PACMAN – Physics-Aware Control MANager for HVAC

Srinarayana Nagarathinam, Yashovardhan S. Chati, Malini Pooni Venkat, Arunchandar Vasan
TCS Research
IIT-Madras Research Park, Chennai, 600113, India

ABSTRACT

Indoor user thermal comfort mainly depends upon air temperature, humidity, and wall temperature. Commonly used proportional-integral-derivative (PID) controllers use air temperature alone as feedback to control building Heating, Ventilation, and Air-Conditioning (HVAC) equipment. This may result in sub-optimal energy consumption or user comfort. Model Predictive Controllers (MPC) improve over PID control, but are limited by the accuracy of the underlying thermal model. Black-box (data-driven) and grey-box (data+domain-driven) thermal models may handle non-linear building thermal dynamics, but require extensive data and may not adhere to physical constraints.

We complement existing works with *PACMAN* – an MPC approach that uses Physics-Informed Neural Network (PINN)-based thermal models to estimate humidity and wall temperature in addition to air temperature. We first adapt PINNs to building HVAC control by overcoming the limitation of the original PINN formulation. Specifically, we build a PINN thermal model that can handle time-dependent exogenous (ambient temperature) and control (setpoint) inputs through a time resetting strategy. We then demonstrate the use of a PINN thermal model in MPC with a receding horizon.

We evaluate *PACMAN* using a simulated environment along two dimensions: 1) thermal model accuracy; and 2) control efficacy. As a baseline for the thermal model, we use a data-driven LSTM model. As baselines for the control, we use an as-is PID controller with fixed and seasonal temperature setpoints; oracle MPC without any thermal model errors; and MPC with an LSTM model. We find that the PINN thermal models improve errors over LSTM by an order of magnitude; and generalize better to out-of-distribution data. *PACMAN* reduces the annual energy consumption by 16% and the percentage of yearly discomfort hours by 26% points over as-is PID control with a fixed setpoint.

CCS CONCEPTS

• **Computing methodologies** → **Simulation evaluation**; • **Applied computing** → **Physics**.

KEYWORDS

physics-informed, neural network, receding horizon, building, HVAC, control, simulation

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3564052

ACM Reference Format:
Srinarayana Nagarathinam, Yashovardhan S. Chati, Malini Pooni Venkat, Arunchandar Vasan. 2022. *PACMAN* – Physics-Aware Control MANager for HVAC. In *The 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3563357.3564052

1 INTRODUCTION

Buildings contribute significantly to energy consumption and carbon footprints. Because HVAC alone accounts for 40% of building energy consumption [4], energy management for HVAC systems has received significant research attention. Most HVAC deployments currently use PID control that uses only an error between the realized and target (setpoint) air temperatures to operate AHUs and chillers. PID control is widespread due to the ease of implementation in firmware. Therefore, the main focus in HVAC energy management has been to save energy through intelligent control that improves over PID-based as-is control even while meeting occupant comfort requirements [1].

Unmeasured thermal parameters: Intelligence in HVAC control improves over PID control by using additional sensing parameters (e.g., occupancy) [7, 13] and/or through optimisation in control (e.g., model-predictive [11] or reinforcement learning based [29, 37]). Since additional sensory deployments or extensive measurements may not be readily possible, the main focus generally has been on better methods for optimisation in control. However, in practice, some parameters that do not have sensory readings can nevertheless be critical in determining system performance irrespective of the control strategy used. Specifically, consider the surface temperature of the walls of a building. In most deployments, sensors typically exist only for the room air temperature (T_a), room humidity (W_a), supply and return temperatures, but not the wall temperature (T_w). Several research works estimate wall temperature using a functional approximation involving outdoor ambient (T_∞) and indoor air (T_a) temperatures [14, 28] or use T_a as a proxy for T_w in thermal comfort calculations [32]. However, as we show later, using a wrong value for T_w (e.g., with 5% error) can cause percentage discomfort to increase (by 80% points) and energy consumption to increase/decrease (by up to 13%). A correct estimation of such unmeasured critical parameters in the absence of hardware sensing can improve the performance of *any* intelligent control strategy.

Better thermal models for MPC: MPC methods [11] improve over PID control, but are highly dependent on the accuracy of the thermal models used to predict the parameters of interest, namely, T_a, T_w, and W_a. One approach to estimating these parameters could be to use purely data-driven models. However, such models are data-intensive and need extensive training and calibration. Moreover, they are not guaranteed to adhere to physical laws governing system dynamics. In this work, we complement existing approaches

by using PINNs-based models for estimating the thermal parameters. PINNs [34] encode the governing differential equations of the underlying system in the backpropagated gradient of the neural network using automatic differentiation. They thus capture the underlying physics of the model. The governing equations require only readily available data in the form of building material constants such as U-value and thermal capacitance. Therefore, PINNs bypass the need for extensive measurements to learn pure data-driven models of thermal parameters. PINNs also serve as a means to soft sense the unmeasured thermal parameter T_w by modelling the governing equation of T_w. Further, once trained, PINNs are much faster than numerically solving differential equations for modelling the dynamics of the system, particularly if the physics is governed by the Navier-Stokes equations, at every control step as the system state update would simply be a forward pass through a neural network for the next time step.

Gaps in existing works in using PINNs for control: Using PINNs to control building systems is non-trivial for several reasons. **(1)** The basic formalism of PINNs [34] is not readily amenable to control problems. Specifically, exogenous inputs (e.g., ambient temperature) and control decisions (e.g., mass flow rates) need to be specified as *functional* inputs to the neural network, which may not be known a priori. The input feature space could be very high dimensional depending upon the duration (monthly, yearly, etc.) and the (min-max) range of the inputs. **(2)** The efficacy of MPC [21] using PINNs is mostly unexplored. **(3)** To the best of our knowledge, most works on physics-constrained neural networks for building thermal models do not account for evolution in humidity [18, 30], which is an important parameter in thermal comfort calculation.

Our approach: We overcome these challenges and complement existing works by adapting PINNs for building HVAC control while explicitly estimating all critical thermal parameters, including the unmeasured T_w and the often neglected W_a. We present *PACMAN* that employs a PINN-based controller in an MPC framework for HVAC energy minimisation even while meeting thermal comfort requirements. We extend the PINN formalism for control problems with time-varying exogeneous and control inputs through a time resetting strategy and a zero-order hold assumption. *PACMAN recommends temperature setpoints that are used as reference values in the underlying PID controller and therefore does not require altering the implementation in firmware.*

To summarize, our **specific contributions** include the following:

- We contribute to the body of PINN literature by adapting PINNs for modelling the building's thermal dynamics and control of HVAC. As a baseline for PINN, we use a purely data-driven Long-Short Term Memory (LSTM) model for estimating the thermal parameters (T_a, T_w, and W_a).
- We demonstrate the need for a high-accuracy thermal model for MPC using experiments. Specifically, we consider model errors between ±5% in steps of 1% in T_a, T_w, and W_a. The metrics evaluated are the annual HVAC energy consumption and the percentage of HVAC operating hours that occupants' comfort is unmet (acceptable comfort being PPD $\leq$ 10%).
- We implement PINN in a simulated environment using real-world building data, including the associated HVAC design and building material properties. The simulator acts as the ground truth where the governing equations are solved using an ODE integrator (fourth-order Runge-Kutta method RK4). The PID controller is tuned using the Ziegler-Nichols technique [12].
- We study the sensitivity of PINN to the time reset length τ. We compare *PACMAN* 's performance with the following baselines: (BL1) as-is control that uses a fixed temperature setpoint of 23 ± 1°C throughout the HVAC operation, (BL2) control that uses seasonal temperature setpoints of 22 ± 1°C during summer months and 23 ± 1°C for the rest of the year, (BL3) Oracle MPC with receding horizon control that assumes perfect knowledge of the system (zero model error), controller, and exogenous disturbances, and (BL4) MPC with the LSTM model.

Our **key findings** are as follows:

- MPC method is equally sensitive to model errors in T_a and T_w and less sensitive to those in W_a. Model errors affect discomfort by up to 80% points and annual energy consumption by up to 13%. This shows T_w, while unused, is equally important.
- *PACMAN* 's PINN model performs better than the baseline LSTM model. Furthermore, LSTM gives unphysical results for W_a while PINN respects physics at all times. When trained on a limited range of inputs, PINN generalized better than LSTM for out-of-distribution data. LSTM gave larger errors (23, 19, 29)% compared to PINN (3, 2, 3)%.
- In the control problem, *PACMAN* performs better than BL1 and BL2. Over BL1, *PACMAN* saves 16% energy and improves percentage comfort by 26% points. Over BL2 *PACMAN* saves 19% energy and improves percentage comfort by 24% points. Compared with BL3 ('oracle' MPC), *PACMAN* gives nearly the same unmet comfort hours for 6% more energy. Compared with BL4 (MPC + LSTM), *PACMAN* consumes 24% less energy and improves comfort by 2% points.
- Finally, we find that the PINN thermal model is sensitive to the time reset length τ. The training runtime varies from 7.3 hours at $\tau = 15$ minutes to 1.2 hours at $\tau = 60$ minutes. Given a simulation time step of 15 minutes, the thermal model error increases with increasing τ, from (0.4, 0.2, 0.7)% at $\tau = 15$ minutes to (7.8, 2.4, 4.3)% at $\tau = 60$ minutes for the parameters (T_a, T_w, W_a).

Paper outline: Section 2 presents related work. The optimal control problem is presented in Section 3. The solution strategy is discussed in Section 4, along with the algorithms. Section 5 presents the experimental design. The findings are presented in Section 6, followed by limitations and conclusions in Section 7 and Section 8, respectively.

2 RELATED WORK

Studies on optimal control of HVAC in buildings can be broadly grouped into thermal model development; model-assisted and model-free based optimal control; and physics-constrained data-driven approaches.

Thermal models: Several studies have looked at thermal model development, and these can be broadly classified into white-box, black-box, and grey-box models. **(1)** *White-box* models require a complete understanding of the system behaviour and are assumed to be deterministic. Tuning the models to account for uncertainties is non-trivial and time-consuming but also integrating with the building management system for real-time control can be challenging and expensive [8]. **(2)** *Black-box models* use statistical [24]

and neural-network [22] based techniques to capture the underlying non-linear relationships. These techniques require no domain knowledge but a lot of data and do not generalise well for unseen data in the training set [6]. Furthermore, because black-box models do not account for physics, they may give unphysical results leading to unreliable predictive control, as we show in Section 6.1.3.
(3) *Grey-box* models use prior domain knowledge and data to calibrate system dynamics [10]. One widely used grey-box model is the resistance-capacitance (RC) representation [16]. Depending on the complexity, RC forms a system of coupled differential equations that are challenging to solve for real-time optimal control problems [9].
Model-based and model-free : MPC is a commonly used technique to improve building's energy savings and comfort. Cost-saving of 20% was demonstrated in real-world case studies over a naive rule-based control [35]. Black-box-based optimal temperature was considered in [20]. Grey-box based RC model with deterministic and stochastic exogenous variables was used for optimal control in [11] and [33], respectively. The efficacy of MPC is sensitive to model errors, as we show in Section 6.1.1. Recently, model-free agent-based approaches such as deep reinforcement learning are gaining importance due to the challenges involved in developing accurate models for MPC. The agent interacts directly with the environment and, over time, learns an approximate control policy [37]. However, such agent-based methods would require significant exploration/interaction with real-world systems, making them impractical due to various operational constraints [29].
Physics-constrained ML: Methods encoding the prior domain knowledge as part of the neural network architecture were shown in [17, 27] and using custom loss function in [25]. A technique to directly solve the governing differential equation through automatic differentiation of neural networks was considered in [34]. A significant drawback of this method is that accounting for time-dependent exogenous and control inputs is non-trivial since the functional form of input features is required a priori. In [18], multi neural network architecture was proposed to account for exogenous input variables. However, [18] lacked a proper baseline for performance comparison. [30] proposed a recurrent neural network (RNN) architecture together with a physics-constrained formulation. While their approach was 50% more accurate than a purely data-driven method, the accuracy relied heavily on the availability of historical data, which may not always be available, particularly for wall temperature. A recent study, [21], looked at mitigating the issues with the original PINN formulation for control problems by considering the historical logs of the exogenous and control inputs as part of the input feature space and using an encoder-decoder neural architecture. However, [21] did not study the efficacy of the controller using the PINN model. A comprehensive review of informed machine learning can be found in [36].

We propose *PACMAN* - a physics-aware optimal HVAC control manager that complements existing works. We extend the idea of a time resetting strategy and a zero-order hold assumption, first proposed in [31], to a more complex system of coupled ODEs involving both a time-varying ambient temperature and a time-varying control input. In the time resetting strategy, we split the entire time-widow into multiple time slots, each of length τ; and we reset the time to zero at the start of each time slot. Next, we apply the zero-order hold assumption where the ambient temperature and the control input are constant within each time slot. We refer to Section 4.1 for more details. We use the standard lumped model representation for state evolution, and unlike previous studies on PINN, we also account for humidity. Although less important than T_a and T_w, high error in W_a can lead to increase in discomfort hours, as we will show in Section 6.1.1. We also study the sensitivity of PINN model errors and training runtime to the time reset length. Finally, we evaluate the efficacy of MPC with PINN.

Table 1: Notation used.

Symbol	Meaning (units)
Thermal model and optimisation	
t, δt	Time instant and control time step (s)
$\mathcal{H}$	Prediction horizon (s)
T, W	Temperature (°C) and Humidity ratio (gkg^{-1})
MRT	Mean Radiant Temperature (°C)
$\mathbf{T}_{SP}$	Temperature setpoint vector (°C)
C	Thermal capacitance ($J{\cdot}K^{-1}$)
C_P	Specific heat capacity of air ($J{\cdot}kg^{-1}{\cdot}K^{-1}$)
ρ	Air density ($kg{\cdot}m^{-3}$)
V	Room volume (m^3)
E_{HVAC}	Energy consumed by HVAC (W·s)
$\dot{Q}_i, \dot{Q}_L$	Building internal and total cooling load (W)
h	Specific enthalpy ($J{\cdot}kg^{-1}$)
R_i, R_o	Indoor and outdoor thermal resistance ($W^{-1}{\cdot}K$)
$\dot{m}$	Mass flowrate of AHU ($kg{\cdot}s^{-1}$)
$\dot{m}_g$	Building internal moisture generation ($kg{\cdot}s^{-1}$)
$\kappa_p, \kappa_i, \kappa_d$	PID gain constants
e	PID error (°C)
PPD	Predicted Percentage Dissatisfied (%)
v_a	Air speed ($m{\cdot}s^{-1}$)
MET	Metabolic Rate (met)
Clo	Clothing insulation (Clo)
Subscripts and Superscripts	
a, w, ∞, sa	Air, wall, outdoor ambient, and supply air
$\hat{x}$	Approximation of x
t	At time instant t
0	At time instant 0
PINN	
$\mathcal{M}$	Neural network model
τ	Time reset length (s)
θ	Parameters/weights of neural network
α	Learning rate
$\mathcal{L}$	Loss function
$\mathcal{R}$	Residual of Equations 3–5
N_p	# collocation points for ODE residual losses
N_0	# points for initial value losses
EPOCHS	# of training epochs

3 PROBLEM FORMULATION

The main aim of building HVAC equipment is to keep the user's comfort within acceptable limits. The standard metric used to evaluate the user thermal comfort is Fanger's Predicted Percentage Dissatisfied (*PPD*) [19], and $PPD \leq 10\%$ is considered as an acceptable thermal condition [1]. The three important thermal parameters that determine *PPD* are the indoor air temperature T_a, mean radiant temperature MRT, and humidity ratio W_a. The other factors, such

as v_a, MET, and Clo, are usually assumed to be constants. MRT is determined by T_w [3]. While many combinations of T_a, T_w, and W_a may lead to $PPD \leq 10\%$ criteria, they may lead to different energy consumptions. We are interested in a combination that minimises energy. The control knob typically available in the building is the temperature setpoint T_{SP}. Hence, there is a scope to dynamically adjust T_{SP} so as to provide just enough comfort at minimum energy. Notations are summarized in Table 1.

3.1 Optimal control problem

We formally define the optimal HVAC control problem as:

$$\min_{\mathbf{T_{SP}}} \quad \sum_{t}^{t+\mathcal{H}} E_{\text{HVAC}} \tag{1}$$

Here $\mathcal{H}$ is the optimisation horizon; E_{HVAC}, the HVAC energy consumption; and $\mathbf{T_{SP}}$, the temperature setpoint vector over the optimisation horizon. $E_{\text{HVAC}} = \dot{Q}_L/\text{COP}$, where $\dot{Q}_L$ is the cooling load and COP the coefficient of performance of the HVAC chiller. $\dot{Q}_L = \dot{m} \cdot (h_a - h_{sa})$ where h_a and h_{sa} are the specific enthalpies of the room air and supply air, respectively. The specific enthalpy h is a function of air temperature and humidity [2]. We consider compressor energy consumption as the objective function as it accounts for 80% of the total HVAC energy [15].

The constraints to the optimisation problem are as follows:

(1) **Thermal comfort satisfaction [1]:**

$$all(\mathbf{PPD}[t : t + \mathcal{H}]) \leq 10\%, \tag{2}$$

where $\mathbf{PPD}[t : t + \mathcal{H}]$ are the PPD values over the horizon.

(2) **Environment/plant model:** The evolutions of air temperature, humidity ratio, and wall temperature are governed by,

$$\frac{dT_a}{dt} = \frac{\dot{Q}_i}{C_a} + \frac{1}{C_a \cdot R_i}(T_w - T_a) + \frac{\dot{m}C_P}{C_a}(T_{\text{sa}} - T_a), \tag{3}$$

$$\frac{dW_a}{dt} = \frac{\dot{m}_g}{\rho V} + \frac{\dot{m}}{\rho V}(W_{\text{sa}} - W_a), \tag{4}$$

$$\frac{dT_w}{dt} = \frac{1}{C_w \cdot R_i}(T_a - T_w) + \frac{1}{C_w \cdot R_o}(T_\infty - T_w). \tag{5}$$

For simplicity, we omit the terms related to outdoor air infiltration and solar radiation. The internal heat load and moisture generation depend on the room occupancy, electrical fixtures, plants in the room, etc. We assume the internal heat load to be constant in this work, corresponding to a constant room occupancy. Note that these assumptions are not limitations of *PACMAN* and can be relaxed in a more general setting.

(3) **AHU mass flow rate evolution model:** The evolution of AHU mass flow rate ($\dot{m}$) is governed by the HVAC PID control logic and is of the form,

$$\dot{m}^t = \dot{m}^{t-\delta t} + \kappa_p \cdot e^t + \kappa_i \cdot \sum e^t \delta t + \kappa_d \cdot \frac{e^t - e^{t-\delta t}}{\delta t}, \tag{6}$$

where the error $e^t = T_a^t - T_{SP}^t$.

3.2 Optimal control framework

The closed-loop control is achieved by solving the optimisation problem together with the constraints (Equations 1–6) with a receding horizon technique at every control step using a prediction model of the system. A typical model-based controller interacts with the real-world environment at every control time step, as shown in Figure 1. The controller consists of a thermal model together with an optimiser. The output of the controller is the optimal temperature setpoints for the entire prediction horizon. A building management system (BMS) modulates the AHU fan by translating the setpoint of the very next step to mass flow rate through a PID control logic. The environment moves to a new state, and BMS monitors state variables. At the next control step, the new state variables are given as inputs to the model-based controller and the closed-loop control repeats. *PACMAN* uses a PINN-based thermal model to predict the future states.

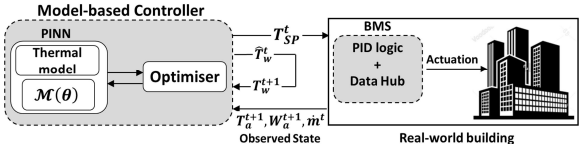


Figure 1: Optimal control framework.

3.3 PINN background

First proposed by [26] and recently made popular by [34], PINNs approximate the solution of the governing physics-based ordinary differential equations (Equations 3 – 5) by adding the ODE residuals to the loss function of a neural network. This approach is useful, particularly when labeled data is limited or unavailable (self-supervising mode). A schematic of the original PINN architecture [34] is shown as a computation graph in Figure 2. We explain the concept of PINN using the following initial-value problem: $\dot{\mathbf{x}}^t = \psi\left(\mathbf{x}^t, \mathbf{u}^t\right)$ given $\mathbf{x(0)} = \mathbf{x^0}$, where $t \in \mathbb{T}$, $\dot{\mathbf{x}}^t$ is the rate of change of $\mathbf{x}^t$, $\mathbf{x^0} \in \mathbb{X}$ (the system state), and $\mathbf{u} \in \mathbb{U}$ (set of control actions). Given the function $\mathbf{u}^t$ and the initial condition $\mathbf{x^0}$, the neural network of PINN takes t as the input. PINN approximates $\mathbf{x^t}$ as $\hat{\mathbf{x}}^\mathbf{t}(\theta)$, where θ is the vector of network weights. The physics-informed intelligence comes from the automatic differentiation of $\hat{\mathbf{x}}^\mathbf{t}$ with respect to the independent input variables (in this case, t). The neural network loss $\mathcal{L}(\theta)$ is the summation of labeled data loss, physics loss (residual of ODE), and initial condition loss, as shown in Figure 2. Interested readers may refer to [34] for more details. In this paper, we only consider the self-supervising mode of the PINN and do not consider any labeled data.

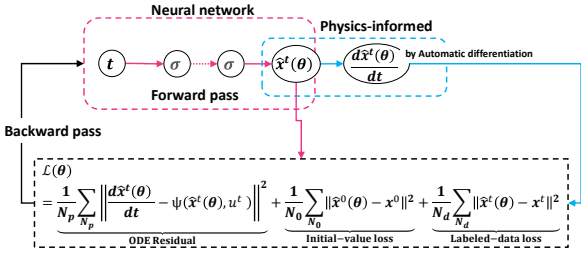


Figure 2: Original PINN computation graph.

3.4 Limitation of original PINN model

Consider a room exposed to a time-varying ambient temperature $T_\infty(t)$ and no cooling control with the following approximate governing equation for the room temperature: $\frac{dT_a}{dt} = K(T_\infty - T_a)$. Even

though $\frac{dT_a}{dt}$ at a point $t{=}t'$ depends upon only the current $T_\infty(t{=}t')$, the solution $T_a(t{=}t')$ at time t' would depend upon *the entire path* of $T_\infty(t)$ over all points t in $[0, t']$ and not just $T_\infty(t{=}t')$. In other words, to correctly solve the system, one needs to know the *function* $T_\infty(t)$ and not the scalar $T_\infty(t = t')$. Because T_a depends upon a time-varying function T_∞, its total derivative can be written as follows:

$$\frac{dT_a}{dt} = \frac{\partial T}{\partial T_\infty} \times \frac{dT_\infty}{dt} + \frac{\partial T_a}{\partial t}$$

In the original PINN formulation, T_∞ is assumed constant with time, so we have $\frac{dT_\infty}{dt}$=0 and so $\frac{dT_a}{dt} = \frac{\partial T_a}{\partial t}$. Note that the auto-differentiation would compute $\frac{\partial T_a(\theta)}{\partial t}$ and so, the *residual* error between $\frac{\partial T_a(\theta)}{\partial t}$ from the neural network; and $\frac{\partial T_a}{\partial t}$ from the governing equation can be minimised as function of θ. For HVAC control, we need to handle a non-zero $\frac{dT_\infty}{dt}$. If we do not do this, the output could be quite different than reality as illustrated in Figure 3 that shows: 1) the driving time-varying ambient temperature; 2) the correct time-varying room temperature; and 3) the incorrect PINN prediction that does not account for time-variation in the ambient temperature.

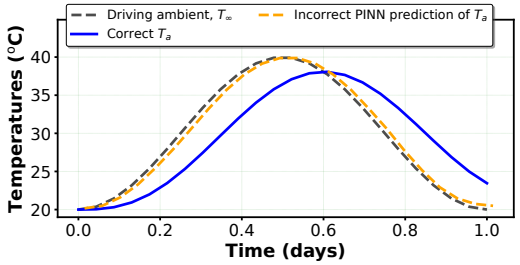


Figure 3: Original PINN model does not handle a time-varying $T_\infty(t)$.

Naive approaches: One approach to handling a time-varying $T_\infty(t)$ would be to specify the function $T_\infty(t)$ as a discretized vector input $\mathbf{T}_\infty$ to the PINN for discrete points t. However, this significantly increases the dimensionality of the neural network input space; and is not scalable with the number of points t. Another approach would be specify $T_\infty(t)$ as a closed analytical form function of t as part of the ODE residual. This may not be feasible in general. Even if it were possible for time-varying $T_\infty(t)$, for model-predictive control, time-varying control inputs (from the optimiser) also need to be handled in exactly the same way as time-varying $T_\infty(t)$ for the PINN to be adapted for HVAC control. Clearly, the time-varying optimal control cannot be specified as a closed analytical form to the PINN as input, because the control would itself be decided through optimisation on the PINNs output.

4 SOLUTION STRATEGY

4.1 Overcoming original PINN limitations

Figure 4 shows the schematic of our approach. We divide the time horizon into multiple time slots of equal length τ (time reset). Within each τ, we assume that the external inputs and the initial values are invariant (zero-order hold assumption). This eliminates the need to know a priori the exact functional variation of the external inputs with time. The original PINN, trained only over the period τ, can now be used with constant initial values and constant external inputs as additional input features. This is because the assumption of constant external inputs within τ results in the partial derivative of the thermal parameters estimated by automatic differentiation to be a correct estimate of the total derivative. Another advantage of the simplification is the reduction in sample space that is computationally tractable. Our PINN modification is a natural candidate for MPC with a receding horizon technique where the control input is constant over the control time step. An advantage of the PINN formulation over the numerical integrators available to solve the ODE (such as RK4) is that we can directly predict the output for any time $t' \in [k\tau, (k+1)\tau]$ with a forward pass without the need to predict at intermediate points within $[k\tau, t']$. As τ is reduced, the prediction given by the PINN approaches the true solution for the thermal parameters given by solving the ODEs for the actual functional inputs.

As the PINN is used with MPC in a receding horizon technique, the predicted outputs at t's such as $\hat{x}^{t+1}, ..., \hat{x}^{t+m}$, which become inputs at their respective following control step, are replaced with the actual observed values from the environment. This reduces error accumulation across control time steps. Only those parameters that are sensed in the BMS (such as T_a, W_a, T_∞, and $\dot{m}$) are directly fed back to the prediction model. T_w is usually not measured and thus needs to be approximated. For this purpose, we use $\hat{T}_w$ predicted by the model at the current time step t as a soft-sensed input T_w^{t+1} to the next control step, as shown in Figure 1.

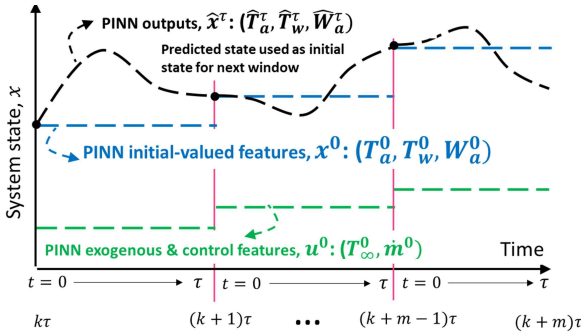


Figure 4: Schematic of our PINN implementation.

4.2 PINN thermal model training

Based on the time resetting and zero-order hold assumption, the following six input features are chosen for our PINN formulation:

(1) Time instant $t \in [0, \tau]$
(2) T_∞^t assumed constant over $[k\tau, (k+1)\tau]$ $\forall k$
(3) $\dot{m}^t$ assumed constant over $[k\tau, (k+1)\tau]$ $\forall k$
(4) The initial value T_a^0 at the beginning of each time slot
(5) The initial value T_w^0 at the beginning of each time slot
(6) The initial value W_a^0 at the beginning of each time slot

The three output variables of the PINN are $\hat{T}_a^t$, $\hat{T}_w^t$, and $\hat{W}_a^t$, for any time t given as the input.

We present the outline of our self-supervised (without labeled data) PINN training in Algorithm 1. First, we initialize the hyper-parameters: learning rate α, EPOCHS, N_p, N_0, and τ in Line 1 to Line 5. Then we sample N_p collocation examples of (t, T_∞, $\dot{m}$, T_a^0, T_w^0, W_a^0) and N_0 initial-value examples of (0, T_∞, $\dot{m}$, T_a^0, T_w^0,

Algorithm 1: PINN training.

Hyperparameters:

1 α // Learning rate of neural network
2 EPOCHS // Number of iterations

Inputs:

3 N_p // Number of collocation points for physics loss term
4 N_0 // Number of points for initial-value loss term
5 τ // Time reset length
6 $\mathcal{X}_p \leftarrow$ Sample$[(t, T_\infty, \dot{m}, T_a^0, T_w^0, W_a^0), N_p]$ // Draw samples for physics loss
7 $\mathcal{X}_0 \leftarrow$ Sample$[(0, T_\infty, \dot{m}, T_a^0, T_w^0, W_a^0), N_0]$ // Draw samples for initial value loss
8 $\mathcal{Y}_0 \leftarrow \mathcal{X}_0[:, (T_a^0, T_w^0, W_a^0)]$
9 Randomly initialize model $\mathcal{M}$ parameters θ
10 **for** $e = 1$ to EPOCHS **do**
11 $(\hat{T}_a(\theta), \hat{T}_w(\theta), \hat{W}_a(\theta)) \leftarrow$ Evaluate $\mathcal{M}(\mathcal{X}_p)$
12 $\hat{\mathcal{Y}}_0 \leftarrow$ Evaluate $\mathcal{M}(\mathcal{X}_0)$ // Forward pass
13 $(\frac{d\hat{T}_a(\theta)}{dt}, \frac{d\hat{T}_w(\theta)}{dt}, \frac{d\hat{W}_a(\theta)}{dt}) \leftarrow$ AutoDiff $(\hat{T}_a, \hat{T}_w, \hat{W}_a)$ // Automatic differentiation
14 $\mathcal{R}_{T_a}, \mathcal{R}_{W_a}, \mathcal{R}_{T_w} \leftarrow$ Obtain residuals using Lines 11 and 13 for Equations 3, 4, 5 as functions of θ
15 $\mathcal{L}_{\text{phy}} \leftarrow \frac{1}{N_p}||\mathcal{R}_{T_a}||^2 + \frac{1}{N_p}||\mathcal{R}_{T_w}||^2 + \frac{1}{N_p}||\mathcal{R}_{W_a}||^2$
16 $\mathcal{L}_{\text{init}} \leftarrow \frac{1}{N_0}||\hat{\mathcal{Y}}_0 - \mathcal{Y}_0||^2$
17 $\mathcal{L}(\theta) \leftarrow \mathcal{L}_{\text{phy}} + \mathcal{L}_{\text{init}}$
18 Evaluate gradients, $\nabla_\theta \mathcal{L}(\theta)$
19 Update parameters, $\theta \leftarrow \theta - \alpha \cdot \nabla_\theta \mathcal{L}(\theta)$
20 **end for**
21 **Return** $\mathcal{M}(\theta)$

W_a^0) in Lines 6 – 8 for evaluating physics and initial-value losses, respectively. We use the Latin Hypercube Sampling (LHS) technique, which has better sampling coverage than standard random sampling [38], to sample the input features from their respective ranges. We initialize the network parameters/weights θ in Line 9. In the main iteration loop (Line 10), in Lines 11 and 12, we do a forward pass of the neural network to get an estimate of the thermal model parameters at $\mathcal{X}_P$ and $\mathcal{X}_0$. We evaluate the derivatives of the estimates of temperatures and humidity ratio with respect to t at Line 13 using automatic differentiation (as functions of θ). Next, we plug in the model estimates and their gradients along with the external and control inputs in Equations 3–5 to calculate the ODE residuals (difference between the left hand and right hand sides of the ODE); this is shown in Line 14. The physics loss $\mathcal{L}_{\text{phy}}$ is calculated as the sum of the individual mean squared ODE residuals (Line 15). The initial value loss $\mathcal{L}_{\text{init}}$ is calculated as the mean squared error between $\hat{\mathcal{Y}}_0$ (estimated in Line 12) and $\mathcal{Y}_0$ (sampled in Line 8). The total loss function $\mathcal{L}(\theta)$ is the sum of the physics loss and the initial-value loss. The gradients of the loss function with respect to the network parameters are evaluated in Line 18. Finally, the network parameters are updated using the standard gradient descent algorithm in Line 19. Though not shown here, loss contributions from each thermal parameter could be weighted differently in order to improve the performance of the trained PINN.

4.3 PINN in control

Algorithm 1 returns the trained PINN model $\mathcal{M}(\theta)$, which is used as the thermal model in the optimal control framework, as shown in Figure 1 and detailed in Section 3.2.

5 EXPERIMENTAL SETUP

We design experiments to evaluate: **(1)** the accuracy of PINN thermal model in isolation; and **(2)** the efficiency of PINN when used in *PACMAN* control. For the former, we use LSTM as baseline. For the latter, we use different controls as baselines.

5.1 Thermal model evaluation

Ground truth: We simulate the environment using a numerical integration of the thermal model (Equations 3–5). Specifically, we use the fourth-order Runge-Kutta method (RK4) [5], which is an accurate and widely used method for solving initial-value problems governed by first-order ODEs. The simulation time step is 15 minutes. The constants used in the thermal model (Equations 3–5) are summarised in Table 2. Note that $\dot{m}_g$ in Equation 4 varies dynamically since it is a function of the latent load (a constant) and latent heat of vaporisation (a function of T_a). We also simulate a PID control logic (Equation 6), the gain constants of which are tuned using the Ziegler-Nichols methods [12]. Furthermore, the PID control logic (Equation 6) usually operates within a temperature dead-band, that is, $\dot{m}$ is not adjusted if $T_a \in [T_{\text{SP}} \pm \text{dead-band}]$. We also implement an anti-windup logic where the integral error is not propagated if $T_a \in [T_{\text{SP}} \pm \text{dead-band}]$.

Table 2: Thermal model parameter list. All values are in SI units unless otherwise specified. Temperatures are in °C.

Parameter	Value	Parameter	Value
$\dot{Q}_i$	5000	C_a, C_w	50e6, 306e6
R_i, R_o	3.2e-5, 1.1e-5	ρ, C_P	1.2, 1005
$T_{\text{sa}}, W_{\text{sa}}$ (g·kg^{-1})	13, 8	Latent load	1250
$\dot{m}$	$\in [0., 3.]$	T_{SP}	$\in [15., 30.]$
$\kappa_p, \kappa_i, \kappa_d$	0.1, 0.0001, 0.	PID dead-band	±1

PINN specific details: Theoretically, $\tau \geq$ control time step. The time reset parameter τ is taken 15 minutes. Depending on the choice of τ, there will be a trade-off between computation speed and accuracy, as we show in Section 6.1.4. The input features, which are the time instant t, temperatures T_∞, T_a^0, T_w^0, humidity ratio W_a^0, and mass flow rate $\dot{m}$ are sampled from $[0, \tau]$, [14.5, 39.5] °C, [5., 16] g·kg^{-1}, and [0., 3.] kg·s^{-1}, respectively. The neural network loss $\mathcal{L}(\theta)$ is the sum of physics loss $\mathcal{L}_{\text{phy}}$ (ODE residuals) and initial value loss $\mathcal{L}_{\text{init}}$ (refer to Figure 2 and Algorithm 1). We have used *Tanh* activation function for all hidden layers as it works better than other activation functions such as *sigmoid* and *ReLU* for problems involving approximating non-linear governing ODEs/PDEs [23]. The number of iteration EPOCHS = $2M$.

LSTM specific details: The labeled temporal examples of T_∞, T_a, W_a, and $\dot{m}$ required for LSTM model are sampled from the environment observations. Note that, although T_w is usually not measured, to make a fair comparison with PINN, we also use T_w temporal examples as an input to LSTM. The sampling frequency is 15 minutes. We implement LSTM using the standard Tensorflow-Keras function

calls. Unlike PINN, LSTM uses only the mean-squared-error (MSE) between predicted and labeled values as a loss function.

Table 3 summarizes a few of the hyperparameters used in the training of both the PINN and LSTM models. We consider a typical annual ambient profile of a building in a tropical climatic region where $T_\infty \in [14.5, 39.5]^\circ$C.

Table 3: PINN and LSTM common hyperparameters.

Hyper-parameter	Value	Hyper-parameter	Value
Size of input layer	6	# hidden layers	2
# nodes per hidden layer	20	Size of output layer	3
Hidden layer activation	Tanh	Optimiser	Adam
Output layer activation	Linear	α	1e-4
Input normalisation	*Min-Max*, [-1, +1]		

Performance metric for thermal model evaluation: We evaluate the efficacy of thermal models using the average relative error between the predicted and the ground truth values for T_a, T_w, W_a.

5.2 Control evaluation

Recall that the decision variable for optimal control is the indoor air temperature setpoint vector $\mathbf{T}_{sp}$, which is used as a reference value in the PID control logic. In this work, we assume complete knowledge of the PID control logic. We consider the following methods to evaluate the control efficacy of *PACMAN* .

(1) **BL1 (As-is):** We assume a constant setpoint of $T_{SP} = 23 \pm 1^\circ$C throughout the HVAC operation. This is the most commonly observed control strategy implemented in buildings.
(2) **BL2 (Seasonal T_{SP}):** We consider a constant setpoint of $T_{SP} = 22 \pm 1^\circ$C during the summer months and $T_{SP} = 23 \pm 1^\circ$C during the rest of the year. This seasonal choice of setpoints is expected to increase occupants' comfort during the warmer months.
The following two baselines and *PACMAN* undertake model-based optimal setpoint determination in a receding horizon control framework as explained in Section 4. The control time step δt and the prediction horizon $\mathcal{H}$ have the same length of 15 minutes. These control techniques differ only in the model used for predicting the thermal parameters over the prediction horizon.
(3) **BL3 ('oracle' MPC)**: We assume complete knowledge of the system and external disturbances of the thermal parameters as seen by the environment. Though unrealistic, this baseline quantifies the best that can be achieved. The thermal model used is the ground truth RK4.
(4) **BL4 (MPC + LSTM)**: We use the LSTM-model (as described in Section 5.1) to predict the thermal parameters over the optimisation horizon. The PINN thermal model in Figure 1 is replaced by the LSTM model.
(5) ***PACMAN* :** We use PINN as thermal model in optimal control framework (Figure 1). The time reset length τ is the same length as the control step, that is, $\tau = 15$ minutes.

Performance metrics for control evaluation: We consider two metrics: **(1)** The annual HVAC energy consumed in MWh, **(2)** The percentage of HVAC operation time that the comfort is unmet, that is, $PPD > 10\%$. Lower energy and unmet comfort hours indicate a better control strategy. In the PPD calculation, we fix v_a, *MET*, and *Clo* at 0.1 m/s, 1.2 met, and 0.5 Clo, respectively.

6 RESULTS

6.1 Thermal model evaluation

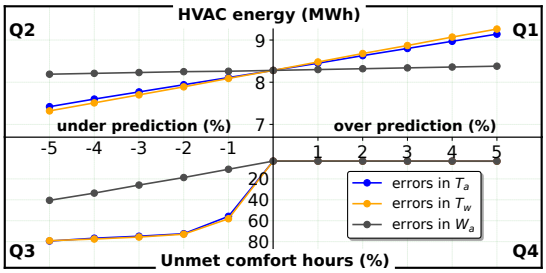
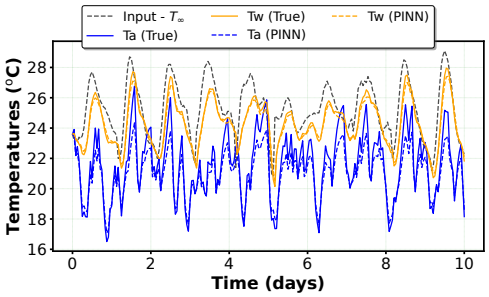


Figure 5: Sensitivity of optimal control solution to model errors.

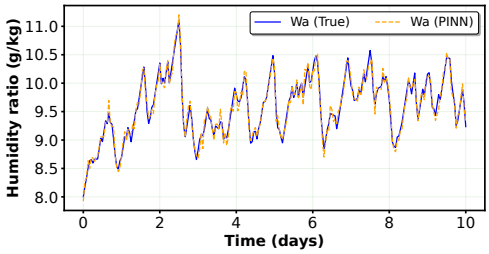
6.1.1 **Need for accuracy in thermal model.** Figure 5 shows the sensitivity of the optimal control solution to model errors. The X-axis represents the percentage of errors injected into the thermal model. The top Y-axis (Q1-Q2) is the annual HVAC energy consumed in MWh, and the bottom Y-axis (Q3-Q4) is the percentage unmet comfort hours. We consider the following model errors here: ±1, ±2, ±3, ±4, and ±5%. We use positive errors to evaluate the model's over-predictions and vice versa. Errors are introduced in one parameter at a time. Specifically, when an error is introduced in T_a, the model errors in T_w and W_a are kept at zero. We infer the following from Figure 5: **(1)** The optimal control solution (energy and unmet comfort hours) is more sensitive to errors in T_a and T_w compared with W_a. **(2)** Thermal parameters T_a and T_w are equally important. **(3)** The HVAC energy varies almost linearly with errors in T_a and T_w because the HVAC energy is modelled as a function of enthalpy, which is a linear function of T_a. Note that errors in T_w affect the enthalpy through T_a (T_a and T_w are connected via the governing equations). **(4)** An interesting trend is observed in the unmet comfort hours. When the model over-predicts thermal parameters, the unmet hours remain nearly unchanged. However, when the model under-predicts the parameters, particularly T_a and T_w, the unmet hours increase non-linearly. This behaviour is explained as follows. For cooling, a higher temperature setpoint leads to a lower HVAC energy and vice versa. The thermal comfort is bounded by $PMV \pm 0.5$, which is equivalent to $PPD \leq 10\%$. During over-prediction (under-prediction), the model picks T_{SP} closer to lower (higher) value of T_{SP}^{opt}. Because the optimiser is trying to minimise energy, it picks a higher T_{SP} among the acceptable values, and the chosen T_{SP} during over-prediction has more cushion to the thermal comfort boundaries compared with T_{SP} picked during under-prediction.

For model errors in the range of (-5% to +5%) in T_a and T_w, the percentage change in the energy from the 'oracle' model (zero errors) is -11% to +13% and the unmet comfort hours vary from 80% to 3%. For model errors in W_a in (-5% to +5%), the energy changes are marginal from -1.1% to +1.2% while the unmet hours vary from 40% to 3%. *These experiments demonstrate the need for an accurate thermal model.*

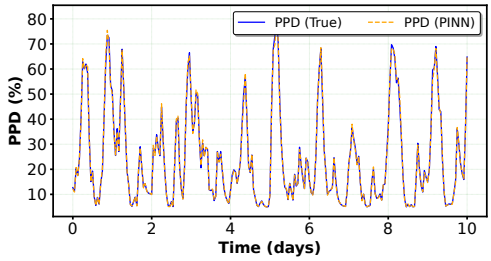
6.1.2 **Prediction accuracy of PINN vs. ground truth.** Figure 6 shows the thermal parameter profiles obtained by PINN and compares the solution with the ground truth. The X-axis represents the time in days, and the Y-axis represents the three primary thermal parameters, namely T_a, T_w (Figure 6a), W_a (Figure 6b), and a derived metric *PPD* (Figure 6c). We observe that PINN compares well with the ground truth data (the solid and dashed lines almost overlap) and gives 0.4%, 0.2%, and 0.7% average errors in T_a, T_w, and W_a over the annual profiles (we have zoomed to 10 days time window in Figure 6 to highlight the goodness of the predictions). Because PPD is a non-linear function of T_a, T_w, and W_a, a slightly higher error of 2.2% is noted in PPD.



(a) PINN temperature model vs true temperature



(b) PINN humidity model vs true humidity



(c) PPD predicted vs true PPD

Figure 6: Comparison of PINN solution with ground truth.

Figure 7 shows the reduction in ODE residuals for each of the three differential equations (ϕ_{T_a}, ϕ_{T_w}, and ϕ_{W_a}) along with the total loss ϕ_t. We observe a smooth decrease in the loss for all physical parameters of interest with training.

6.1.3 **PINN vs purely data-driven models.** The temperature and humidity ratio predictions with LSTM are shown in Figure 8. The temperature predictions are zoomed to 10 days, while the humidity ratio predictions are shown for the entire year to demonstrate unphysical solution. The LSTM model gives 5%, 3%, and 7% errors in T_a, T_w, and W_a, respectively, while an error of at most 1%

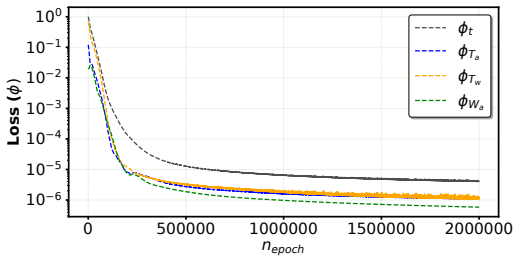
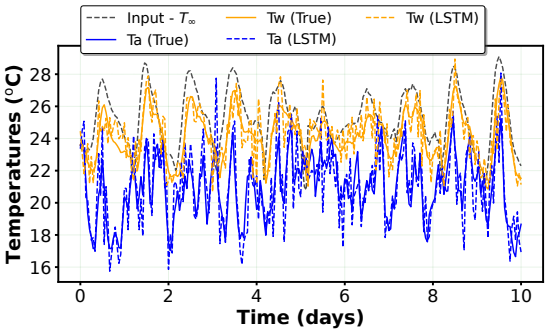
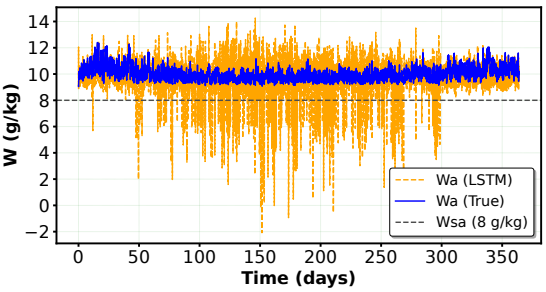


Figure 7: PINN convergence.

for T_a and T_w is needed to avoid severe discomfort. Furthermore, LSTM is a purely data-driven model and so may result in unphysical results, as shown in Figure 8b. The humidity ratio is observed to go below the supply value W_{sa} = 8 g/kg and is also seen to take negative values, which is unphysical. Because the humidity is added to the room by the occupants in our experiments, the resulting W_a cannot go below W_{sa} for the operating conditions we consider. The poor performance of LSTM was also noted for a building heating and cooling problem in [30]. *PINN has the required accuracy and does not violate any physical constraints on the parameters due to its underlying physics modelling.*



(a) LSTM temperature model vs true temperature



(b) LSTM humidity model vs true humidity

Figure 8: Comparison of LSTM solution with ground truth.

Generalizability: Any data-driven model would need to train over a wide range and a large number of sample values of the physical parameters to generalize well at unseen input values. However, since PINN is physics-constrained, it can learn the underlying physical process even with limited data and generalize better than a purely data-driven approach. We demonstrate generalizability by training both LSTM and PINN with summer months data, where the ambient temperature values are sampled from 28 – 40°C. Next, we use the trained models to predict the solution at out-of-distribution ambient

temperatures from 15 − 25°C. Figure 9 compares training and generalization (test) errors for LSTM and PINN. We observe that LSTM generalizes poorly over the out-of-distribution test dataset for all the thermal parameters, whereas *PINN generalizes well over the out-of-distribution test dataset* by learning the underlying physics. Although better than LSTM, PINN generalization errors on out-of-distribution samples are still higher than the desired errors (< 1%). The finding is consistent with [26], where PINN was shown to have higher generalization error when the samples deviated significantly from the samples used in training.

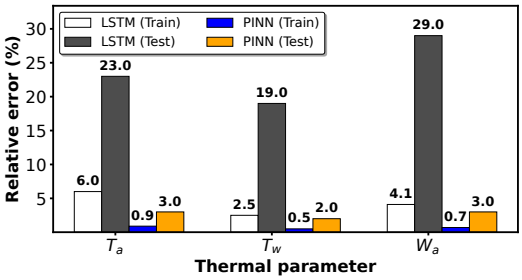


Figure 9: Generalization error: LSTM vs PINN.

6.1.4 **Sensitivity of PINN to τ.** Figure 10 studies the sensitivity of PINN to the time reset length τ. The X-axis represents τ. The primary Y-axis represents the PINN's CPU runtime during training and the secondary Y-axis is the thermal model error. We considered τ = 15, 30, 45, and 60 minutes. As expected, the runtime drops

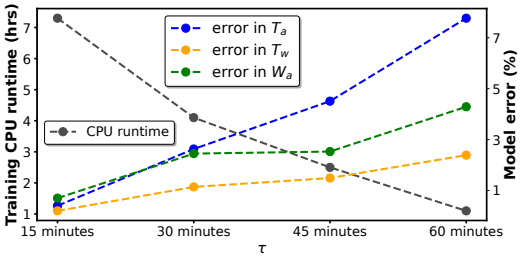


Figure 10: Sensitivity of CPU runtime and model errors to τ.

sharply from 7.3 hours at τ = 15 minutes to 1.2 hours at τ = 60 minutes. Because the inputs to the model are held constant over the time reset window, we expect that the smaller the τ is, the lower is the model error. In Figure 10, we observe that the errors for all the thermal parameters progressively increase with increasing τ, as expected.

6.2 Control evaluation of *PACMAN*

Figure 11 compares *PACMAN* 's control performance with that of the baselines. The numerals on top of the bar chart for BL1 indicate the raw energy consumption in MWh and the percentage of unmet comfort hours. For BL2 through *PACMAN*, the numerals indicated are percentage relative changes for energy and percentage point changes for the percentage unmet comfort hours compared to BL1. The as-is control (BL1) uses a fixed setpoint of $T_{SP} = 23 \pm 1$°C throughout the HVAC operation and consumes approximately 10.3 MWh of annual energy with 29% annual unmet comfort hours (that is, for 29% of the year the PPD > 10%). BL2 uses a seasonal

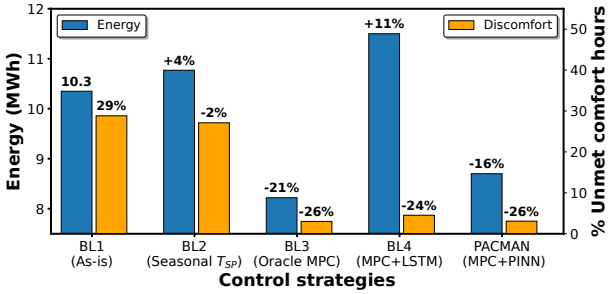


Figure 11: Comparison of *PACMAN* 's performance with baselines. The numerals over the orange bars (*from BL2 to PACMAN*) are % point changes for the percentage unmet comfort hours over BL1.

setpoint strategy, that is, $T_{SP} = 22 \pm 1$°C during summer months and $T_{SP} = 23 \pm 1$°C during the rest of the year. BL2 reduces the percentage unmet comfort hours by a marginal 2% points while consuming 4% more energy over BL1 due to using a lower setpoint during the summer months. The 'oracle' MPC (BL3) limits the maximum possible energy reduction while reducing unmet comfort hours. BL3 results in 21% energy savings and reduces percentage unmet comfort hours by 26% points over BL1. In BL4, the thermal model of MPC is represented by LSTM. Recall that LSTM may result in unphysical results particularly for W_a prediction. For the sake of control with LSTM, we lower bound W_a to an acceptable range. While BL4 reduces the percentage unmet comfort hours by 24% points compared to BL1, it consumes 11% more energy. Finally, *PACMAN* outperforms all other baselines except BL3 (the 'oracle' controller) in reducing energy and unmet comfort hours. *PACMAN* reduces energy by 16% and percentage unmet comfort hours by 26% points over BL1. *PACMAN* 's unmet comfort hours are comparable to BL3, albeit by consuming 6% more energy than BL3. This shows that PINNs can help to improve control optimisation by improved modelling.

7 LIMITATIONS

We highlight some limitations of our approach, which can guide future research. **(1)** All the thermal parameters and exogenous inputs are assumed to be deterministic. However, any real-world system will be affected by noise, and the effect of this noise on the controller's performance needs to be investigated. **(2)** We have assumed complete knowledge of the constants (such as thermal resistance and capacitance) in the governing thermal equations. In reality, these may be unknown and need to be inferred, for which, again, some labeled data are required. PINNs have been shown to work in this inverse problem of calibrating equation parameters using data [34]. We have also assumed a complete knowledge of the HVAC PID controller that maps the setpoint to the HVAC supply mass flow rate. In reality, the controller logic may be proprietary to the manufacturer and may need to be approximated. **(3)** As this paper focuses on the PINN model in a control framework, the prediction horizon is kept equal to the control time step for simplicity of optimisation. However, the horizon is usually longer than the control time step, making the optimisation more challenging. More powerful optimal setpoint determination methods based on Reinforcement Learning can be explored. Such techniques have been

shown to work in real-life stochastic settings too. **(4)** The lumped thermal model considered in this paper treats the thermal parameters to be homogenous in the entire room. However, in practice, the parameters may be non-uniform and this non-uniformity must be accounted for better thermal comfort quantification. Besides, the true advantage of PINN comes from solving the governing partial differential equations such as the Navier-Stokes equations to model the spatial variations in parameters of interest. We propose to look at this problem and the computational benefit that can be achieved in real-time control as part of our future work.

8 CONCLUSION

We proposed *PACMAN* to solve the optimal HVAC control problem in buildings that minimises energy while meeting the user comfort. We used a physics-aware neural network framework to model the thermal dynamics. We reformulated the original PINN to make it suitable for optimal control problems. Specifically, we used a time resetting strategy and a zero-order hold assumption that helped negate the functional specification requirement for the exogenous and control inputs and reduce the sampling space. We showed that physics-aware architecture helps better predict the thermal parameters than a purely data-driven model (LSTM), and the solution obeys physical laws. PINNs resulted in an error < 1% for all thermal parameters. Our evaluation showed the need for high thermal model accuracy in MPC. Our approach performed better than most baselines, next only to 'oracle' MPC, and saved 16% energy and reduced percentage unmet comfort hours by 26% points compared with the as-is control strategy. Finally, we showed that PINNs are sensitive to the time reset length; increasing the time window degraded the solution because the approximation of holding the input variables constant over a longer time window introduces higher model errors. Our future work includes evaluating the utility of PINNs in modelling spatial variations in thermal parameters and real-time control benefits over a full computational fluid dynamics-based approach.

REFERENCES

[1] ASHRAE standard 55: Thermal environmental conditions for human occupancy. ASHRAE, Atlanta, 2010.

[2] Psychrometrics (equations). Built Environment Research,Civil, Architectural and Environmental Engineering Illinois Institute of Technology, US, 2017.

[3] *ISO 7726: 2001 Ergonomics the thermal environment. Instruments for measuring physical quantities.* 2001.

[4] *Buildings Energy Data Book.* 2011.

[5] M. L. Abell and J. P. Braselton. 2 - first-order ordinary differential equations. In *Differential Equations with Mathematica (Fourth Edition)*, pages 45–131. Academic Press, Oxford, fourth edition edition, 2016.

[6] Z. Afroz, G. Shafiullah, T. Urmee, and G. Higgins. Modeling techniques used in building hvac control systems: A review. *Renewable and Sustainable Energy Reviews*, 83:64–84, 2018.

[7] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng. Occupancy-driven energy management for smart building automation. In *Proceedings of BuildSys*, pages 1–6. ACM, 2010.

[8] K. Arendt, M. Jradi, H. Shaker, and C. Veje. Comparative analysis of white-, gray- and black-box models for thermal simulation of indoor environment: Teaching building case study. In *Proceedings of IBPSA-USA*, pages 173–180, 2018.

[9] J. Arroyo, F. Spiessens, and L. Helsen. Identification of multi-zone grey-box building models for use in model predictive control. *Journal of Building Performance Simulation*, 13(4):472–486, 2020.

[10] P. Bacher and H. Madsen. Identifying suitable models for the heat dynamics of buildings. *Energy and Buildings*, 43(7):1511–1522, 2011.

[11] A. Beltran and A. E. Cerpa. Optimal HVAC building control with occupancy prediction. In *Proceedings. of BuildSys*, pages 168–171. ACM, 2014.

[12] V. Bobál, J. Macháček, and R. Prokop. Tuning of digital pid controllers based on ziegler - nichols method. *IFAC Proceedings Volumes*, 30(21):145–150, 1997.

[13] J. Brooks, S. Goyal, R. Subramany, Y. Lin, T. Middelkoop, L. Arpan, L. Carloni, and P. Barooah. An experimental investigation of occupancy-based energy-efficient control of commercial building indoor climate. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 5680–5685. IEEE, 2014.

[14] T. Chaudhuri, Y. C. Soh, S. Bose, L. Xie, and H. Li. On assuming mean radiant temperature equal to air temperature during pmv-based thermal comfort study in air-conditioned buildings. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 7065–7070, 2016.

[15] E. Cheng. Dynamic chiller plant optimization. ATAL Building Services Engineering Ltd, 2018.

[16] R. D. Coninck, F. Magnusson, J. Åkesson, and L. Helsen. Toolbox for development and validation of grey-box building models for forecasting and control. *Journal of Building Performance Simulation*, 9(3):288–303, 2016.

[17] F. Djeumou, C. Neary, E. Goubault, S. Putot, and U. Topcu. Neural networks with physics-informed architectures and constraints for dynamical systems modeling. *arXiv*, 10.48550/ARXIV.2109.06407, 2021.

[18] J. Drgoňa, A. R. Tuor, V. Chandan, and D. L. Vrabie. Physics-constrained deep learning of multi-zone building thermal dynamics. *Energy and Buildings*, 243:110992, 2021.

[19] P. Fanger. *Thermal Comfort: Analysis and Applications in Environmental Engineering.* McGraw Hill, 1970.

[20] P. Ferreira, A. Ruano, S. Silva, and E. Conceição. Neural networks based predictive control for thermal comfort and energy savings in public buildings. *Energy and Buildings*, 55:238 – 251, 2012.

[21] G. Gokhale, B. Claessens, and C. Develder. Physics informed neural networks for control oriented thermal modeling of buildings. *Applied Energy*, 314, 2022.

[22] H. Huang, L. Chen, and E. Hu. A neural network-based multi-zone modelling approach for predictive control system design in commercial buildings. *Energy and Buildings*, 97:86–97, 2015.

[23] A. Jagtap, K. Kawaguchi, and G. Karniadakis. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136, 2019.

[24] A. Jain, F. Smarra, and R. Mangharam. Data predictive control using regression trees and ensemble learning. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 4446–4451, 2017.

[25] A. Karpatne, W. Watkins, J. S. Read, and V. Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *ArXiv*, abs/1710.11431, 2017.

[26] I. Lagaris, A. Likas, and D. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998.

[27] M. Lutter, C. Ritter, and J. Peters. Deep lagrangian networks: Using physics as model prior for deep learning. In *7th International Conference on Learning Representations (ICLR)*. ICLR, 2019.

[28] S. Nagarathinam, H. Doddi, A. Vasan, V. Sarangan, P. Venkata Ramakrishna, and A. Sivasubramaniam. Energy efficient thermal comfort in open-plan office buildings. *Energy and Buildings*, 139:476–486, 2017.

[29] S. Nagarathinam, V. Menon, A. Vasan, and A. Sivasubramaniam. MARCO - multi-agent reinforcement learning based control of building HVAC systems. In *Proceedings 11th ACM e-Energy*, page 57–67, 2020.

[30] L. D. Natale, B. Svetozarevic, P. Heer, and C. N. Jones. Physically consistent neural networks for building thermal modeling: theory and analysis. *CoRR*, abs/2112.03212, 2021.

[31] J. Nicodemus, J. Kneifl, J. Fehr, and B. Unger. Physics-informed neural networks-based model predictive control for multi-link manipulators. *arXiv*, 10.48550/ARXIV.2109.10793, 2021.

[32] B. Olesen and K. Parsons. Introduction to thermal comfort standards and to the proposed new version of en iso 7730. *Energy and Buildings*, 34(6):537–548, 2002.

[33] Parisio, Alessandra and Varagnolo, Damiano and Risberg, Daniel and Pattarello, Giorgio and Molinari, Marco and Johansson, Karl H. Randomized model predictive control for HVAC systems. In *Proceedings of BuildSys*, pages 1–8. ACM, 2013.

[34] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[35] D. Sturzenegger, D. Gyalistras, M. Morari, and R. S. Smith. Model predictive climate control of a swiss office building: Implementation, results, and cost–benefit analysis. *IEEE Transactions on Control Systems Technology*, 24(1):1–12, 2016.

[36] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, M. Walczak, J. Pfrommer, A. Pick, R. Ramamurthy, J. Garcke, C. Bauckhage, and J. Schuecker. Informed machine learning - a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[37] T. Wei, Y. Wang, and Q. Zhu. Deep reinforcement learning for building HVAC control. In *Proceedings 54th ACM DAC*, 2017.

[38] H. Xiao, W. Pei, W. Deng, L. Kong, H. Sun, and C. Tang. A comparative study of deep neural network and meta-model techniques in behavior learning of microgrids. *IEEE Access*, 8:30104–30118, 2020.