NN	Neural Networks
RL	Reinforcement Learning
HVAC	Heating, Ventilation, and Air Conditioning Automation
VAV	Variable air volume
RBC	Rule-based controllers
USGBC	United States Green Building Counsel
GUI	Graphical user interface

Introduction

HVAC consumes 40 to 60% of the total energy use of a building, especially commercial buildings (Wetter et al., 2021). To reduce building energy consumption of HVAC, different customized improvements have been done by engineers like energy modeling and improved building automation system (BAS) control sequences (For example, ASHRAE 36). However, a building is considered a cyber-physical system where human, and weather physical systems interact with the different cyber building systems like heating ventilation and air conditioning (HVAC), lighting and plumbing systems. The loads that human and weather impose on a building can't be fully predicted, it is challenging to find a generic control sequence that can achieve minimal energy consumption for different buildings that have different weather conditions and occupancy patterns.

With the rapid development of computational power in 20th century, artificial intelligence (AI) gained a recognizable reputation of numerous successes in different business and industrial fields (Yuan et al., 2021). For example, it is used in business analytics to predict stock prices and in automotive industry to improve engine efficiency, implement autonomous driving vehicles and reduce simulations computational power (Paridie et al., 2022). In this paper, a review is done on the research work where AI is utilized for HVAC BAS to reduce buildings energy consumption (Urieli al., 2013).

To begin, an overview is done on the commonly used AI algorithms used by researchers to embed AI with HVAC control systems. Since most (not all) of AI algorithms need information about the environment on which the AI algorithm learns to optimize and to make it feasible for the AI algorithm to be pretrained before being implemented on the real-time building for the HVAC system, review is done on the evolution of the design of buildings systems simulations models used as testbeds to analyze a buildings energy consumption and investigate the performance of different rule-based controllers (RBC) and AI controllers. The techniques used to establish communication and data exchange between a building model, or a real-time environment and the implemented AI controller are also explained. Afterwards, for multiple case studies, the architecture used to embed AI algorithms with a building energy model to implement an AI aided HVAC control sequences are discussed and the resulting energy savings for each algorithm are demonstrated. Finally, real time applications where AI is used as an assistive approach to reduce buildings carbon footprint are demonstrated and the reasons AI can't be independently used to automate an HVAC system without the aid of an RBC are discussed.

AI algorithms review

Artificial intelligence can be categorized into three major disciplines, supervised learning, unsupervised learning, and reinforcement learning (Tao et al., 2019).

Since reinforcement learning algorithms are mostly used as the main structure in the software architecture of an HVAC control system while supervised learning and unsupervised learning are used as assistive techniques, this review glances supervised learning and unsupervised learning and is more focused on defining the structure of reinforcement learning algorithms.

Supervised learning techniques are used to predict the desired outputs of a system when the data about the inputs and their outcomes of a given system are fully defined. Neural networks (Figure 0-1) are one of the commonly used supervised learning techniques that has the ability to build a function that predicts the outcomes of a system for a given set of inputs. For example, the data recordings of the HVAC BAS of an existing building about the environment states (temperature, humidity) and actions (supply air setpoint) are used to predict the actions for a given state (Zhang et al., 2019).

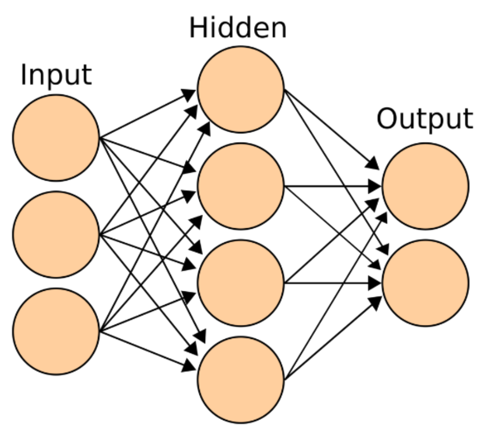

Figure 0-1: Graphical representation of a feedforward, single hidden layer NN

Unsupervised learning techniques are used when the data for a given system output is fully defined and only needs to be analyzed, most of the analysis is related to the categorization of the given data. Anomaly detection (Figure 0-2) is an example of the application of unsupervised learning in building industry. In anomaly detection, building outputs (Energy consumptions of HVAC, lightning, plumbing) are analyzed to detect the times where the consumptions don’t follow a regular pattern like local maximum and minimum consumptions. These anomalies are then used to inspect the dynamics of the system when this anomaly reading occurs and address the potential issues or leverage the potential opportunities. An example for an anomaly issue is increased energy consumption (Kurte et al., 2021). Analyzing building dynamics during this period would result in knowing the reasons for the increased consumption like malfunctioning meters or an equipment setpoint which has overridden during a maintenance procedure and wasn't released to be automated after maintenance is done. On the other hand, an anomaly leverage would be using the periods where energy consumption is minimal by introducing energy storages to store energy if the periods where consumption is minimal are sufficient to store enough energy and discharge it when energy consumption is high and reduce demand peak, demand ramping and utility bills.

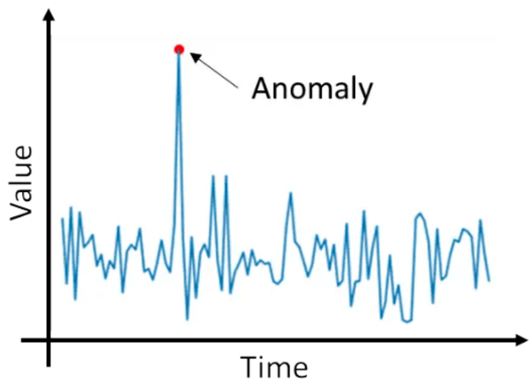

Figure 0-2: Graphical representation of anomaly detection done by unsupervised learning

Reinforcement learning (RL) techniques (Harmon et al., 1997) are used when there is an agent placed in an environment, where the set of inputs (All possible actions that can done by an agent in an environment) and the set outputs (All possible states of an environment) are known, but no mapping (control policy) exists between the inputs and outputs, and the agent is required to interact with an environment to build a control policy that determines which actions should be selected at given states that would result in desired outcomes (maximum reward and lowest cost) as shown in Figure 0-3.

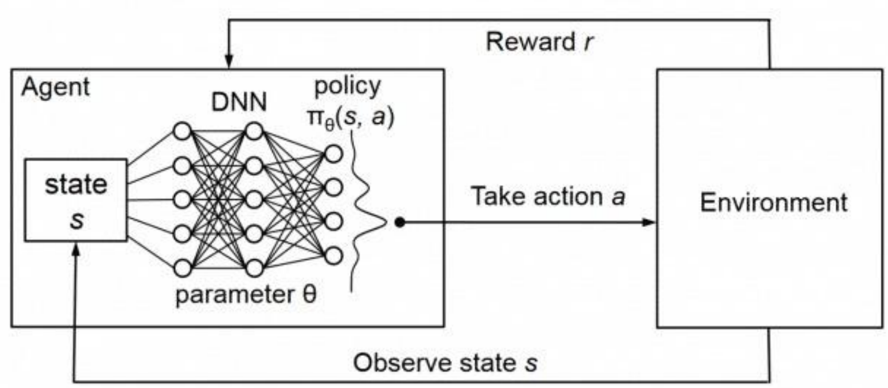

Figure 0-3: RL architecture

For example, an HVAC system would be an environment (Wei et al., 2017) whose states (temperature and humidity) are observed. and a controller (agent) would iteratively interact with the HVAC system (environment) and perform actions (altering equipment setpoints and states) to minimize energy consumption (cost) and maximize thermal comfort (reward). RL can be categorized into two main categories based on the environment availability (Azuatalam et al., 2020), the first category is model-free RL where a simulated environment model or even the data about the real-time environment are not available for the pre-training of the RL agent and the agent is required to interact with the environment by doing actions to know its dynamics, construct an environment model, and use the build-in-progress environment model to find optimal control policy. The second category is model-based RL where an environment model is available, and the RL agent is pretrained on the environment model to obtain an initial trained policy before the RL agent is introduced to the real-time environment. Under each one of the model-free and model-based RL categories, there are many algorithms where each algorithm may overlap and be used for both model-free and model-based RL categories. For example, Q-learning RL algorithm (discussed later in this section) is used for both model-free and model-based RL.

Since RL is a learning process that is based on iterative interactions of an agent with an environment, it is risky to directly use model-free RL where the iterative learning process of the RL agent is done on a real-time environment (a new building) because during these iterations, the RL agent may do actions that would lead to reaching states where a tolerable state is reached (increased energy consumption than the consumption obtained when using a programmable controller), even worse, it many reach forbidden states where catastrophic failure may occur (HVAC equipment failure or mishandling of an increased static pressure in the ducts of an HVAC system). Therefore, energy modeling is used to implement a simulation environment that represents the HVAC system of a new building or BAS data collected from a cloud are used to represent the dynamics of an HVAC system of an existing building (Chen et al., 2019). The simulated environment is then used by the RL controller to do model-based RL where the agent iterative learning process is done without causing any hazards to the equipment of the real-time building.

Although the model-based RL agents are pre-trained before being implemented on the real-time building, it is probable that the pre-trained the model-based RL agent would behave as a model-free RL agent (Xu et al., 2021) where a catastrophic failure may occur due to the following reasons:

- Uncertainty in the energy model of the new building on which the RL agent was pre-trained.
- Incompleteness of the BAS data obtained for an existing building on which the RL agent was pre-trained. For instance, an unencountered hot day with high temperature that wasn't covered in the collected data for an existing building, or an unexpected occupancy pattern where a building can be fully occupied on a holiday because employees decide to have an

indoor party in one year then the same building would be unoccupied in another year when employees decide to have a party outdoors.

To address the uncertainty issue in a building model for both new and existing buildings where a RL agent would encounter a model-free situation even if the agent was pretrained, active research for model-free RL is still being done to reduce the time needed for a pre-trained model to converge when implemented on the real-time building and to reduce the number of data need to be collected about a real-time environment for convergence to occur (Murugesan et al., 2020). One of the methods investigated is using model-free RL (Yu et al., 2020) while assuming that the existing data from an existing simulation environment for new buildings or from a BAS for existing buildings represent a real-time building to test the speed of convergence of the model-free RL agent in terms of finding an optimized control policy to investigate whether a model-free RL agent converges quickly and can be used directly on a new real-time building without the need for building data for pre-training.

A RL algorithm that is commonly used for both model-based and model-free RL is Q-learning (Watkins et al., 1992). Q-leaning is an iterative learning process where the RL agent keeps interacting with the environment to build a control policy. The control policy is represented by a map between the environment states and the agent actions that would result in a maximum value of the reward function at a given environment state (Xu et al., 2020). The mapping between the states and actions in a Q-leaning process is done using a Q-table for discrete variables or finite continuous variables (For example, a single thermostat with temperature as a continuous variable and an cooling equipment with change of state or on/off control), or, a Q-network (NN) that takes the current states as an input and results in the desirable actions as an output (For example, an integrated HVAC system with multiple setpoints for different equipment and thermal zones which have multiple thermostats) as shown in Figure 0-4.

Q-Table

	ACTION 1	ACTION 2	...	ACTION N
STATE 1	2.7	3.4	...	-1.3
STATE 2	6.7	-0.5	...	0.0
...	...	...	...	...
STATE N	-1.1	2.4	...	5.9

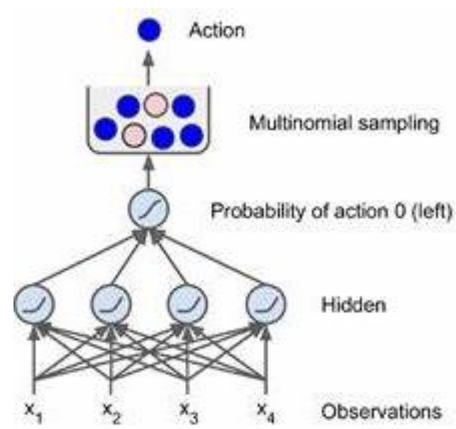

Figure 0-4: Q-table and Q-network architectures for a Q-leaning algorithm

Q-learning is one of the RL algorithms used to train a RL model. There are numerous algorithms different RL algorithms that are different in terms of the internal architecture, training methodology, and used data types (discrete or continuous and deterministic or stochastic) like A2C trust region policy optimization (TRPO), and proximal policy optimization (PPO). However, all of them have the same goal, to find a mapping or a policy that determines which action should be done at a given state to obtain maximal reward and minimal cost. This paper is also concerned about the architecture of how different RL algorithms (regardless of their type) are configured, manipulated, and connected to accommodate the requirements of an integrated HVAC control system in terms of inputs/outputs configuration, data analysis, algorithms training process, and decision making.

Since training a RL algorithm for integrated systems where the states and actions spaces are relatively large may become computationally expensive and take long time to converge even for model-based RL (like an integrated HVAC system for a building where there are multiple thermal zones with many thermostats and multiple HVAC equipment with many states and setpoints), transfer learning was proposed to save on computational power (Chen et al., 2018). Transfer learning is defined as taking the architecture of a pretrained AI model that is used to solve a predefined problem and use the existing trained architecture to initialize an untrained model that uses a similar architecture and solves a similar problem. For example, a NN that is trained to predict a unit heater state for an existing office building can be used to initialize the NN for a similar heater in a new office building.

Additional sub-models use statistical learning methods in HVAC industry to model or predict a portion of an HVAC system. For example, a Bayesian learner, which is one of the statistical learning methods, is used to predict an occupancy pattern for a building (Barrett et al., 2015). A Bayesian learner uses information obtained for an independent variable x and a dependent variable y where y is a function of x to build a probability distribution that represents the occurrence of y given the occurrence of a certain value of x.

Evolution of buildings energy modeling environments

Since most of AI algorithm require data about a building (supervised learning algorithms) or at least an environment to interact with to collect information about a given system and train to select actions that would lead to desirable result RL, efforts have been done by building industry engineers and researchers (Wetter, 2011) to design simulated environments and testbeds with the target to model a real time building energy consumption.

DOE2 and eQuest

In 1970s and 1980s, computer programs were made to simulate buildings energy use with the goal of reducing energy consumption. By the 90s, the US Department of Energy (DOE) developed a robust program, free to the public for this purpose and was called DOE2. Since

DOE2 required coding knowledge to be used, a graphical user interface (GUI) called eQuest (Figure 0-5) was developed to facilitate for the users to implement a building model.

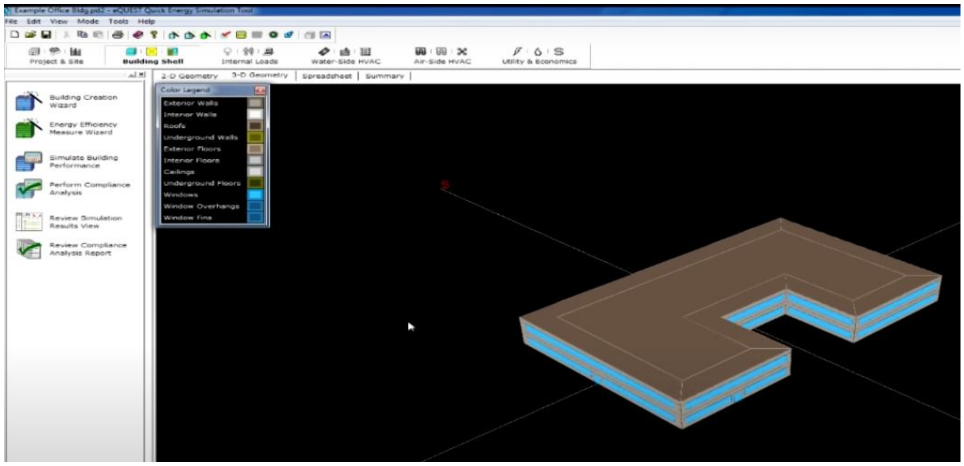

Figure 0-5: eQuest GUI – building geometry implementation stage

Energy Plus and Open Studio

In 1990s, the DOE (Crawley et al., 2001) began developing the next generation of energy simulation program called "Energy Plus ". It allows engineers and scientists in construction industry to predict and simulate how a building uses energy through its lifetime. Energy Plus uses a lot of mathematical models to calculate energy use for a building. In addition, just like DOE2, it is very obscure programming language-oriented program which is not very user friendly. By the late 2000s, DOE realized in order to get a wide spread of the program, a robust, easy to use GUI need to be developed. Therefore, a GUI called Open Studio (Figure 0-6) was developed for making inputs to Energy Plus.

The workflow flow of constructing a building energy model is similar for different energy modeling programs. The process is demonstrated for Open Studio as a case study. The process in Open Studio starts with making geometry using Floor Space JS, located within the Open Studio program. If the geometry is complex geometry, it can be plotted in SketchUp and imported into Open Studio using an Open Studio plug-in. Alternatively, the geometry can also be an imported geometry from IDF files GBXML files, SDD files, or IFC files. The 3D model plotted acts as a shell that holds all of the building energy modeling information.

After implemented 3D model, space types and thermal zones are assigned to the implemented 3D model. From there, the building model can be modified by changing different parameters such as:

- The number of people that are in the building
- Weather zones
- lighting power densities.
- ventilation rates.
- schedules for occupancy.
- Building operation schedules, like when a building is opened or closed.
- water usage
- HVAC systems set points.
- Hot water heating systems

After the model of the building is assembled, it is exported out to Energy Plus. Energy Plus then solves the equations that represent the mathematical model of the building implemented in Open Studio and delivers information about the building such as:

- Annual energy use.
- Building envelope performance.
- Peak space and HVAC loads.
- Peak water usage and ventilation.

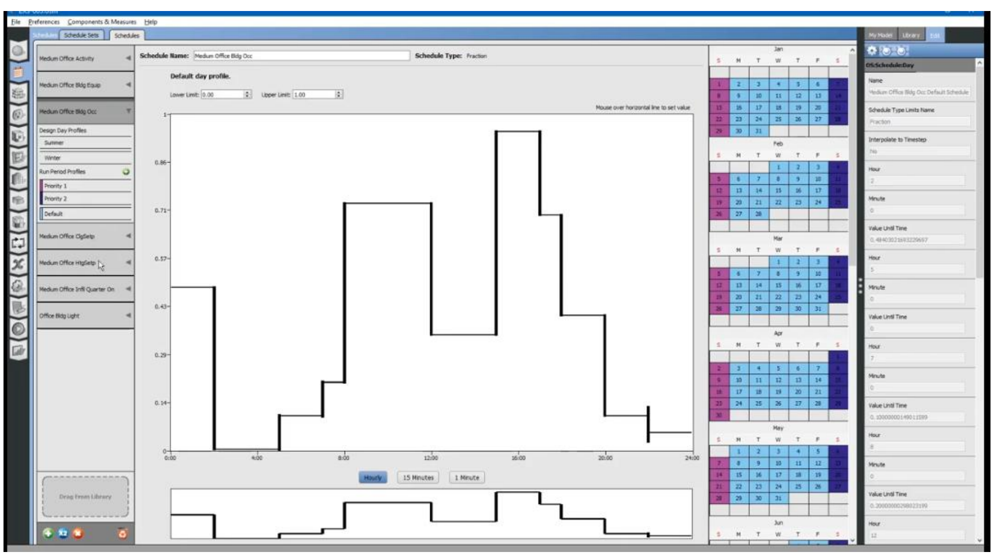

Figure 0-6: Open Studio GUI – occupancy schedule implementation stage

Open Modelica

Although Energy Plus took into consideration different aspects of building operation like annual HVAC system energy consumption and plumbing system usage, the building mathematical model was used for long-term evaluation and not for control decisions where short-term building dynamics can't be ignored. For example, local control scenarios were be idealized in Energy Plus (Wetter et al., 2014). If a building load decreases, the cooling capacity provided to the building decreases instantly without focusing on the dynamic response of the HVAC equipment like cooling coils and fans. Ignoring short-term dynamics when testing AI controllers may lead to inaccurate assessment of an AI controller performance.

To address this issue, opensource programs that model control systems dynamics were investigated with the target of implementing an HVAC control system model where short-term dynamics can be observed, analyzed, and controlled. Modelica, an open-source programming language, was used to implement mathematical modeling of physical systems and to solve the implemented equations to obtain a solution for the system response. In 1998, Linköping University developed Open Modelica which is an open source and free to the public GUI used to implement mathematical modeling of physical systems using Modelica programming language.

In 2014, Wetter et al. implemented a library (

Figure 0-7) that models the equipment of an HVAC control system like pumps, fans, cooling equipment and heating equipment. The advantage of modeling an HVAC system using Modelica is the ability to monitor HVAC system dynamics and energy consumption for small time steps like one second. This feature makes possible to test the performance of different control sequences since the control sequence can be implemented programmatically and the setpoints can be dynamic, which is not the case in Energy Plus as discussed earlier in this section where load-based models are used to take decisions for the implemented control sequence rather than the actuators commands and sensor signals (Wetter et al., 2015 and Booten et al., 2012). For example, in Energy Plus, if the load increases, the energy consumption for the cooling system would increase regardless of the equipment used to increase the cooling to the thermal zone. On the other hand, in Open Modelica, the temperature of the room would increase, and based on the control sequence algorithm, an equipment setpoint or state would be changed, for example, a fan speed would increase to push more air to the thermal zone, or the chilled water valve opening would increase to give additional cooling to the thermal zone which makes it more of a real-time HVAC control system modeling. The disadvantage though of using Modelica instead of energyplus is the longer simulation time since in Modelica solves, for small time steps, the system of equations modeling the building thermal model, the dynamic HVAC control sequence decisions, and the HVAC control system equipment consumption and states. On the other hand, in Energy Plus, the building thermal model and rule-based energy model are just computed, which makes it faster than Open Modelica.

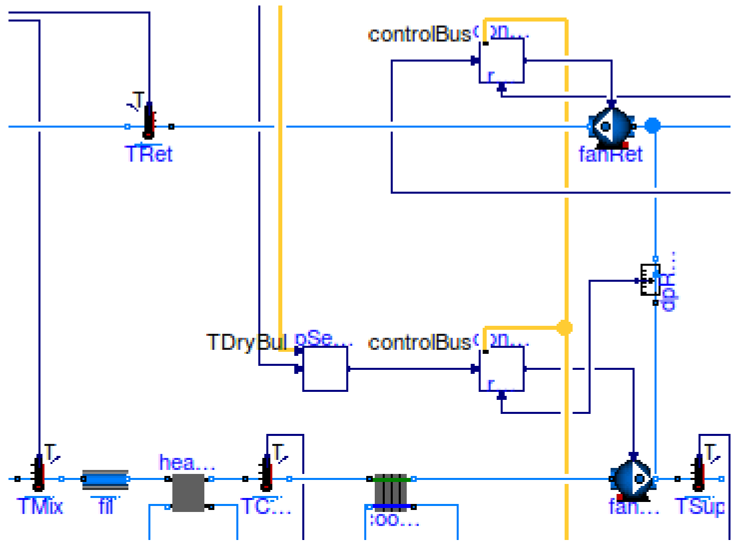

Figure 0-7: Partial view of an implementation of a variable air volume flow system in Open Modelica.

Energy Plus/Modelica co-simulation

To address the issue of the long simulation time when using Open Modelica to model a building thermal model, HVAC control system equipment and HVAC control sequence, a co-simulation model is designed where the HVAC equipment control system is modeled on Open Modelica and the building model is simulated on Energy Plus. The Functional mockup interface (FMU, Blochwitz et al., 2011) was used as a medium to implement a co-simulation schematic between Energy Plus and Open Modelica where simulation data between the two programs are shared. It is shown that when using the building thermal model from Energy Plus and using the HVAC system equipment and control sequence from Open Modelica, the simulation is almost 50% faster than using only Open Modelica to model the building thermal model, HVAC system controls equipment and control sequence (Wetter et al. 2014).

Python

In 2022, python has become the most trending open-source programming language in 2022 (Cass, 2022), therefore, different companies from different technological disciplines, including building industry, have been issuing libraries for the users to be able to interface with their programs and technologies using python. An Energy Plus python library is developed by US DOE to implement an Energy Plus building model in python, use Energy Plus simulation engine through python to compute the building model energy consumption, and leverage Energy Plus analysis tools to analyze the resulting consumption in python. In 2012, Asghar et al. developed OMpython to build and simulate Open Modelica models using python including HVAC control system equipment library. The basic implementation of python libraries for Energy Plus and Open Modelica is by converting a software model to an FMU, then the FMU is compiled into a python code.

In 2020, Lee at al. designed a pyhton based HVAC environment testbed called Pyfmi. To implement Pyfmi, Open Modelica and Energy Plus libraries are converted to a co-simulation FMU interface using PyModelica and EnergyPlustoFMU respectively where simulation data between the two programs are shared. Afterwards, Pyfmi python library is implemented based on the implemented co-simulation FMU interface. Pyfmi library (Figure 0-8) makes it possible to build a Modelica-energyplus building model using python and to use linear programming and/or open-source python libraries for AI and ML to experiment different dynamic HVAC control sequences on the implemented model.

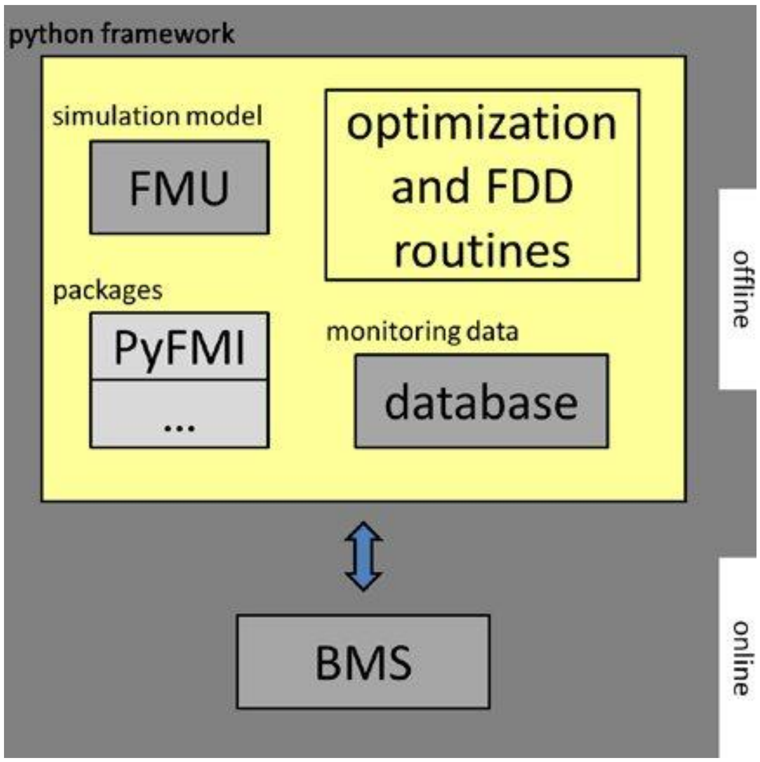

Figure 0-8: Pyfmi workflow

José et al., (2019a) implemented a python-based simulation environment for buildings to study the effect of different control sequences on the performance of the buildings energy storages. The implemented a simulation environment consists of 9 buildings where each building has its own cooling load, heating load, electric load and energy storage configuration. The average hourly load is used to model the buildings environment for 5 years. The developed environment though is dependent on the excel documented data of a building like building loads which makes it only valid for existing buildings where building data are available.

CITYSIM

In 2009, Robinson et al. developed CITYSIM software (Figure 0-9) with the target of not only doing an energy modeling to a building, but also to include the effect of the environmental surrounding on the building in the energy modeling process like:

- A building carbon footprint and transportation traffic time needed to reach the target building.
- Energy consumption study for irrigation water and the analysis of evapotranspiration effect of vegetated surfaces.
- The analysis of the water consumption of a building and the assessment of the quality of wastewater.
- Modeling of rainwater gathering and storage and simulation of the recycling of black water.
- Economic savings done when using domestic water from multiple sources like recycled water, harvested water from rain and major sources like city supply.

The target of CITYSIM is to model the energy consumption of an integrated city rather than studying each building separately. To verify the consistency of CITYSIM, Walter et al. (2015) compared the annual energy consumption of a simplified thermal model of a building to the results of CITYSIM for the same building and the maximum error was 5.1%.

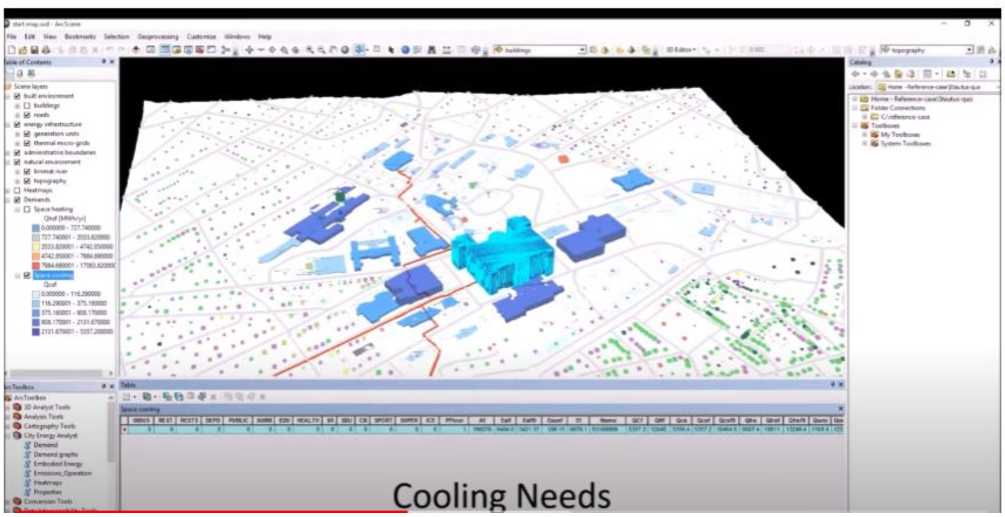

Figure 0-9: CITYSIM – cooling demand implementation stage

Miscellaneous programs

Different programs were also designed by different companies to model a building energy use or to compute the load requirements of a building. For example, Trane3D is a program similar to Energy Plus where a 3D model of the building is implemented, and the energy use of a building is analyzed. However, Trane3D is not free to public. Other programs also like HAP are used to compute a building thermal load based on the information given about a building material and HVAC equipment and has the ability to give suggestions for building design edits that would contribute to reducing a building thermal load like a building north direction orientation. The building north direction is important because it affects the solar load a building encounter. HAP GUI though takes only numerical inputs rather than having a 3D model that visualizes buildings like Open Studio and Energy Plus.

After a building thermal load analysis is done and the equipment that covers the thermal loads are selected. Programs like HVAC controls professional are used to design a control circuit HVAC control system needed to operate the selected HVAC equipment. In the next section, the different investigations done to experiment with different AI aided HVAC control sequences and building design algorithms on the discussed building environments in this section are demonstrated.

AI aided HVAC control sequences

Two main applications are reviewed, for each application, different architectures for the AI aided HVAC control sequences and the simulation environment with which the sequence interacts with are explained. The first application is utilizing RL for energy storages control sequences to reduce peak demand and demand ramping, and the second one is utilizing model-free and model-based RL to develop an adaptive dynamic HVAC control sequence that would result in a reduced energy consumption for the HVAC system of a building.

One of the advantages of an AI aided HVAC control system is that it has a NNs in its architecture. These NNs has the ability to leverage building dynamic features to enhance energy savings by constructing a prediction model for the dynamic features and using the predictions to find opportunities where the dynamic features can be used to reduce energy consumption.

For example, one of the dynamic features considered is the thermal capacitance of a building thermal zone. Thermal capacitance is an indication of the zone ability to store thermal energy, the higher the capacitance, the higher the zone ability to keep its temperature. A zone capacitance is variable over time, for example, at 2 pm an office may be occupied, and the capacitance would be the office components like furniture in addition to people, while at 8 pm, an office would be empty and only the office components capacitance is considered. If capacitance can be predicted using a NN, an AI controller would be able to leverage a thermal zone capacitance to store thermal energy inside the zone and reduce the external heating/cooling needed from an HVAC equipment.

AI aided Energy storage control sequence

Energy storage is used in building industry to reduce the peak demand, the demand ramping and the energy consumption of a building as shown in Figure 0-1. It does so by storing energy when the building demand is low or when the utility cost is low (for example, at night) then discharging the stored energy when the building demand is high. Energy storage type depends on the application it targets to reduce energy consumption. For example, chilled water or ice tanks are used to support cooling loads and reduce air conditioning cooling equipment demand, hot water tanks for heating loads to reduce gas or electric heating consumption, and Batteries/UPS systems are used to backup electric loads like lightning. In addition to saving energy, they act as backup resources in the case a building encounters a power outage.

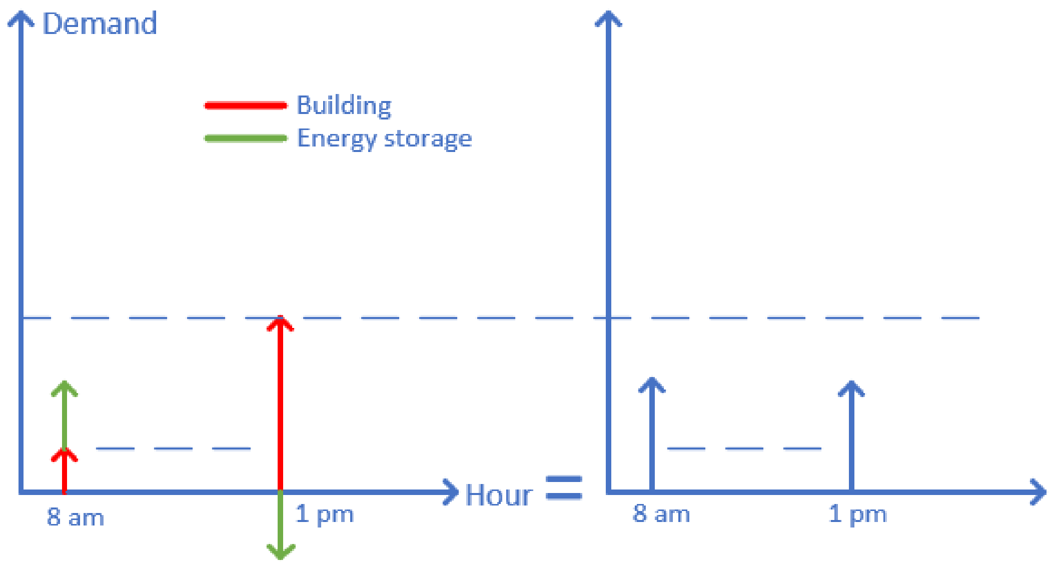

Figure 0-1: Demonstration of demand peak and ramping reduction done by an energy storage

The common approach for recent control sequences for energy storages which doesn't use AI is to charge the storage at night where electricity pricing is cheaper (Julian et al., 2022). As an improvement to only charging at night, a model-free RL learning (see section 1.1) is used to predict the instantaneous amount the energy storage is being charged or discharged along the day not only at night (José et al., 2017 and José et al., 2019c). In 2017, José et al. modeled an HVAC environment on CITYSIM (see section 0) for a heat pump and an energy storage tank. A traditional rule-based controller (RBC) would operate the heat pump when the tank temperature falls below a minimum temperature setpoint to increase the tank temperature to a maximum temperature setpoint. To experiment whether the RBC achieves optimal control that would result in minimum heat pump consumption (RL agent cost), Q-learning, which is a RL algorithm, is used as a controller to determine the reference temperature of the storage tank (RL action). Figure 0-2 shows that the RL aided energy storage controller reduces the utility cost of a building to value less than the ones obtained by a rule-based controller (RBC).

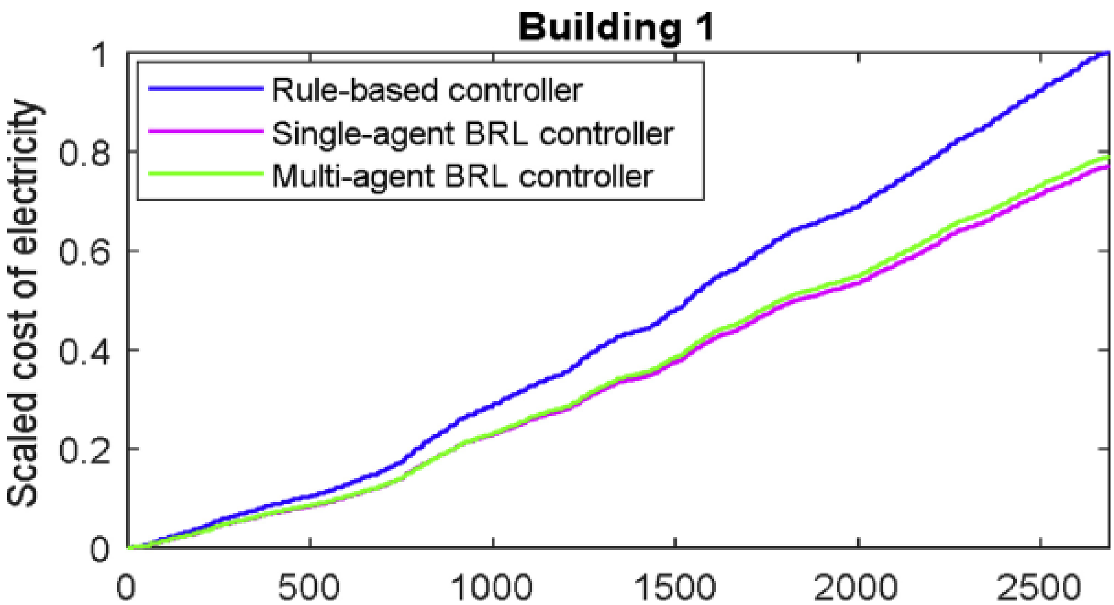

Figure 0-2: Utility cost when using an RBC vs RL controller (José et al., 2019c)

AI aided thermostats

For building systems like HVAC, a control sequence is needed to coordinate the HVAC system equipment operations to meet the thermal comfort requirements for occupants and the thermal demands of the building. Different control sequences are designed by different engineers with the goal of reducing the energy consumption of the HVAC system while meeting the thermal comfort and thermal demands requirements. What if an adaptive AI aided control system can be designed to alter the HVAC equipment setpoints and state dynamically to obtain a tailored control sequence that would result in minimal energy consumption regardless of the type of building the AI aided control system encounters? This AI sequence would then reduce the effort needed to design a control sequence for each building and would target the minimal energy consumption that would address the thermal comfort requirements for occupants and the thermal demands of the building.

In 2015, Barrett et al. developed a simple heat transfer model to implement a single zone which is being heated by an electric heater. The heat transfer model consists of two equations. One is

$$\dot{Q}_{environment} = UA(T_{outside} - T_{room})$$

where

$\dot{Q}$ is rate of heat transfer
U is over all heat transfer coefficient
A is area of walls, roof and floor
T is temperature
to represent heat transfer between outside air and room air, and the other equation is

$$(\dot{Q}_{heater} + \dot{Q}_{environment})time = \rho V C_p (T_{room\ at\ time\ t+1} - T_{room\ at\ time\ t})$$

where

ρ is average air density
V is room volume
C_p is air thermal capacity

to represent the change in the temperature of the room for a given time step. An occupancy schedule is also assigned to this zone and a daily periodicity (occupancy pattern is repeated daily) of the occupancy pattern is assumed. A gaussian noise is added to the occupancy pattern to represent the slight changes that occur to an occupancy pattern from day to day. The challenges in this model are to:

- Predict the time to occupancy.
- Decide the time needed to preheat the room before an occupancy occurs that would result in minimal energy consumption and achieving the room thermostat setpoint.
- Predict the overall pattern of the state of the heater (on/off) during building operation that would result in minimal energy consumption and achieving the room thermostat.

To demonstrate the effectiveness of using a model-based RL aided HVAC controller to maintain the zone thermostat setpoint, two leaning processes are included. The first stage is using a Bayesian learner (see section 1.1) to predict the occupancy pattern of a thermal zone. The predicted occupancy pattern is then used to predict the remaining time till occupancy occurs at a given moment. The predicted time to occupancy is then used by the RL controller as an environmental state that contributes to the decision of choosing an action for RL controller.

To construct the RL construct for this case study, three environment states are observed which are the room thermostat temperature, outside temperature and time to occupancy. Also, two actions are used which are the states of the heater (on/off). The reward and cost function for this RL model is constructed to be:

- A maximum reward is earned when thermostat setpoint is achieved while the thermal zone is occupied, and the heater is turned OFF.
- A minimum reward (maximum cost) is penalized when thermostat setpoint is NOT achieved while the thermal zone is occupied, and the heater is turned ON.

Since the number of states and actions is small (three states and two actions), Q Learning (RL agent) collects data about the environment temperature, outside temperature and time for occupancy (states), then predicts the state of the unit heater (actions), to minimize the energy consumption (penalty), and maximum the thermal comfort (reward). Figure 0-3 demonstrates the effectiveness of the RL controller and its ability to reduce the daily cost to value less than the ones obtained by the RBC. This heat transfer model has major drawbacks:

- The environment represents heating only without cooling.
- The model is simple and doesn't accurately simulate real-time building.
- The environment is for a single zone with a single HVAC equipment which doesn't represent an integrated HVAC system.
- Offline training is needed for the RL controller for it to be able to perform better than the RBC which means that for a real time thermal zone, a high accurate model is needed for the RL controller to show improved results.

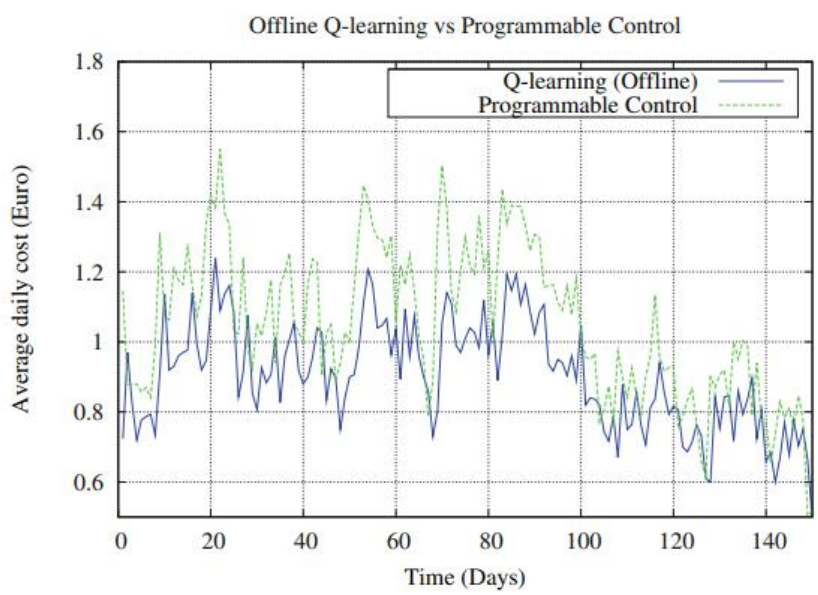

Figure 0-3: Daily cost when using an RBC vs RL controller (Barrett et al., 2015)

The reason this simple model is explained in detail is that the advanced building models and RL controllers in following subsections may look different, but the same general concept is followed. A RL controller is implemented on a simulated environment to learn a control policy that performs the actions leading to observations that would result in a maximal reward and minimal losses.

AI aided Air Handling Unit

In 2020, Debmalya developed a more advanced heat transfer model than the one implemented by Barrett et al. (2015) to model an air handling unit (AHU) that does cooling and heating to a single zone in a factory by controlling the chilled water and heating hot water coils flow rates respectively using control valves. Offline training is done for a model-based RL controller that uses Q- learning to predict the cold water and hot water control valves openings that would result in a reduced energy consumption. Based on results shown in Figure 0-4, the cooling and heating valves openings readings computed by the RL controller are less than the opening computed by the proportional, integral, differential (PID) controller, hence, the flowrate to meet the thermal comfort requirements for the industrial zone is smaller and the target of having energy savings is achieved. The same drawback though is still carried from Barrett et al. (2015) work which is the need for offline training for the RL controller for it to be able to perform better than the RBC.

	Room Temperature ValueY	Room Humidity ValueY	Heating Valve ValueY	Cooling Valve ValueY	Humidifier Valve ValueY	Reheat Valve ValueY
mean	22.027662	50.037363	9.372753	10.498550	30.770681	20.468397
std	0.066739	0.922815	20.184126	14.498391	29.895784	31.726365
min	21.743055	46.518517	0.839120	0.607639	0.000000	0.491898
25%	21.990740	49.479164	1.012731	0.839120	0.000000	0.665509
50%	22.019676	50.086803	1.215278	5.049190	29.300000	1.157407
75%	22.063078	50.665508	4.166667	17.100695	49.960000	34.143517

PID based HVAC control readings

	Room Temperature ValueY	Room Humidity ValueY	Heating Valve ValueY	Cooling Valve ValueY	Humidifier Valve ValueY	Reheat Valve ValueY
mean	22.009329	49.529951	8.446135	7.832123	25.714461	11.139734
std	0.152894	1.419767	12.799755	9.436951	23.573753	21.848293
min	21.151621	44.531250	0.839120	0.578704	0.000000	0.491898
25%	21.931133	48.726852	0.925926	0.781250	2.157418	0.578704
50%	22.019676	49.469906	1.591435	3.327546	24.203690	0.954861
75%	22.092014	50.173611	11.400463	12.413195	38.818600	9.143518

RL based HVAC control readings

Figure 0-4: HVAC readings when using an PID vs RL controller (Debmalya, 2020)

OCTOPUS

Xianzhong et al., 2019 developed a RL controller for BAS controllers called OCTOPUS. Not only it counts for reducing the HVAC system energy consumption, energy savings for lightning systems of a building are also counted for in OCTOPUS agent architecture where shading and lightning is implemented in the building environment. Xianzhong et al. (2019) also upgraded the building environment implemented by Zhang et al., (2018) to include the lightning and shading system in addition to the HVAC system.

Since this integrated environment would have many states and actions, an adjusted Q-learning architecture is used by OCTOPUS to predict the actions for both HVAC and lightning system. Instead of having a single Q-network that represent HVAC and lightning system, four local Q-networks are used as shown in Figure 0-5. Each local Q-network would observe the local data for the system it is assigned (HVAC, lightning, windows and blinds systems) to train the local network to perform local actions that would result in a maximal global reward (thermal comfort) and minimal global penalty (energy consumption).

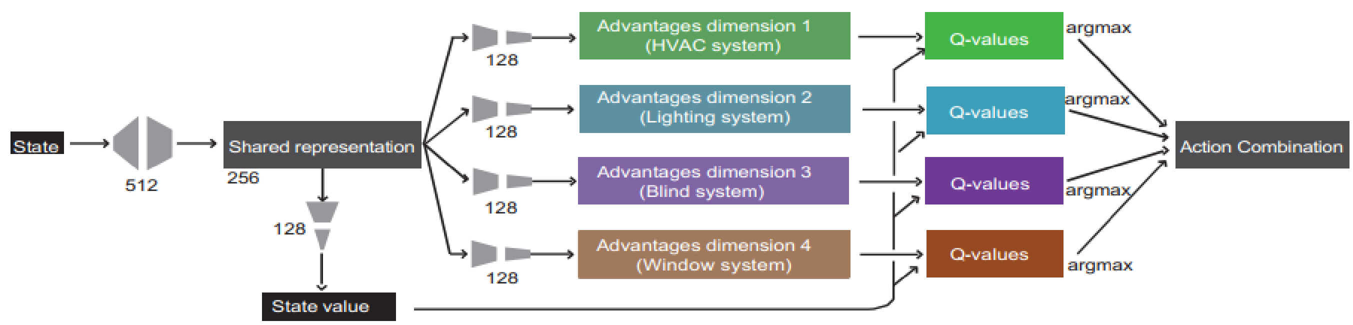

Figure 0-5: OCTOPUS architecture (Xianzhong et al., 2019)

In Figure 0-6, the resulting daily energy consumption when using OCTOPUS agent as a BAS controller is compared to the energy consumption obtained when using an RBC and the HVAC RL agent implemented by Zhiang et al. (2018) as controllers. It is noted that as the number of days increase, the performance of OCTOPUS becomes better.

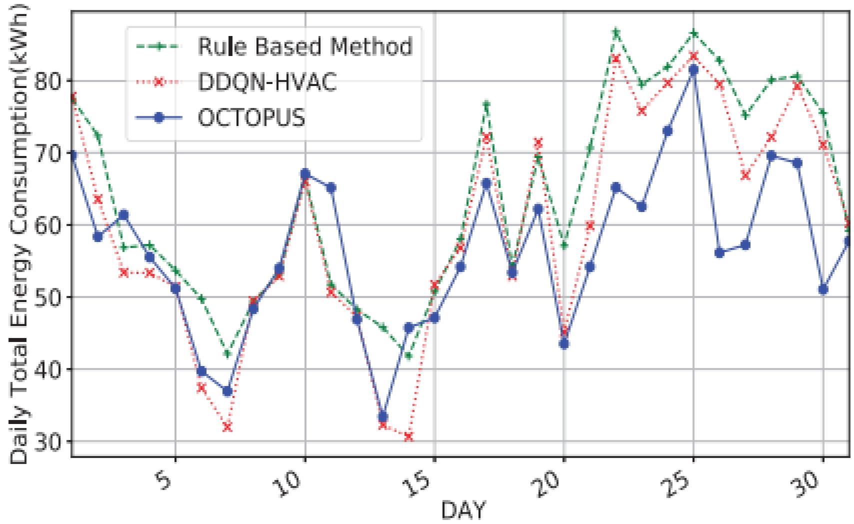

Figure 0-6: Daily energy consumption when using OCTOPUS vs RBC and HVAC RL controller (Zhiang et al., 2018)

MARCO

In 2020, Nagarathinam et al. developed MARCO, a model-free RL aided controller for HVAC systems where transfer learning is embedded within RL Q-network architecture. In MARCO, each HVAC equipment is assigned a local Q-network that optimizes the local energy consumption of the equipment which is similar to the work done by Zhiang et al., 2018. Once one of the Q-networks for a local HVAC equipment finds a control policy that would result in a reduced energy consumption, the trained policy is transferred to other similar equipment in the HVAC system (a Q-network for a chiller controls would be transferred to another chiller for example not a boiler) and the transferred policy is used as an initial policy for the equipment as shown in Figure 0-7. Since each equipment is placed in a different zone and each zone has its distinct thermal loads, the transferred RL control policy is trained to find an optimal control policy for the local zone where the equipment is assigned.

The advantage of using transfer learning is decreasing the time consumed on training the RL controller since it starts with a pre-trained initial condition.

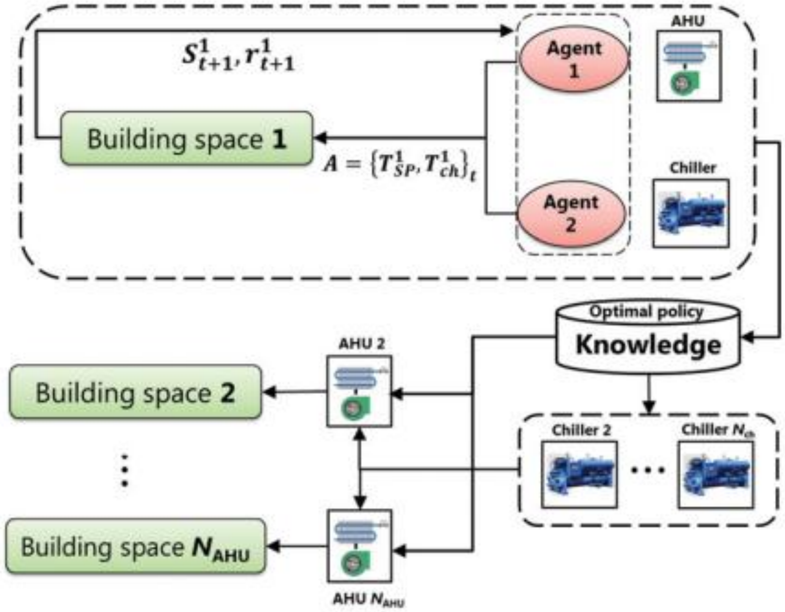

Figure 0-7: Transfer learning framework (Nagarathinam et al., 2020)

Figure 0-8 is a comparison of the resulting thermal discomfort and energy consumption when using different controllers. While both MARCO and BL3 (which is the solution for an optimal control problem that has knowledge of the HVAC system prior to performing the optimization procedures, in contrast to MARCO which is a model-free RL model without prior knowledge of the HVAC system) provide 100% comfort, MARCO consumes just 2% more energy. This confirms that MARCO can learn a policy close to what is practically optimal.

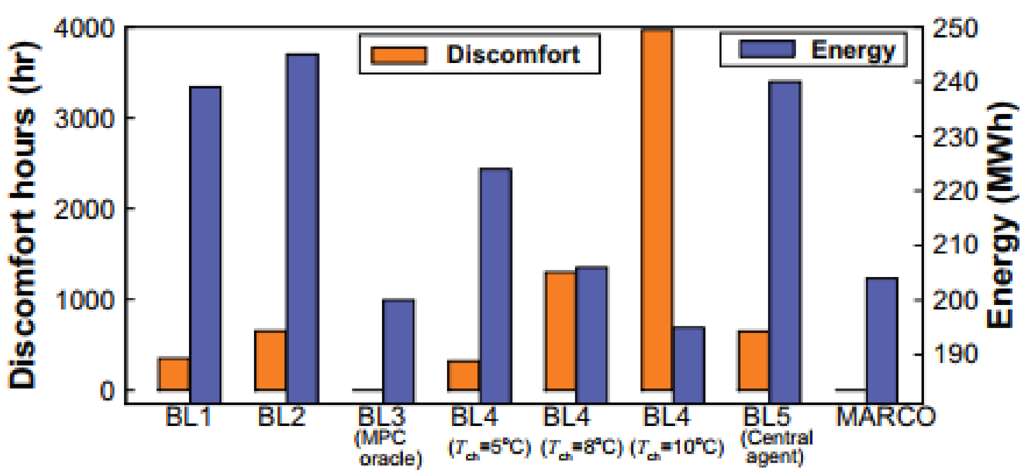

Figure 0-8: Energy consumption and thermal discomfort when using an MARCO vs different controllers (Nagarathinam et al., 2020)

A major drawback though is that model-free RL (like MARCO) needs a considerable period of iterations that may reach up to 14 years of a building life cycle (Xianzhong et al., 2019) to collect enough data about the dynamics of a building environment to converge and find an optimized solution.

Pyfmi

Using pyfmi (see section section 1.2.5) as a python test bed for an HVAC system environment, Lee at al. (2020) compared the performance and the energy consumption of different model-free RL algorithms on a case study of twelve-floor office building. A Reward function whose target is to reduce energy consumption (cost) and increase thermal comfort (reward) of the RL agent is used to direct the agent to select the actions (For this case study, the actions was the HVAC controller setpoints). Energy consumption is computed as the annual energy consumption of the HVAC system of the twelve-floor office building. For thermal comfort, it is computed using predictive mean value approach implemented by Liu et al. (2014) where 10 % PPD indicates that a penalty is given to the RL controller when 10 % of the occupants are unsatisfied.

Since Lee at al. (2020) models an integrated HVAC system for a 12-floor office building, the number of elements in a state space and action space would be very large and it would be impractical to represent them using a Q-table (see section 1.1), therefore, PPO2 and A2C RL algorithms are used where action space and state spaces represented by neural networks (Lee at al., 2020). Episode variable shown on the x-axis of Figure 0-9 represents the RL iterating through a whole year. Since PPO2 is shown to have enhanced energy savings after 105 episodes, it means that for a model-free RL that is implemented on a real-time building, it would take 105 years for the model-free RL controller to obtain favorable results which is not practical (like the problem encountered by (Zhiang et al., 2018). On the other hand, if this was a model-based controller, and the energy model exactly represents the real-time building, this RL controller would be a golden opportunity since RL controller consumes less annual energy than a programmed control sequence. This golden opportunity is challenging though due to the uncertainties in a building energy model as discussed in section 1.1

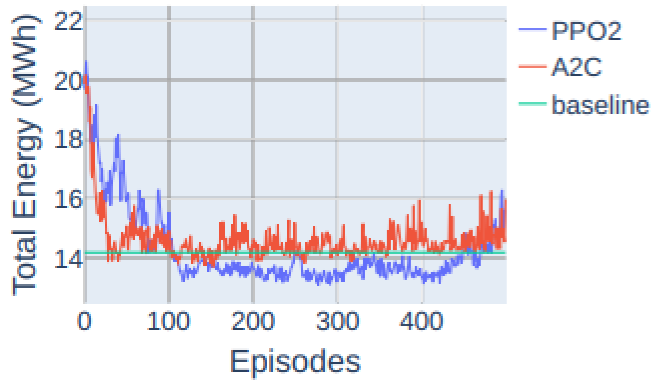

Figure 0-9: Energy consumption when using an RBC vs RL (PPO2 and A2C) controller (Lee at al., 2020)

It is noted though that since is a model-free RL agent, the number of episodes needed to collect is high and is not practicable for a real-time operation which is similar to the problem encountered by Zhiang et al., 2018.

MB²C

In 2020, Ding et al., developed MB²C, a combination of model-free and model-based RL techniques. MB²C consists of two main architectures running in parallel, one is a group of NNs that perform online training (training in parallel while the HVAC system is operating) using the online data collected about a building environment states and selected actions at a given state to predict the building dynamics (predicting the building environment state for the next time step for a given current state and a proposed action as an input to the NN). The second architecture solves a discrete optimization problem to determine which action should be selected at a given building state to obtain minimum energy consumption and maximum thermal comfort (reward). MB²C starts operating as a model-free RL model till it obtains enough data. It is shown by Ding et al. (2020) in Figure 0-10 that MB²C converges faster than model-free RL since it needs less data to do accurate predictions about building environment states for the next time step and to select the action that would result in favorable rewards.

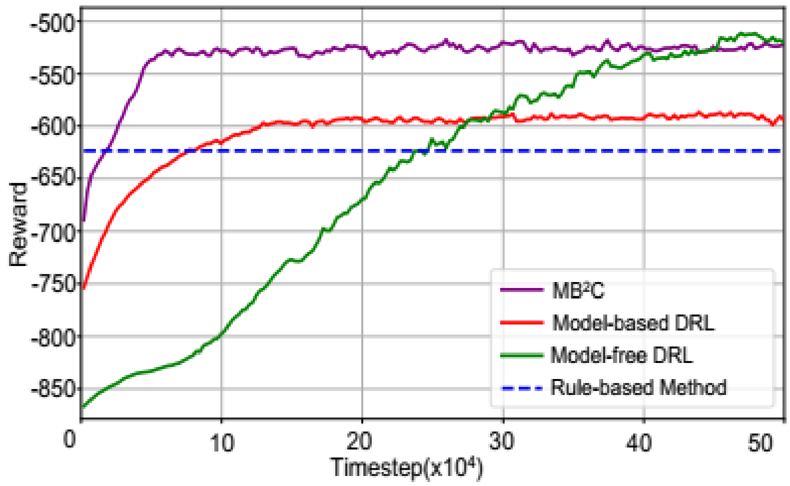

Figure 0-10: Time steps for convergence when using an MB^{2}C vs other controllers (Ding et al., 2020)

On the other hand, MB^{2}C energy savings are slightly smaller than to a trained model-free RL controller as shown in Figure 0-11 since the environment is bounded by the NNs predictions and their accuracy. On the other hand, model-free RL is based on the online data of the building that are the ground truth for the building environment states.

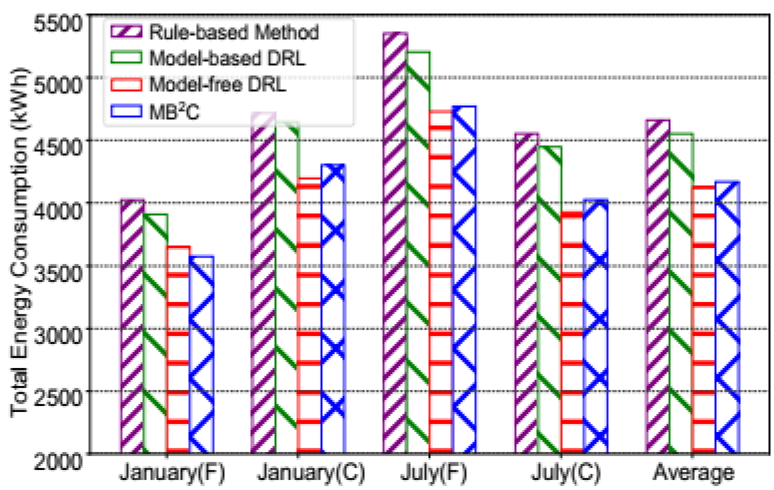

Figure 0-11: Energy consumption when using an MB^{2}C vs other controllers (Ding et al., 2020)

Real time AI for HVAC and recent challenges

To demonstrate real-time applications where an AI is used to operate an HVAC system of a building, a real-time case study is demonstrated. In recent energy summits, companies represented how machine learning, which is a branch of AI is used as an assistive approach along with building energy modeling or an existing buildings HVAC control sequences to generate an adaptive algorithm that finds a control sequence that enhances the energy savings obtained for an HVAC system.

More to add, some application programming interface (API) platforms leverage the data collected about an environment and stored in a cloud (a cloud is similar to a disk storage except it needs online access) to do data analytics (including AI algorithms) and find optimized AI sequences to enhance the energy savings for different applications (solar panels, HVAC, lightning, plumbing).

Some Companies also leverage deep learning neural networks, cloud-based computing and algorithms, to predict a building's thermal load and enables the HVAC system to operate autonomously, for maximum impact on energy consumption and occupant comfort. Model-based RL is used to control the setpoints of an HVAC system where the AI section modules is effective when sufficient data are obtained from a cloud to model the HVAC environment of new or existing building.

It is noted though in most of the real-time applications mentioned that AI wasn't used as stand-alone system that would operate a new system from the beginning of its lifecycle (Zhiang et al., 2018). AI was used as an assistive approach to enhance energy savings where programmable sequences were used to obtain data about a system before AI comes into action. In the introduction, it was stated that utilizing AI for different industries became popular in 20th century.

The main reason for this AI evolution was that most AI algorithms that are used in real time applications like control systems (Xu et al., 2020) are based on an iterative process where computers learn by iterations to do the tasks done by humans.

Before 20th century, computers weren't powerful to do millions of iterations in a reasonable period. In addition, it is noted that in and Nagarathinam et al. (2020) and Lee at al. (2020), a model-free based RL was used where learning process improvement is heavily dependent on the number of iterations where the system explores the environment by doing actions without any prior knowledge of the environment outcomes. For an integrated HVAC system of a building, doing actions without the knowledge of their outcomes may lead to catastrophic outcomes that may not only lead to thermal discomfort of the occupants, but it may also lead to system failure. For example, in a programmed static pressure reset sequence, when a thermal zone temperature gets closer to a thermostat setpoint (thermal load decreases), the following sequence occurs:

- The damper of the VAV supplying this thermal zone with cooled air tends to close
- The static pressure increases in ductwork since closing a damper increases pressure losses
- The flow rate being fed to the thermal zone also decreases
- The static pressure sensor would sense the increase in the static pressure
- The fan speed of the AHU supplying this VAV is reduced
- Energy saving occurs
- Static pressure decreases and returns to its setpoint

For an AI aided HVAC sequence that has no prior knowledge on how to act if the static pressure increases, the AI sequence may do non-relevant actions while learning about the HVAC system environment (Hao et al., 2020), static pressure would then keep increasing till insulation of the ducts is damaged or leakage occurs in ductwork. Even worse, if the exploration action chosen by an AI controller increases fan speed which would further increase the static pressure in ductwork, this may cause the ducts to explode, causing a system failure. The same applies for cooling and heating actions, incorrect actions taken by an AI controller may lead to very low temperatures that would cause ice formation in ductwork which may destroy the cooling coil and the AHU fan. In addition, an exploration action that leads to overheating may cause damage to ducts and sensors. For a model-based RL model case where an AI controller is pretrained before being implemented, energy model uncertainties and data incompleteness (see section 1.1) may cause the pretrained controller to behave as a model-free controller. Therefore, there is no guarantee that the operation of an untrained or a pretrained standalone AI controller, which has no assistance from a programmable control sequence, won't lead to a catastrophic failure of the system.

It is also challenging to determine whether the AI controller or the maintenance technician is liable for a mistake if an AI controller is used. AI systems are sensitive and dependent on data obtained, which means, if a technician fixes a setpoint of a zone for inspection or maintenance purposes, and the AI controller takes an action that would lead to a system failure, it would be challenging to determine which claim for liability is true, the person who designed the AI controller without counting for technicians actions or the technician who was doing a commonly known inspection and made a mistake by overriding the setpoint of an HVAC equipment with an inappropriate value. For a programmable control sequence implemented by a BAS engineer or designer, the consequences of exceptional cases that may occur randomly are counted, while for an AI controller, it has been trained to generalize its action selection based on a given environment dataset, its action are unknown for special cases that are not covered in the given environment dataset.

Conclusion

The supervised, unsupervised and RL categories that classify the types of AI algorithms are reviewed and an HVAC related application for each category is discussed. Since RL category is heavily used for the controllers of HVAC systems, model-free and model-based subcategories that classify RL are reviewed in depth. For both model-free and model-based RL, the advantages and disadvantages of each model are demonstrated based on reviews and experience, commonly used algorithms like Q-learning are discussed, and the applications for each model are explained. It is deduced that model-based RL has leverage over model-free since it has lower risk in terms of the implementation of the RL-controller on a real-time building since it can be pretrained.

An overview is also done about miscellaneous leaning methods like statistical learning and transfer learning. It is observed that these methods aren't mostly used as a stand-alone controller for an integrated HVAC system, but they can be embedded into the architecture of a RL model to improve its performance and reduce its training time. Since model-based RL needs a building model for a new building or BAS data for an existing building to be pretrained, a review is done on the evolution of the programs used to model a building. It is concluded that since python is becoming one of the most trending programming languages for data analytics and AI, standalone programs like Energy Plus and Open Modelica that were implemented before tended to migrate to python by making python libraries that enable the user to interface with these programs through python and use python libraries for AI along with python libraries for energy modeling programs.

Afterwards, a review is done on the different architectures used to interface model-based-free RL and model-free RL algorithms with an HVAC control system environment to implement an operable AI controller. It is concluded that embedding transfer learning in the architecture of the RL controller (like MARCO) reduced the training time. However, even for MB^2C, which has 10X reduction in the data needed for a mode-free RL to be trained, no author guaranteed that actions that would lead to failure are avoided during the period of data collection.

Finally, real-time case studies where AI was implemented to enhance energy savings for buildings are explained. It is concluded that most of the real-time applications use AI as a part of an integrated module that includes an RBC since there is no guarantee how a standalone AI controller would behave and there is no guarantee that an AI controller would avoid the actions that would lead to a system failure at a given state.

Contribution

Review for different AI algorithms, building energy modeling software, theoretical and real-time applications where AI is used to reduce the energy consumption of HVAC systems.

Addressing the critical drawbacks of using stand-alone AI controllers without the assistance of rule-based controllers.

References

1. Asghar A., Palanisamy A., 2012, github.com/OpenModelica/OMPython, Linköping University
2. Azuatalam, D., Lee, W. L., de Nijs, F., & Liebman, A. (2020). Reinforcement learning for whole-building HVAC control and demand response. Energy and AI, 2, 100020.
3. Barrett, E., Linder, S. (2015). Autonomous HVAC Control, A Reinforcement Learning Approach. In: , et al. Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2015. Lecture Notes in Computer Science(), vol 9286. Springer, Cham. https://doi.org/10.1007/978-3-319-23461-8_1
4. Bingqing Chen, Zicheng Cai, and Mario Bergés. 2019. Gnu-RL: A Precocial Reinforcement Learning Solution for Building HVAC Control Using a Differentiable MPC Policy. In Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '19). Association for Computing Machinery, New York, NY, USA, 316–325. doi.org/10.1145/3360322.3360849
5. Blochwitz, Torsten & Otter, Martin & Arnold, Martin & Bausch, Constanze & Clauß, Christoph & Elmqvist, Hilding & Junghanns, Andreas & Mauss, Jakob & Monteiro, Manuel & Neidhold, Thomas & Neumerkel, Dietmar & Olsson, Hans & Peetz, Jörg-Volker & Wolf, Susann. (2011). The Functional Mockup Interface for Tool independent Exchange of Simulation Models. Proceedings of the 8th International Modelica Conference. 105-114. 10.3384/ecp11063105.
6. Booten, C. & Tabares-Velasco, P. C. Using EnergyPlus to Simulate the Dynamic Response of a Residential Building to Advanced Cooling Strategies: Preprint, article, August 1, 2012; Golden, Colorado.
7. Cass S., 2022, spectrum.ieee.org/top-programming-languages-2022, IEEE Spectrum
8. Chen, B., Cai, Z., & Bergés, M. (2019, November). Gnu-rl: A precocial reinforcement learning solution for building hvac control using a differentiable mpc policy. In Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation (pp. 316-325).
9. Chen, Y., Norford, L. K., Samuelson, H. W., & Malkawi, A. (2018). Optimal control of HVAC and window systems for natural ventilation through reinforcement learning. Energy and Buildings, 169, 195-205.
10. Crawley, Drury B. et al. (2001). “EnergyPlus: creating a new generation building energy simulation program”. In: Energy and Buildings 33.4. Special Issue: BUILDING SIMULATION’ 99, pp. 319–331. ISSN: 0378-7788. DOI: doi.org/10.1016/S0378-7788(00)00114-6.
11. Debmalya Biswas, 2020, Reinforcement Learning based HVAC Optimization in Factories. In Proceedings of the Eleventh ACM International Conference on Future Energy Systems (e-Energy '20). Association for Computing Machinery, New York, NY, USA, 428–433. https://doi.org/10.1145/3396851.3402363

12. Ding, X., Du, W., & Cerpa, A. (2020). MB2C: Model-Based Deep Reinforcement Learning for Multi-zone Building Control. Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation.
13. Energy and Buildings, Volume 40, Issue 3, 2008, Pages 394-398, ISSN 0378-7788, doi.org/10.1016/j.enbuild.2007.03.007.
14. Hao, J., Gao, D. W., & Zhang, J. J. (2020). Reinforcement learning for building energy optimization through controlling of central HVAC system. IEEE Open Access Journal of Power and Energy, 7, 320-328.
15. Harmon, M. E., & Harmon, S. S. (1997). Reinforcement Learning: A Tutorial.
16. José R. Vázquez-Canteli, Jérôme Kämpf, Gregor Henze, and Zoltan Nagy. 2019a. CityLearn v1.0: An OpenAI Gym Environment for Demand Response with Deep Reinforcement Learning. In Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '19). Association for Computing Machinery, New York, NY, USA, 356–357. doi.org/10.1145/3360322.3360998
17. José R. Vázquez-Canteli, Stepan Ulyanin, Jérôme Kämpf, Zoltán Nagy, Fusing TensorFlow with building energy simulation for intelligent energy management in smart cities, Sustainable Cities and Society, Volume 45, 2019c, Pages 243-257, ISSN 2210-6707, doi.org/10.1016/j.scs.2018.11.021.
18. José R. Vázquez-Canteli, Zoltán Nagy, Reinforcement learning for demand response: A review of algorithms and modeling techniques, Applied Energy, Volume 235, 2019b, Pages 1072-1089, ISSN 0306-2619, doi.org/10.1016/j.apenergy.2018.11.002.
19. José Vázquez-Canteli, Jérôme Kämpf, Zoltán Nagy, Balancing comfort and energy consumption of a heat pump using batch reinforcement learning with fitted Q-iteration, Energy Procedia, Volume 122, 2017, Pages 415-420, ISSN 1876-6102, doi.org/10.1016/j.egypro.2017.07.429.
20. Julian David Hunt, Andreas Nascimento, Behnam Zakeri, Jakub Jurasz, Paweł B. Dąbek, Paulo Sergio Franco Barbosa, Roberto Brandão, Nivalde José de Castro, Walter Leal Filho, Keywan Riahi, Lift Energy Storage Technology: A solution for decentralized urban energy storage, Energy, Volume 254, Part A, 2022, 124102, ISSN 0360-5442, doi.org/10.1016/j.energy.2022.124102.
21. Kurte, K., Amasyali, K., Munk, J., & Zandi, H. (2021, November). Comparative analysis of model-free and model-based HVAC control for residential demand response. In Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (pp. 309 313).
22. Lee, Joon-Yong, Huang, Sen, Rahman, Aowabin, Smith, Amanda D., and Katipamula, Srinivas. Flexible Reinforcement Learning Framework for Building Control using EnergyPlus-Modelica Energy Models. United States: N. p., 2020. Web.

23. Liu Yang, Haiyan Yan, and Joseph C. Lam. 2014. Thermal comfort and building energy consumption implications – A review. Applied Energy 115 (2014), 164 –173. doi.org/10.1016/j.apenergy.2013.10.062
24. Luis Pérez-Lombard, José Ortiz, Christine Pout, A review on buildings energy consumption information,
25. Michael Wetter, Thierry S Nouidui, David Lorenzetti, Edward A Lee, and Amir Roth. 2015. Prototyping the next generation energyplus simulation engine. In Accepted: 13-th IBPSA Conference. International Building Performance Simulation Association.
26. Michael Wetter, Wangda Zuo, Thierry S. Nouidui and Xiufeng Pang. Modelica Buildings library. Journal of Building Performance Simulation, 7(4), 253–270, 2014. DOI:10.1080/19401493.2013.765506
27. Murugesan, S., Jiang, Z., Risbeck, M. J., Amores, J., Zhang, C., Ramamurti, V., ... & Lee, Y. M. (2020, November). Less is More: Simplified State-Action Space for Deep Reinforcement Learning based HVAC Control. In Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities (pp. 20-23).
28. Nagarathinam, S., Menon, V.P., Vasan, A., & Sivasubramaniam, A. (2020). MARCO - Multi-Agent Reinforcement learning based COntrol of building HVAC systems. Proceedings of the Eleventh ACM International Conference on Future Energy Systems.
29. Robinson, Darren & Haldi, Frédéric & Leroux, Philippe & Perez, Diane & Rasheed, Adil & Wilke, Urs. (2009). CITYSIM: Comprehensive Micro-Simulation of Resource Flows for Sustainable Urban Planning. IBPSA 2009 - International Building Performance Simulation Association 2009.
30. Tao Liu , Chengliang Xu , Yabin Guo , Huanxin Chen , A novel deep reinforcement learning based methodology for short-term HVAC system energy consumption prediction, International Journal of Refrigeration (2019), doi: doi.org/10.1016/j.ijrefrig.2019.07.018
31. Urieli, D., & Stone, P. (2013, May). A learning agent for heat-pump thermostat control. In Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems (pp. 1093-1100).
32. Walter, Emmanuel & Kämpf, Jérôme. (2015). A verification of CitySim results using the BESTEST and monitored consumption values.
33. Watkins, Christopher & Dayan, Peter. (1992). Technical Note: Q-Learning. Machine Learning. 8. 279-292. 10.1007/BF00992698.
34. Wei, T., Wang, Y., & Zhu, Q. (2017, June). Deep reinforcement learning for building HVAC control. In Proceedings of the 54th annual design automation conference 2017 (pp. 1-6).
35. Wetter M., Benne K., Ravache B. (2021), Software Architecture and Implementation of Modelica Buildings Library Coupling for Spawn of EnergyPlus, DOI: doi.org/10.3384/ecp21181325

36. Wetter, M. (2011). Co-simulation of building energy and control systems with the Building Controls Virtual Test Bed. Journal of Building Performance Simulation, 4(3), 185-203.
37. Xianzhong Ding, Wan Du, and Alberto Cerpa. 2019. OCTOPUS: Deep Reinforcement Learning for Holistic Smart Building Control. In Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '19). Association for Computing Machinery, New York, NY, USA, 326–335. https://doi.org/10.1145/3360322.3360857
38. Xu, S., Fu, Y., Wang, Y., O'Neill, Z., & Zhu, Q. (2021, November). Learning-based framework for sensor fault-tolerant building hvac control with model-assisted learning. In Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (pp. 1-10).
39. Xu, S., Wang, Y., Wang, Y., O'Neill, Z., & Zhu, Q. (2020, November). One for many: Transfer learning for building hvac control. In Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation (pp. 230-239).
40. Xu, S., Wang, Y., Wang, Y., O'Neill, Z., & Zhu, Q. (2020, November). One for many: Transfer learning for building hvac control. In Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation (pp. 230-239).
41. Yu, L., Sun, Y., Xu, Z., Shen, C., Yue, D., Jiang, T., & Guan, X. (2020). Multi-agent deep reinforcement learning for HVAC control in commercial buildings. IEEE Transactions on Smart Grid, 12(1), 407-419.
42. Yuan, X., Pan, Y., Yang, J., Wang, W., & Huang, Z. (2021, February). Study on the application of reinforcement learning in the operation optimization of HVAC system. In Building Simulation (Vol. 14, No. 1, pp. 75-87). Tsinghua University Press.
43. Zhang, C., Kuppannagari, S. R., Kannan, R., & Prasanna, V. K. (2019, November). Building HVAC scheduling using reinforcement learning via neural network based model approximation. In Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation (pp. 287-296).
44. Zhang, Z., Chong, A., Pan, Y., Zhang, C., Lu, S., & Lam, K. P. (2018). A deep reinforcement learning approach to using whole building energy model for HVAC optimal control. ASHRAE and IBPSA-USA Building Simulation Conference, 675-682.
45. Zhe Wang and Tianzhen Hong. 2020. Reinforcement learning for building controls: The opportunities and challenges. Appl. Energy 269 (July 2020), 115036. doi.org/10.1016/j.apenergy.2020.115036

46. Zhiang Zhang and Khee Poh Lam. 2018. 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 (BuildSys '18). Association for Computing Machinery, New York, NY, USA, 148–157. doi.org/10.1145/3276774.3276775