Analysis of the deployment strategies of reinforcement learning controllers for complex dynamic systems

Ibrahim Ahmed[1], Marcos Quinones Grueiro[2], and Gautam Biswas[3]

[1,2,3] *Vanderbilt University, Nashville, TN, 37235, USA*
ibrahim.ahmed@vanderbilt.edu
marcos.quinones@vanderbilt.edu
gautam.biswas@vanderbilt.edu

ABSTRACT

This paper benchmarks several strategies for deploying reinforcement learning (RL)-based controllers on heterogeneous hybrid systems. Sample inefficiency is often a significant cost for RL controllers because we need sufficient data to train them, and the controllers may take time to converge to an acceptable control policy. This can be doubly costly if system health is degrading, or if the network of such systems in turn cannot afford a gradually improving controller in its constituents. Learning speed improvement can be achieved via transfer learning across controllers trained on different tasks: simulations, data-driven models, or separate instances of similar systems. This paper discusses near- and far- transfers across tasks of varying similarities. These approaches are applied on a test-bed of models of cooling towers operating on office and residential buildings on a university campus.

1. INTRODUCTION

Commercial buildings in the United States account for 18% of total energy consumption (*Use of energy explained*, n.d.). Of that, a total of 47% is used for refrigeration, ventilation, and cooling (*Energy use in commercial buildings*, n.d.). This presents an attractive target to optimize for minimal environmental and economic cost. With the proliferation of smart building technologies and the internet of things (IoT), access to data pertaining to commercial infrastructure operation has never been easier. In this work data from buildings is used to optimize energy usage for cooling and ventilation.

Heating, Ventilation, and Air Conditioning (HVAC) systems are used to regulate temperature and humidity in large buildings. An HVAC, when cooling, relies on the refrigeration cycle to transport heat from the source (living spaces) to the sink (outside environment). The heat exchange takes place using a fluid refrigerant. It evaporates by absorbing heat from the source, and condenses by expelling it to the sink. Usually water is used as an intermediary transport medium to absorb the refrigerant's heat and expel it into the environment. Water warmed in the condenser flows through a cooling tower where it loses heat via evaporation. Energy is consumed by refrigerant compressor and water pumps in the chiller, and cooling tower fans.

Optimal control of HVACs is a complex problem. There are subsystems, each with multiple control variables which have trade-offs in terms of performance. For instance, speeding up water flow through the cooling tower will result in a smaller temperature decrease but will increase the volume of water in contact with the refrigerant. Similarly speeding up fans will increase air flow which will increase evaporative cooling, but at a marginally decreasing rate. This is further compounded by the unique dynamics of each machine depending on wear-and-tear and environmental factors. A static control policy can be a good heuristic but will be suboptimal over a population of HVAC systems.

Data-driven control offers a solution. Using empirical measurements, a model of the system can be developed. This bypasses the need of complex physical representation of internal dynamics which are ultimately impertinent to the controller. Such a model can be treated as a black box and optimized over the space of control inputs. By capturing common dynamics across applications and reusing learned parameters, a data-driven controller can transfer to another application by fine-tuning on new data.

In this work, the application of a data-driven controller to a cooling tower in a HVAC system is documented. The challenges related to data collection and processing are discussed. Finally the resulting controllers are benchmarked against industry standards.

This paper is organized as follows. Section 2 documents existing theory and applications of HVAC control. The system and approach are described in greater detail in 3. Following that, section 4 presents the evaluation of the proposed methodologies.

2. PRELIMINARIES

2.1. Existing work

This section reviews extant literature on HVAC optimization. First, surveys are documented for context. Second, literature on physics-based modeling of cooling towers is discussed. This is followed by research on optimal control approaches using physical models. Finally, control using data-driven models is discussed.

Optimal control of HVAC systems has been extensively addressed in research work. (S. Wang & Ma, 2008) surveys the landscape of control approaches and categorizes them into local and optimal control. Local control is a rudimentary class of approaches where a system operates based on a rule-set or tracking error with reference signals. Examples include proportional-integral-derivative (PID) control or simple thresholded on/off control. Optimal control seeks to minimize a cost function with respect to overall system performance and controllable variables. The cost function can be based on a physics or data-driven model and then minimized. The cost function can also be implicitly optimized via reinforcement learning to yield a control policy. Another approach is to use an expert system where the control policy represents the optimal points of the cost function.

The survey (S. Wang & Ma, 2008) further documents optimization algorithms used in optimal control. Linear approaches include least squares and its variants. Non-linear optimization is divided into local and global approaches. In local optimization, successive solutions are in each other's vicinity. This includes gradient-free (simplex) and gradient-based approaches (gradient descent, Newton's method, Lagrange multipliers). Global optimization explores solutions all over the domain of the cost function. This includes simulated annealing and evolutionary algorithms.

Model-predictive control (MPC) for HVAC systems is explored in depth by (Afram & Janabi-Sharifi, 2014). The review classifies control approaches four ways. Classical control involves corrective control like PID systems. Hard control includes MPC and optimal control. Soft control encapsulates fuzzy logic and data-driven input-to-control-action mappings like artificial neural networks (ANNs). Hybrid control is a combination of any number of these approaches. For both MPC and optimal control, the system model and/or the cost function need to be optimized. The survey documents approaches including linear programming, genetic algorithms, and particle swarm optimization for control design.

HVAC control is evaluated on several metrics. These include energy and economic savings, smoothness of control actions, thermal efficiency of HVAC systems, computational complexity of controllers, and robustness to disturbances in the environment.

(Jin, Cai, Lu, Lee, & Chiang, 2007) and (Cortinovis, Ribeiro, Paiva, Song, & Pinto, 2009) develop models of mechanical draft cooling towers in HVAC systems from first principles. In the former work, rate of heat rejection from a cooling tower is modeled as a 3-parameter function of entering water temperature, wet-bulb temperature, and flow rates of air and water. The function parameters are learned through Levenberg-Marquardt optimization on the mean squared error (MSE). 1440 points at 1-minute intervals (equivalent to a day's readings) are used for model training. The model is evaluated on data collected on the very next day and a few months after. The relative root mean squared error (RMSRe) remains under 0.1. The latter work models exiting water temperature of a cooling tower as a 3-parameter function of air and water flow rates, environmental conditions, and physical properties of the tower. The model was trained over a dataset of unspecified size and temporal resolution. Over the course of 2 experimental runs prediction errors were limited to 0.3 ° C.

Control based on physical models is done in a follow up work to (Cortinovis, Ribeiro, et al., 2009) in (Cortinovis, Paiva, Song, & Pinto, 2009). The cost function is a sum of economic costs of fans and water pumps. The control variables are fan speed and excess hot water removal rate from the cooling tower. A grid search is done over the domain to optimize cost. They conclude that prioritizing fan speed increase over hot water removal leads to lower overall costs.

In (Sayyaadi & Nejatolahi, 2011), a comparison is drawn between single- and multi-objective optimization approaches for economic and thermal costs for a refrigeration system. The model used is physics-based. There are 8 control variables including flow rates and temperatures. Genetic algorithms are used to find optimal parameters for a single cost function, or a pareto-frontier of parameters for multi-objective optimization. For the case of multi-objective optimized parameters, they deviated less from the economically and thermally ideal points, than did the parameters optimized for a single cost metric.

(Vu et al., 2017) exploits domain knowledge, particularly affinity laws, to develop a composite model of a chiller plant using polynomial regression (PR) and multi-layer perceptrons (MLPs). The model predicts total power consumption as a function of temperature and flow rate of chilled water coming from cooling towers. Models are trained on 15 days of 5-minute measurements. Control variables span a narrow range of values. The training data is augmented by randomly perturbing control variables to aid the model's generalization. As a result the mean absolute percentage error (MAPE) drops from 7.25% to 0.65%. The model is used to find control yielding smallest energy. It is evaluated over 3 months. The prediction error for power consumption drops from around +10% to -10%. This result, however, is ambiguous. It can either mean that the model underestimates power consumption in the new control regime, or that actual power consumed in fact rises.

In (J.-G. Wang, Shieh, Jang, Wong, & Wu, 2014) the objective is the operate a cooling tower fan to conserve energy while maintaining cooling. Data are collected at 5 minute intervals over 5 months. Adaptive models are learned using non-negative garrote optimization. Each model is developed from a small window of past measurements. Under optimal control, the power consumed by the cooling tower goes down but the temperature of the water loop goes up by 3 °C. However, this optimization is local and may cause energy to spike in the overall chiller system.

A more general optimization problem is addressed by (Wei, Xu, & Kusiak, 2014). Total energy cost of 4 chiller plants with different thermal efficiencies is minimized. Control variables are water flow rate, water temperature change, and an on/off switch for each plant. Energy models using MLPs are learned for each plant. Gradient-free optimization is used to find control points. First, a genetic algorithm selects which plants are on, then particle swarm optimization (PSO) selects candidate points using the remaining two control variables. Over 2 days, the predicted energy consumption is 14% less than the measured consumption. However, this may also be an artifact of the energy models being inaccurate out of their training domain.

(Kusiak & Xu, 2012) employ MLPs using autoregressive features for indoor temperature and energy consumption models with PSO. Two MLPs are used with a time-window of features to model temperature and energy consumption. The time window depends on the autocorrelations of each feature. MAPE of less than 0.1% is achieved on both models. Particle swarm optimization with constraints in indoor temperature is used to find optimal control. A 30% reduction in energy is predicted by the models. However, like the previous case, the prediction is not guaranteed to be accurate.

2.2. Reinforcement learning

Reinforcement learning (RL) is a semi-supervised machine learning approach. It relies on a controller interacting with an environment which yields feedback: a reward signal. The optimization objective is to select control actions to maximize cumulative rewards over time. The function that selects control actions is called a policy π.

A RL task can be represented by a Markov Decision Process (MDP). An MDP consists of states (x), actions (u), a reward function ($r_t \leftarrow R(x_t, u_t, x_{t+1})$, and a state transition function ($x_{t+1} \leftarrow T(s_t, u_t)$). The functions can be stochastic. Using these, the optimal action at time $t = \tau$ becomes:

$$u_\tau \leftarrow \arg\max_u \left(r_\tau + \sum_{t=\tau+1}^{\inf} \gamma^{t-\tau} r_t \right) \tag{1}$$

Where $\gamma \in \mathbb{R}[0, 1]$ is a discount factor to prioritize immediate rewards. The cumulative rewards of optimal actions proceeding from a state are its value V. Equation (1) is recursive an can be solved via dynamic programming, as first introduced by (Bellman, 1966). Later improvements such as Q-Learning (Watkins & Dayan, 1992) iteratively tabulated the cumulative rewards (i.e. values) of actions to then derive the most rewarding action. Later still, value-function approximations were used with the help of neural networks (Mnih et al., 2013) to great success. This process of estimating state and action values so the most valuable one can be picked is called policy iteration.

Policy gradient approaches (Sutton, McAllester, Singh, Mansour, et al., 1999) directly iterate over a policy function $u \leftarrow \pi_\theta(x)$ parametrized as θ. They bypass the need of value function approximation to evaluate each state. This is specially useful for continuous action spaces. Proximal Policy Optimization (Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017) is one such approach, where the policy function outputs action probabilities, and which takes care not to change the policy drastically with each iteration of the optimization process.

2.3. Transfer learning

Transfer learning methods are designed to automatically build prior knowledge from the solution of a set of source tasks (i.e., training tasks) to be used during the learning process on a new task (i.e., testing task). The idea is to retain and reuse the knowledge across different but related tasks to improve the learning performance.

Formally, we define a RL task $M \sim \Omega$ as a MDP, where Ω represents the distribution from the available space of tasks. The goal of a transfer learning algorithm is to extract knowledge from a set of L_t source tasks to improve the learning process and/or performance on a target task M_t.

Typically there are three performance metrics considered for transfer learning problems: jump-start improvement, asymptotic improvement, and learning speed improvement. The first one measures the initial performance of a policy compared to random initialization. The second one measures the improvement of the final performance achieved by the policy. The third one measures the efficiency of learning by reducing the required interactions with the environment.

3. APPROACH

In this section, the overall problem and approach to a solution are described.

3.1. System Description

In this work, two mechanical draft cooling towers are analysed. A cooling tower is a terminal component of HVACs. Cooling towers expel heat from a chiller into the environment. A chiller is the central heat-exchange mechanism of a HVAC system. It uses either a vapor compression or an absorption-refrigeration cycle to extract heat into the refrigerant and generate chilled water which is supplied to a building. The hot refrigerant gas then condenses and expels its heat into another water loop. That loop passes through a cooling tower where the refrigerant's heat is dissipated into the environment.

A cooling tower primarily uses evaporation, and conduction and radiation secondarily, to get rid of excess heat from water. In a mechanical draft cooling tower, fans circulate air through a column while hot water falls under gravity. The contact between air and water leads to heat exchange. Fills can be added inside the tower column to increase contact surface area and time for more heat exchange. At the bottom of the tower cool water is collected and circulated back to extract excess heat from the chiller.

Evaporation depends mainly on three factors: temperature of water, surface area in contact with air, and partial pressure of water in air. Warmer water molecules have more kinetic energy and will escape into air faster. A larger surface area means a greater mass of water will evaporate in the same time period. Conversely, higher partial pressure of water in air (corresponding to high humidity) will reduce evaporation. Therefore dry air, or fast-blowing air such that humid air is displaced over water faster, will increase rate of evaporation.

The maximum amount of cooling possible depends on the wet-bulb temperature (T_{wb}) of ambient air. Wet-bulb temperature is the point at which air will become fully saturated with water vapor and will not be able to absorb more water. Evaporation will not be possible. Therefore the lowest temperature of water exiting from a cooling tower is bounded by the wet-bulb temperature.

Figure 1 illustrates the cooling tower and pertinent variables used in this work. Controllable variables in a cooling tower are the fan speeds for air flow, and condenser pump for water flow rate.

The cooling towers operate on a campus building, where each tower is attached to an 800-ton chiller. The towers operate one at a time. Each tower has two variable frequency drive fans which run in unison.

3.2. Problem Description

The overarching goal is to train controllers that can adapt fast to a new environment, either as a result of a fault or a result of a new deployment. The control objective is to maximize temperature drop of water passing through the cooling tower ($T_{ct,i} - T_{ct,o}$), whilst keeping the fan power ($P_{ct,f}$) low. The hypothesis is that the cooler the water flowing into the chiller's evaporator unit, the more efficiently will heat be exchanged with the refrigerant. Given that the bulk of energy consumption of an HVAC is attributed to the chiller, a marginal drop in water temperature will have a multiplicative effect on net energy usage.

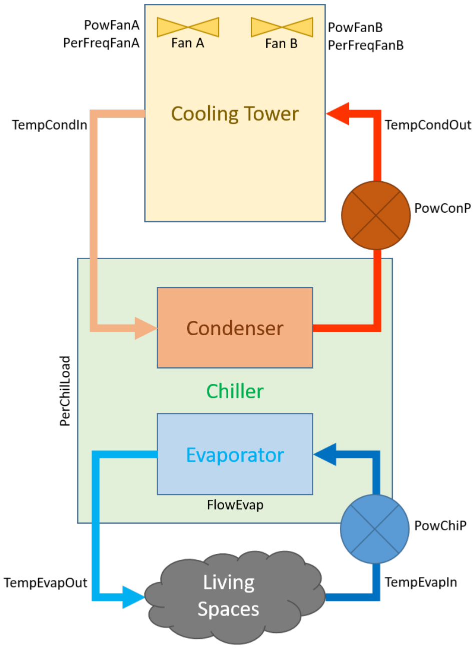


Figure 1. Schematic of an HVAC system showing relationships between components and measured variables.

According to Newton's law of cooling, the rate of cooling is proportional to the instantaneous temperature differential with the surroundings. In this case the differential is relative to the differential with the wet-bulb temperature ($T_{ct,i} - T_{wb}$). If the marginal cooling with increase in fan speed is not positive, there is no utility in turning the cooling tower fan higher.

For this application, control is exercised through the temperature setpoint for water coming out of the cooling tower ($T_{ct,o}$). The internal logic of the HVAC uses the setpoint and an obfuscated PID controller to modulate fan speeds.

First, a data-driven model of the cooling tower is learned to predict exiting water temperature as a function of control variables. Then an optimal control policy is developed by exploring the control space. Finally the policy is evaluated in a data-driven environment of the cooling towers.

3.3. Data

Data for each cooling tower were collected from the HVAC system installed at the Engineering Science Buiding at Vanderbilt University. Measurements were taken at 5 minute intervals. Table 1 documents fields in the dataset.

Table 1. Original and derived fields in dataset and corresponding nomenclature.

Column Name	Variable	Description
	$F_{ct,w}$	Cooling tower water flow rate
PressDiffCond	$D_{ct,p}$	Differential pressure in condenser loop pump
	$F_{ct,a}$	Air flow rate through cooling tower
PerFreqFanA		Fan frequency as a percentage of maximum
PerFreqFanB		Fan frequency as a percentage of maximum
	F	Average fan frequency as a percentage of maximum
PerHumidity		Relative humidity
PowChi	P_{ch}	Power consumed by chiller
PowChiP	$P_{ch,p}$	Power consumed by chiller water pump
PowConP	$P_{ct,p}$	Power consumed by cooling tower water pump
Tonnage	L	System Load. Rate of heat extraction from building
PowFanA	$P_{ct,fa}$	Power consumed by fan A
PowFanB	$P_{ct,fb}$	Power consumed by fan B
PowFan	$P_{ct,f}$	Average power consumed by fans
TempAmbient	T_a	Ambient air temperature
TempCondIn	$T_{ct,i}$	Temperature of water out of the cooling tower
TempCondOut	$T_{ct,o}$	Temperature of water into the cooling tower
TempEvapIn	$T_{ch,i}$	Temperature of incoming chilled water
TempEvapOut	$T_{ch,o}$	Temperature of cooled chilled water
TempWetBulb	T_{wb}	Wet-bulb temperature
Setpoint	S	Setpoint for $T_{ct,o}$

3.4. Modeling Cooling Tower Temperature

From the theoretical discussion in section 3.1, and the physical models developed by (Jin et al., 2007) and (Cortinovis, Ribeiro, et al., 2009), the exiting water temperature of the cooling tower $T_{w,o}$ is modeled as a function of incoming water temperature $T_{w,i}$, ambient temperature T_a, wet-bulb temperature T_{wb}, air flow rate S_a, and water flow rate $S_{w,ct}$. In this case, correlated variables are used to reflect the availability of data:

$$T_{ct,o} = f(T_{ct,i}, T_a, T_{wb}, F_{ct,a}, F_{ct,w}) \tag{2}$$

$$T_{ct,o} = f(T_{ct,i}, T_a, T_{wb}, D_{ct,p}) \tag{3}$$

A multi-layer perceptron (MLP) is chosen to model this function. A MLP, also known as a feed-forward neural network, is a time-invariant mapping from input features to output targets (unlike recurrent neural networks, which have temporal dependencies). A MLP can act as a universal function approximator over a compact real space (Hornik, 1991).

A physical model of energy rejection dQ/dt by a cooling tower, developed by (Jin et al., 2007), can be written as:

$$\frac{dQ}{dt} = \frac{c_1 {F_{ct,w}}^{c_3}}{1 + c_2 \left(\frac{F_{ct,w}}{F_{ct,a}}\right)^{c_3}} (T_{ct,i} - T_{wb}) \tag{4}$$

Where $dQ \propto (T_{ct,o} - T_{ct,i})$, and (c_1, c_2, c_3) are learnable constants. Assuming slowly changing flow rates and ambient conditions, the solution is an exponential function of $T_{ct,i} - T_{wb}$. This can be modeled by a MLP. The model in equation 3 substitutes flow rates with differential pressure, and implicitly models the fan speed control logic from ambient conditions.

3.5. Reinforcement learning environment

The RL environment's dynamics are derived from the previously described model. The state vector of the environment has three categories of variables. First, independent ambient variables (T_a, T_{wb}) change regardless of control actions and describe the extraneous phenomena. Secondly, independent system variables $(D_{ct,p}, L)$ change at the behest of other controllers. Finally, the dependent system variable $(T_{ct,i})$ is a result of the previous state and control action.

In consideration of the optimization objective, the model in equation 3 is augmented as in equation 5. The tonnage variable L reflects the overall load of the chiller system and the amount of heat extracted from the building. The additional outputs $P_{ct,f}, T_{ct,i}$ are used to predict the power consumption to optimize, and the water temperature into the cooling tower for the next time interval, after exchanging heat with the chiller.

$$T_{ct,o}, P_{ct,f}, T_{ct,i} = f(T_{ct,i}, T_a, T_{wb}, D_{ct,p}, L) \tag{5}$$

Each episode of the environment constitutes a 24 hour period divided into 5 minute intervals for a total of 288 time steps. A ticker tape of independent variables is fed to the state vector at each time step. The model is used to predict the dependent variable, and the inputs to the reward function. The model and the ticker together make up the state transition function ($x_{t+1} \leftarrow T(s_t, u_t)$).

Due to the stostically trained model, the outputs may not always fulfil physical constraints. In which case the environment clips outputs of the neural network model to adhere to physical laws, described in equation 6.

$$\begin{aligned} T_{ct,o} &= \max\left(\min(T_{ct,o}, T_{ct,i}), T_{wb}\right) \\ T_{ct,i} &= \max(T_{ct,o}, T_{ct,i}). \end{aligned} \tag{6}$$

The reward function optimizes for a high cooling tower efficiency $0 \leq E_{ct} \leq 1$ and a low fan power consumption, $0 \leq p_{ct_f} \leq 1$ which is the nominal power consumption $P_{ct,f}$ scaled to $[0, 1]$. Equation 7 describes the feedback the controller receives for each action.

$$\begin{aligned} x &= [T_{wb}, T_a, T_{ct,i}, L, D_{ct,p}] \\ u_t &= [S] \\ x_{t+1} &= T(x_t, u_t) \\ E_{ct} &= \frac{T_{ct,i} - T_{ct,o}}{T_{ct,i} - T_{wb}} \\ R(x_t, u_t, x_{t+1}) &= E_{ct} - p_{ct,f} \end{aligned} \tag{7}$$

3.6. Training Data-driven model

The extant setpoint logic for the cooling towers follows a fixed approach controller scheme, wherein $S \leftarrow T_{wb} + a$. Where a is a margin acknowledging the inefficiency of the cooling process. An approach too small will cause the fans to spin needlessly towards an unachievable cooling performance. An approach too large will leave room for improvement. To explore this, building administration instituted periods where setpoint was fixed or varied very little. The highly bi-modal nature of data can be seen in figure 2. The setpoint values do not capture the full breadth of system operation. Therefore the environment model's interpolation for missing values will be inaccurate.

To ameliorate data sparsity, a simple feedback controller, henceforth known as "Up-Down" controller was deployed. The controller is parametrized by the step size of the setpoint change ΔS, and the choice of feedback function which in this case was $T_{ct,o}$. Table 2 tabulates the logic of the controller. A positive feedback direction causes setpoint direction to maintain. A negataive feedback direction causes setpoint direction to reverse. Figure 2 shows the distribution of setpoints before and after deployment of the feedback controller.

Table 2. Logic of the simple feedback "Up-Down" controller. The first two columns are the recorded changes in feedback and action. The last column is the next direction of change in action.

$(\Delta T_{ct,o})_t$	$(\Delta S)_t$	$(\Delta S)_{t+1}$
0	0	random
0	+	random
0	-	random
+	0	random
+	+	+
+	-	-
-	0	0
-	+	-
-	-	+

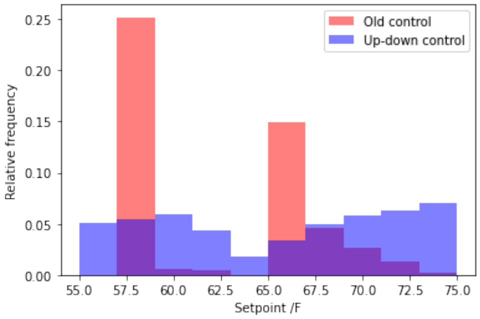


Figure 2. The setpoint distribution under the extant controller is highly bi-modal. An intermediate feedback controller was deployed to capture system dynamics.

4. EXPERIMENTS

This section documents experiments carried out on environments learned using the data-driven models. The experiments evaluated the utility of transfer learning in different scenarios. For each transfer experiment, a RL controller trained in one environment was later trained on another. Secondly, a controller was first trained on a model of the second environment learned from 10% of the data and for 10% of the training steps, and then trained on the second environment. This was to evaluate the utility of preconditioning the controller for the new environment.

- Across equipment,
- Across ambient conditions,
- Across sparsity levels in data.

Each experiment was evaluated by bench-marking reinforcement learning performance during operation. The trained controllers were run over ten days' worth of episodes and the rewards were aggregated. The controllers for comparison used:

1. RL trained from scratch on the new environment,
2. "Up-Down" logic,

3. Fixed approach ($a = 5$),
4. Model predictive control with a 1-step horizon.

Controllers were first trained on data collected using the "Up-Down" controller for each cooling tower. Figure 3 shows the control behavior under identical independent state variables over a single day. Both controllers achieve high rewards per interval. However the actions taken are different. This demonstrates a knowledge gap across environments that transfer learning can solve. Figure 4 is the aggregate operational performance of various controllers on the transfer target: tower 2. The highest total rewards are from RL controllers trained natively on tower 2 and transferred from tower 1 to 2.

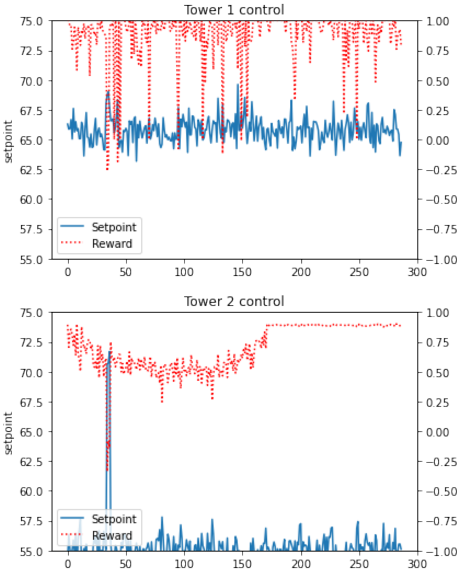


Figure 3. Setpoints for each cooling tower over the course of 24 hours.

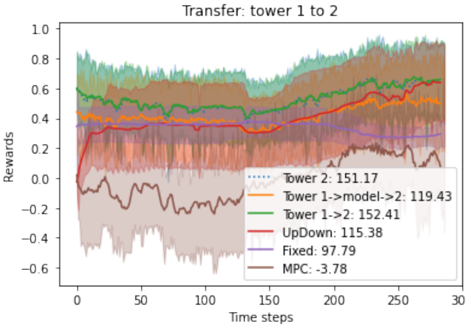
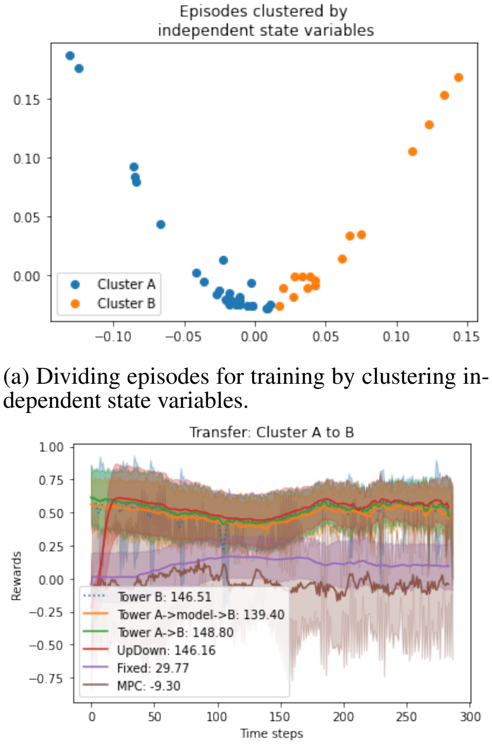


Figure 4. Performance of controller trained on cooling tower 1, later trained on cooling tower 2.

For the second set of experiments, controllers were trained on data from different operating conditions of the same cooling tower (tower 2). Figure 5a shows how data were put into two clusters for each controller to train on. The clusters were generated by calculating similarity measures between T_{wb}, T_a, L independent variables for each pair of days. Dynamic time warping was used to measure similarities. Then spectral clustering was used to divide episodes into two groups, A and B. The objective of the experiment was to transfer controller learned from cluster A to cluster B.

Figure 5b illustrates control performance over multiple episodes. The highest performing are RL controllers and the "Up-Down" controller.



(a) Dividing episodes for training by clustering independent state variables.

(b) Performance when transferring across state variable clusters.

Figure 5. Transfer across clusters of independent state variables.

The choice to use diverse setpoint data by deploying the Up-Down controller was validated by observing the quality of transfer of the data-driven environment model, and the eventual RL transfer performance. Figure 6 shows that the transfer from sparse to diverse setpoint data sets has the largest transfer gap left. For model transfer across towers in figure 6a, the transfer gap is large but is overcome due to the richness in the training data. For the transfer problem on the same tower but under different state variable distributions as shown in figure 6b, the transfer gap is small and easily overcome.

The effects of transfer gap in environment modeling manifest in the RL performance as well (figure 7), where the total reward difference between natively trained and transferred controllers on the target task is the highest for the case sparse to diverse data transfer.

Of note in all experiments is the poor performance of MPC control and fixed-approach controller. The former is explained by the data-driven model not being accurate and respecting physical constraints between temperatures as discussed earlier. Therefore the MPC controller's internal environment model my predict inaccurate states and feedback valuations which lead to suboptimal action choices. For fixed approach controllers, the fixed approach can be too ambitious, causing a power penalty, or be too lax, causing an efficiency penalty.

5. CONCLUSION

This paper presented an applied approach to developing data-driven controllers for a class of HVAC systems with operational differences due to degradation and incipient faults. Challenges with data processing and modeling were presented, especially the need for representative data for modeling a data-driven controller. Finally the utility of using RL and transfer learning was demonstrated in relation to industry standard approaches like fixed-approach and model-predictive controllers. The transfer gap, in terms of model predictions and RL controller rewards, between source and target tasks was smaller when sufficient data was available for capturing environment dynamics during training. Future venues for research include codifying what pairs of tasks are considered near or far for transfer and how to adjust learning strategies to ameliorate any handicaps that it may entail.

APPENDIX

This section documents the hyper-parameters used for experiments. The following hyperparameters were used for the RL agent:

- Learning rate: 3×10^{-4},
- Discount factor: $0.$,
- PPO Value/Policy network architecture: $64, 64$ hidden units,
- Activation: $\tanh$,
- Optimization algorithm: Adam,
- Update interval: 150 steps,
- Learning iterations per update: 10,
- RL training timesteps: 288×30 (One month of simulation).

The following hyper-parameters were used to model the environment (state transition function):

- Network architecture: $32, 32, 32$ hidden units,
- Learning rate: 10^{-3},
- Activation: ReLU,
- Optimization algorithm: Adam.

For consistency, neural network parameter initialization was identically seeded for RL agents.

Code for this research can be obtained from `https://git.isis.vanderbilt.edu/SmartBuildings/EngineeringScienceBuilding`.

REFERENCES

Afram, A., & Janabi-Sharifi, F. (2014). Theory and applications of hvac control systems–a review of model predictive control (mpc). *Building and Environment*, *72*, 343–355.

Bellman, R. (1966). Dynamic programming. *Science*, *153*(3731), 34–37.

Cortinovis, G. F., Paiva, J. L., Song, T. W., & Pinto, J. M. (2009). A systemic approach for optimal cooling tower operation. *Energy Conversion and Management*, *50*(9), 2200–2209.

Cortinovis, G. F., Ribeiro, M. T., Paiva, J. L., Song, T. W., & Pinto, J. M. (2009). Integrated analysis of cooling water systems: Modeling and experimental validation. *Applied thermal engineering*, *29*(14-15), 3124–3131.

Energy use in commercial buildings. (n.d.). `https://www.eia.gov/energyexplained/use-of-energy/`.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, *4*(2), 251–257.

Jin, G.-Y., Cai, W.-J., Lu, L., Lee, E. L., & Chiang, A. (2007). A simplified modeling of mechanical cooling tower for control and optimization of hvac systems. *Energy conversion and management*, *48*(2), 355–365.

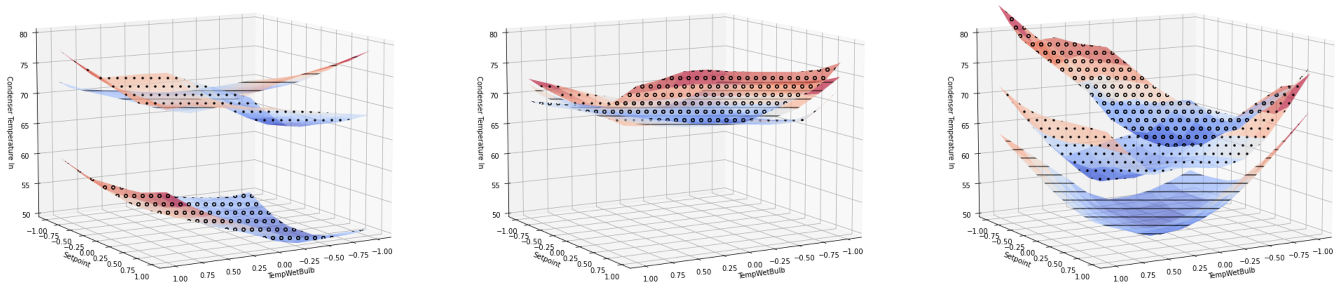
Kusiak, A., & Xu, G. (2012). Modeling and optimization of hvac systems using a dynamic neural network. *Energy*, *42*(1), 241–250.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Sayyaadi, H., & Nejatolahi, M. (2011). Multi-objective optimization of a cooling tower assisted vapor compression refrigeration system. *international journal of refrigeration*, *34*(1), 243–256.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sutton, R. S., McAllester, D. A., Singh, S. P., Mansour, Y., et al. (1999). Policy gradient methods for reinforcement learning with function approximation. In (Vol. 99, pp. 1057–1063).



(a) Tower 1 to 2

(b) Cluster A to B

(c) Sparse to diverse

Figure 6. Model transfer of $T_{ct,o}$ for different experiments. The surfaces are the predictions as the S and T_{wb} axes are varied. All other state variables are identical. The 'o' markers are the transfer source. The '/' markers are transfer target. The '.' markers are the transfer result.

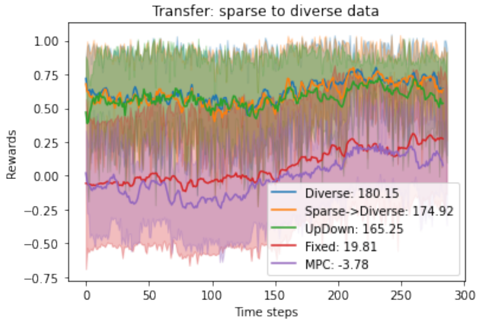


Figure 7

Use of energy explained. (n.d.). `https://www.eia.gov/energyexplained/use-of-energy/`.

Vu, H. D., Chai, K. S., Keating, B., Tursynbek, N., Xu, B., Yang, K., … Zhang, Z. (2017). Data driven chiller plant energy optimization with domain knowledge. In *Proceedings of the 2017 acm on conference on information and knowledge management* (pp. 1309–1317).

Wang, J.-G., Shieh, S.-S., Jang, S.-S., Wong, D. S.-H., & Wu, C.-W. (2014). Data-driven modeling approach for performance analysis and optimal operation of a cooling tower. *Journal of the Taiwan Institute of Chemical Engineers*, *45*(1), 180–185.

Wang, S., & Ma, Z. (2008). Supervisory and optimal control of building hvac systems: A review. *HVAC&R Research*, *14*(1), 3–32.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*(3-4), 279–292.

Wei, X., Xu, G., & Kusiak, A. (2014). Modeling and optimization of a chiller plant. *Energy*, *73*, 898–907.