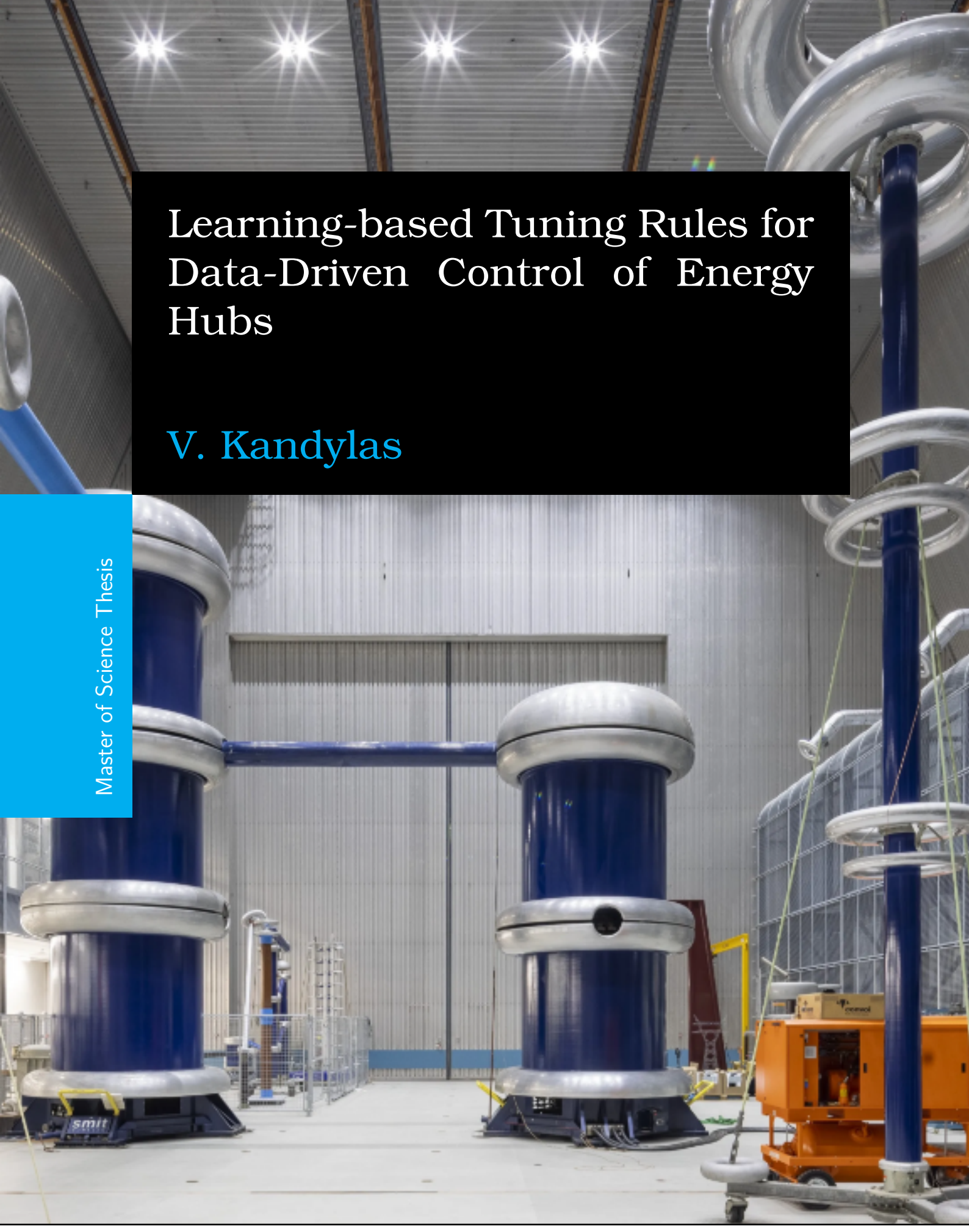


Learning-based Tuning Rules for Data-Driven Control of Energy Hubs

V. Kandylas

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Learning-based Tuning Rules for Data-Driven Control of Energy Hubs

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

V. Kandylas

August 21, 2026

Faculty of Mechanical Engineering (ME) · Delft University of Technology

Abstract

Buildings account for a large share of global energy consumption, and coordinating their heating, storage, and energy conversion components effectively requires a controller that can react to changing occupancy, weather, and pricing conditions. Model Predictive Control has been widely used for this purpose, but its performance depends on an explicit dynamical model of the building, which is costly to obtain and must be rebuilt whenever the building changes. Data-Enabled Predictive Control (DeePC) offers an alternative. A controller can be designed directly from measured input-output data, without identifying an explicit model first. This comes at a cost of its own. DeePC's performance depends strongly on a small number of regularization settings, and no systematic procedure currently exists for choosing them for a new building without repeating a full, building-specific tuning process from scratch.

This thesis develops and evaluates a learned mapping from a building's physical properties and the current season to a recommended regularization setting, using Gaussian process regression. Two Gaussian process models are trained separately for each of four seasons on a grid of regularization settings, one predicting closed-loop comfort violation and one predicting energy cost. Each combination is evaluated in closed-loop simulation across three training buildings differing in size, insulation, and thermal time constant. A recommended setting is selected by identifying the candidates with the lowest predicted violation within a fixed tolerance, then choosing the cheapest among them. The same procedure is applied to the fitted Gaussian process predictions and compared against the measured sweep data.

The fitted models recover each training building's own known optimal setting in most cases, and extend to buildings outside the training set through both interpolation and extrapolation in the physical feature space. No single regularization setting performs acceptably across every building and season at once. Applying a setting outside the building it was obtained for can more than double the resulting comfort violation, confirming the need for a building-specific mapping rather than a single fixed rule.

This thesis shows that a small set of training buildings is enough to learn regularization settings that transfer to buildings and seasons not seen during training, reducing the need for a separate manual tuning procedure for every new building. The results also identify the conditions under which this learned mapping is most reliable, providing a basis for extending the approach to a wider range of buildings and operating conditions in future work.

Table of Contents

Acknowledgements

I would like to thank my supervisor, Dr. Marta Zagorowska, for her guidance throughout this thesis. Her feedback shaped every chapter of this work, from the framing of the research questions to the final details of the results chapter, and her willingness to engage closely with the technical details, down to individual figures and citations, made this a far stronger thesis than it would otherwise have been.

Delft, University of Technology V. Kandylas
August 21, 2026

Chapter 1

Introduction

Buildings account for between 20 and 40% of total final energy consumption in developed countries, exceeding both the industrial and transport sectors [1, 2]. A large share of this consumption is attributed to Heating, Ventilation, and Air-Conditioning (HVAC) systems, which account for roughly half of all energy used in buildings [1]. As buildings increasingly integrate heat pumps, electrical storage, and renewable generation, the control problem becomes more complex and simple strategies are no longer sufficient [3, 4]. The design of optimization-based control methods for this growing class of systems has therefore become a central challenge in building energy management. One of the key obstacles to deploying such controllers in practice, however, is not the design of the controller itself but the selection of its tunable parameters, which must be chosen separately for each building and each operating condition [4]. This thesis addresses that parameter selection problem.

This chapter introduces the controller tuning problem addressed in this thesis. Section 1-1 describes building energy hubs and the control challenges they present, introduces data-driven control as a response to the limitations of model-based approaches, and explains why the resulting hyperparameter tuning problem is the central difficulty. Section 1-2 defines the tuning problem precisely and identifies the gap in existing methods. Section 1-3 presents the proposed approach, the research question, and the main contributions. Section 1-4 defines the research scope, and Section 1-5 provides an outline of the remaining chapters.

1-1 Building Energy Hubs and Their Control

1-1-1 Buildings as Energy Hubs

A modern building contains multiple energy devices that interact with each other [3]. A heat pump converts electricity into heat. A battery stores and releases electricity. The thermal mass of the building stores heat passively. These devices are coupled through a shared electrical network, so that a decision made for one device cannot be made independently of the others [3, 4]. A second, independent form of coupling exists between the rooms themselves:

heat conducts through shared walls and floors, so that the temperature of one room depends not only on its own heating input but also on the temperature of its neighbors [5]. The energy hub formulation captures both couplings by treating the building as a single system in which all energy flows are optimized jointly [3].

In the configuration studied in this thesis, an electrical grid supplies power to a battery and a heat pump. The battery stores electricity and releases it during periods of high grid prices. The heat pump converts electrical energy into heating delivered to the building, and the radiators take heating from the heat pump and distribute it over the rooms of the building. All energy flows are coordinated at every time step, subject to comfort bounds, actuator limits, and energy balance constraints that link the subsystems [3]. A representative energy hub architecture is shown in Figure 1-1.

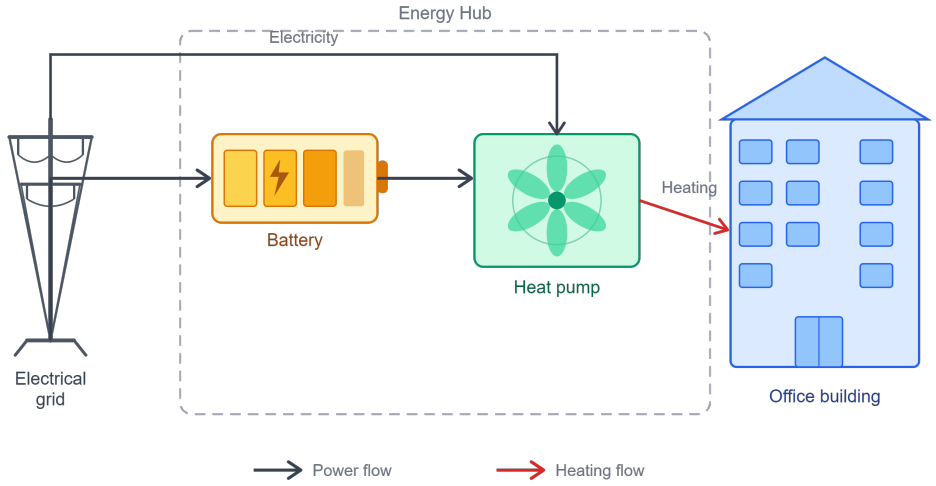


Figure 1-1: Representative building energy hub, illustrating the coupling between electricity and heating networks through conversion and storage devices.

Treating the hub's devices and rooms as a single system addresses the coupling introduced by the shared grid connection and by heat conduction between rooms, but this coupling is defined at a single instant in time and does not by itself determine how the building should be operated over time. Controlling a building energy hub therefore also benefits from a predictive controller with a sufficiently long planning horizon [2]. The thermal dynamics of building envelopes are slow: heat stored in walls, floors, and structural elements is exchanged over timescales spanning several hours [2, 4]. A controller without predictive capability cannot account for this inertia and cannot exploit the time-varying structure of electricity tariffs that require coordinated decisions over a multi-hour window [4].

The control objective is not to hold every room at a fixed temperature, but to minimize energy cost subject to comfort bounds [3]. Because thermal mass allows the building to drift within its comfort bounds and recover later, the controller can shift heating into periods of low electricity price and let the temperature fall during expensive periods, provided it can predict far enough ahead to guarantee that comfort is restored before the building is next occupied. This trade-off between cost and comfort cannot be expressed as a fixed setpoint. It requires optimizing a cost function over the horizon, not merely predicting future temperature.

Operational constraints must also be satisfied simultaneously. These include thermal comfort bounds, actuator power limits, and energy balance equations that couple the electrical and thermal subsystems [3]. These constraints act at the level of the hub rather than at the level of an individual device or room. The power drawn by the heat pump and the battery cannot be decided independently, since both draw from the same grid connection, and the comfort bound on each room depends on the heat conducted from its neighbors. Solar irradiance and ambient temperature act as time-varying disturbances that continuously influence the thermal state of the building and must be accounted for over the prediction horizon [2, 4].

1-1-2 Control of Building Energy Hubs

Model Predictive Control (MPC) addresses these requirements by using a model of the system to predict future behavior and selecting inputs that minimize a cost function subject to all operational constraints, applied in a receding-horizon fashion [2]. Advanced control of building HVAC systems has been reported to achieve average energy savings of 13 to 28% compared to conventional control [2]. However, constructing an accurate model of a building energy hub is costly. A building model must capture heat transfer between zones, the thermal properties of walls and ceilings, and the behavior of HVAC actuators, and in a coupled hub any modeling error propagates through the energy balance constraints and degrades performance across all subsystems [2, 6, 4].

Data-Enabled Predictive Control (DeePC), proposed in [7], removes this modelling requirement by replacing the explicit system model with a data structure built from input-output measurements collected during an offline excitation phase. In this phase, a test signal is applied to the building and the resulting input-output trajectories are recorded over a period long enough to capture the slow thermal dynamics of the system [7, 4]. The recorded input and output trajectories are organized into Hankel matrices, which are partitioned into blocks (U_p, Y_p) collecting past segments and (U_f, Y_f) collecting future segments [7]. By a result known as Willems' Fundamental Lemma, any trajectory the system can produce can then be written as a linear combination of the columns of these matrices [8, 7]:

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix}. \tag{1-1}$$

Here, g is a coefficient vector that selects a combination of the recorded trajectory segments, $(u_{\text{ini}}, y_{\text{ini}})$ are the most recent measured inputs and outputs, and (u, y) are the future inputs and outputs over the prediction horizon [7]. In MPC, a state estimator uses recent measurements to produce an estimate of the system's current state. The model then predicts forward from that estimate. In Eq. (1-1), $(u_{\text{ini}}, y_{\text{ini}})$ takes the estimator's place. Provided enough past measurements are kept, $(u_{\text{ini}}, y_{\text{ini}})$ is consistent with only one possible underlying state. Requiring (u, y) to satisfy the same equation then constrains the predicted trajectory to that state [7]. The pair (u, y) takes the model's place, producing the prediction without either step ever being computed explicitly.

Like MPC, DeePC solves a receding-horizon optimization problem at each time step to compute control inputs that minimize a cost function subject to operational constraints [7]. In

practice, building dynamics are nonlinear and the recorded trajectories are corrupted by sensor noise, which can make the equality constraint Eq. (1-1) infeasible and the optimization sensitive to noise in the stored data [7, 9]. To address this, a slack variable is introduced on the past outputs and regularization terms are added to the cost function [7, 9]. With the quadratic regularization used in this thesis, motivated by the robust formulation of [9], the resulting optimization problem takes the form:

$$\min_{g,\,u,\,y,\,\sigma_y} \; J(u,y)+\lambda_g\|g\|_2^2+\lambda_y\|\sigma_y\|_2^2 \quad \text{subject to} \quad \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} + \sigma_y \\ u \\ y \end{bmatrix}, \; u \in \mathcal{U}, \; y \in \mathcal{Y}. \quad \text{(1-2)}$$

Here, $J(u, y)$ is the operating cost over the horizon, which in the energy hub setting is the electricity cost, σ_y is the slack variable that absorbs the mismatch between the measured past outputs and the data-based representation, and $\mathcal{U}$ and $\mathcal{Y}$ collect the input and output constraints [7, 9]. The weights $\lambda_g > 0$ and $\lambda_y > 0$ are the regularization parameters. The term $\lambda_g\|g\|_2^2$ limits the influence of poorly excited or noise-dominated directions in the recorded data, and the term $\lambda_y\|\sigma_y\|_2^2$ controls how strictly the past-output consistency is enforced [7, 9]. The full description of Eq. (1-1) and Eq. (1-2), including the construction of the Hankel matrices and the roles of the two weights, is given in Chapter 2.

Prediction, state estimation, and optimal control are all handled within the single optimization problem Eq. (1-2), without ever identifying an explicit model [7]. Comfort bounds, actuator limits, and energy balance constraints are incorporated directly into this optimization problem and enforced over the full prediction horizon, in exactly the same way as in MPC [7, 4].

DeePC is particularly well suited to building energy hubs for two reasons. First, it eliminates the modelling burden that prevents wide-scale deployment of MPC in buildings. Deploying DeePC on a new building requires only a data collection experiment, not the expert modelling effort that MPC demands [7, 4]. Second, when the building's operating conditions change, for example due to seasonal load shifts or changes in occupancy patterns, the controller can be updated by collecting new data rather than re-identifying a parametric model [4]. DeePC has been successfully applied to building energy hub control and shown to achieve competitive performance compared to model-based strategies, without requiring any parametric model of the building or hub dynamics [6, 10, 4]. DeePC is therefore chosen in this thesis as the control framework for building energy hubs.

The regularization terms in Eq. (1-2) stabilize the solution and maintain robustness to noise and nonlinearity [9, 4]. Their weights λ_g and λ_y, however, are tunable parameters whose values critically affect closed-loop performance [9, 11, 4]. Selecting these weights is the central problem addressed in this thesis.

1-2 The Hyperparameter Tuning Problem

The regularization parameters λ_g and λ_y in Eq. (1-2) control a fundamental trade-off between data fidelity and robustness to noise [9, 4]. Insufficient regularization causes the controller to overfit the available data and become sensitive to measurement noise, leading to erratic predictions and frequent comfort constraint violations [9, 4]. Excessive regularization causes

the controller to become overly conservative and fail to exploit the available data effectively, reducing economic performance [11, 4]. The effective range between these two failure modes is narrow [11], and it shifts with the characteristics of the building and the operating season [4]. Furthermore, the regularization weights do not act independently. Their effective values depend on the other parameters of the DeePC optimization problem, including the prediction horizon, the initialization horizon, and the length of the offline dataset, so that tuning cannot be reduced to adjusting each parameter in isolation [11, 4].

Evaluating a single candidate parameter combination is expensive. Building temperatures respond to control inputs over hours, and the effect of a parameter choice on energy cost and comfort constraint satisfaction cannot be assessed until the system has been operated over a representative period spanning multiple days [11, 4]. When a grid of candidate combinations must be evaluated for each building separately, the total tuning effort grows quickly as the number of buildings increases [4].

The optimal parameter values are building-specific and cannot be transferred across buildings without re-tuning. Buildings differ in thermal mass, insulation quality, floor area, and zone count, and these physical properties determine how the building responds to control inputs and disturbances [4]. Across published building studies, the selected values of λ_g differ by more than three orders of magnitude [10, 12]. Because the relationship between measurable building characteristics and optimal DeePC parameters is not understood analytically, each new building requires its own independent tuning campaign [4].

Despite the practical success of DeePC in building energy systems, selecting suitable regularization parameters remains a significant challenge. The optimal values depend on the physical characteristics of the building and cannot be determined analytically, yet evaluating each candidate combination requires running a closed-loop simulation over several days [11, 4]. This makes the tuning process expensive, building-specific, and difficult to scale as the number of buildings grows [4]. A method that can predict suitable values of λ_g and λ_y for a new building from its measurable physical characteristics, without requiring any closed-loop simulation of that building, would therefore remove a key barrier to the practical deployment of DeePC at scale.

1-3 Research Objective and Approach

The tuning problem identified in Section 1-2 calls for a method that can predict suitable DeePC parameter values for a new building without requiring any closed-loop evaluation of that building. A method addressing this problem must satisfy three requirements. First, it must learn from closed-loop performance data collected across multiple buildings with different physical characteristics, so that the relationship between building properties and suitable parameters can be identified. Second, it must generalise to buildings not seen during training, producing reliable recommendations from measurable physical characteristics alone. Third, it must be sample-efficient, since each closed-loop simulation of a building energy hub is expensive and the number of available training buildings is small in practice [11, 4].

1-3-1 The Proposed Approach

This thesis proposes to address the tuning problem by learning a mapping from measurable building physical characteristics to suitable DeePC parameter values using Gaussian Process (GP) regression [13]. GP regression has already been used to tune controllers outside the DeePC setting. GP-based Bayesian optimization has been used to tune a controller on a real precision motion system [14]. The same principle has been applied to safe automatic controller tuning in robotics, including quadrotor experiments [15]. These results motivate applying the same approach to DeePC regularization parameters. GP regression is a machine learning method that learns a mapping from observations, and produces both a prediction and a calibrated measure of uncertainty at every query point [13]. Figure 1-2 illustrates this behavior: the posterior mean tracks the observed points closely, while the confidence band narrows near them and widens in regions farther from any observation.

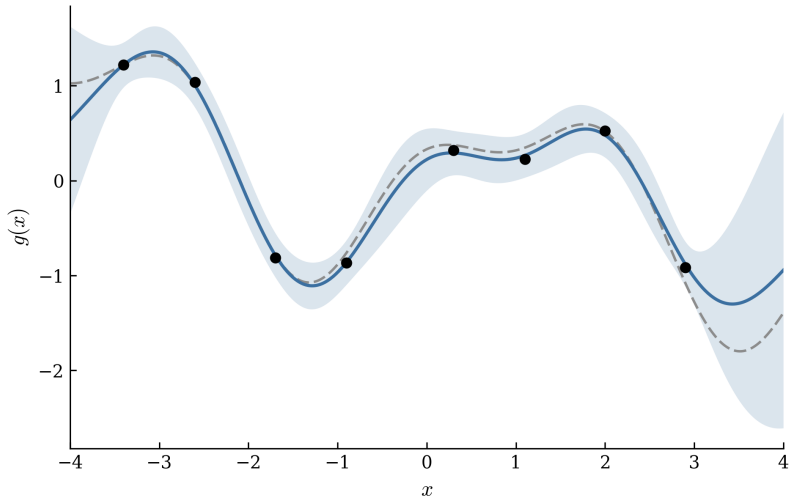


Figure 1-2: A GP posterior mean $\mu(x)$ with a 95% confidence band, fitted to noisy observations of an unknown function $g(x)$. The band narrows around observed points and widens elsewhere, giving a calibrated measure of uncertainty at every query point.

This behavior is exactly what makes GP regression well suited to the tuning problem. The number of available training buildings is small since each closed-loop simulation is expensive, reliable uncertainty estimates are important when recommending parameters for a building that was not seen during training, and the method naturally encodes the assumption that buildings with similar physical characteristics require similar regularization parameters [13, 4]. These properties make GP regression a more suitable choice than methods that require large datasets or do not provide uncertainty estimates, such as neural network regression [13].

The core idea is that buildings with similar physical characteristics respond similarly to control inputs and therefore require similar regularization parameters. By collecting closed-loop performance data across a set of simulated buildings with varying physical characteristics, a GP model can learn how the regularization parameters should be chosen as a function of the building's properties. Once trained, the model can produce a parameter recommendation for a new, unseen building in seconds, based on its measurable physical characteristics alone, without any additional closed-loop simulation.

The central research question of this thesis is:

> *How can a Gaussian Process regression model, trained on closed-loop performance data from a set of simulated building energy hubs, learn transferable tuning rules that map measurable building properties to suitable DeePC hyperparameters?*

To address this question, the following sub-questions are investigated:

SQ1: How does DeePC closed-loop performance vary with regularization parameter choice across buildings with different insulation levels, floor areas, and zone counts, and across different seasons?

SQ2: How accurately does a GP regression model trained on a small set of building energy hubs learn the mapping from building physical properties and regularization parameters to closed-loop performance metrics?

SQ3: How well do the parameter recommendations produced by the trained GP model generalise to buildings not seen during training, and how does the resulting closed-loop performance compare to exhaustive parameter search?

1-3-2 Main Contributions

The main contributions of this thesis are:

1. A systematic parameter sweep study across multiple simulated building energy hubs with varying insulation levels, floor areas, and zone counts, characterising how DeePC closed-loop performance and comfort constraint satisfaction depend on the regularization parameters λ_g and λ_y across different buildings and seasons.
2. A pair of GP regression models, one for comfort violation and one for energy cost, that jointly map the physical properties of a building energy hub and a candidate parameter combination to predicted closed-loop performance, providing a fast and transferable alternative to exhaustive parameter search.
3. An evaluation of the learned tuning rules on unseen building energy hubs, demonstrating that the GP models generalise across buildings not seen during training and produce parameter recommendations that achieve competitive closed-loop performance without any additional closed-loop simulation of the target building.

1-4 Research Scope

This thesis focuses on the two regularization weights of DeePC, applied to simulated building energy hubs. The structural parameters of DeePC, including the prediction horizon, the initialization horizon, and the data window length, are fixed based on established heuristics from prior energy hub studies and are not part of the tuning problem studied here. Similarly,

the comfort violation penalty is fixed and not tuned. The study is conducted entirely in simulation using the Building Resistance-Capacitance Modelling (BRCM) and Energy Hub Component Modelling (EHCM) toolboxes. The simulation environment models single-zone and multi-zone office buildings with an electric heat pump and a battery. Buildings are parameterized by wall and roof insulation thickness, floor area, and zone count, and the study covers a range of insulation levels and building sizes representative of typical office buildings. The GP model is trained on a set of these buildings and evaluated on a separate set of unseen buildings drawn from the same class. Real building deployments are outside the scope of this work. The gap between simulation and real deployment introduces additional challenges, including unmodeled dynamics, sensor noise, and occupancy uncertainty, that are not addressed here. The scope does not extend to fundamentally different building types such as industrial facilities or residential buildings, or to energy hubs with significantly different device configurations.

1-5 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 reviews the theoretical background and the existing literature, covering the DeePC framework and the role of the regularization parameters, applications of DeePC to building energy systems, existing hyperparameter tuning methods, and GP regression for controller tuning, and states the research gap. Chapter 3 describes the methodology, including the simulation environment, the building feature vector and performance metrics, the parameter sweep design, the GP models, the selection rule, and the evaluation procedure. Chapter 4 presents the results, covering the sweep results and GP model fit for the three training buildings across four seasons, and the validation on buildings outside the training set through interpolation, extrapolation, and a cross-seasonal analysis. Chapter 5 describes how the learned tuning rules would be applied to a real building, identifies the remaining gaps between the simulation study and practical deployment, draws conclusions, and outlines directions for future work. Figure 1-3 summarizes this structure.

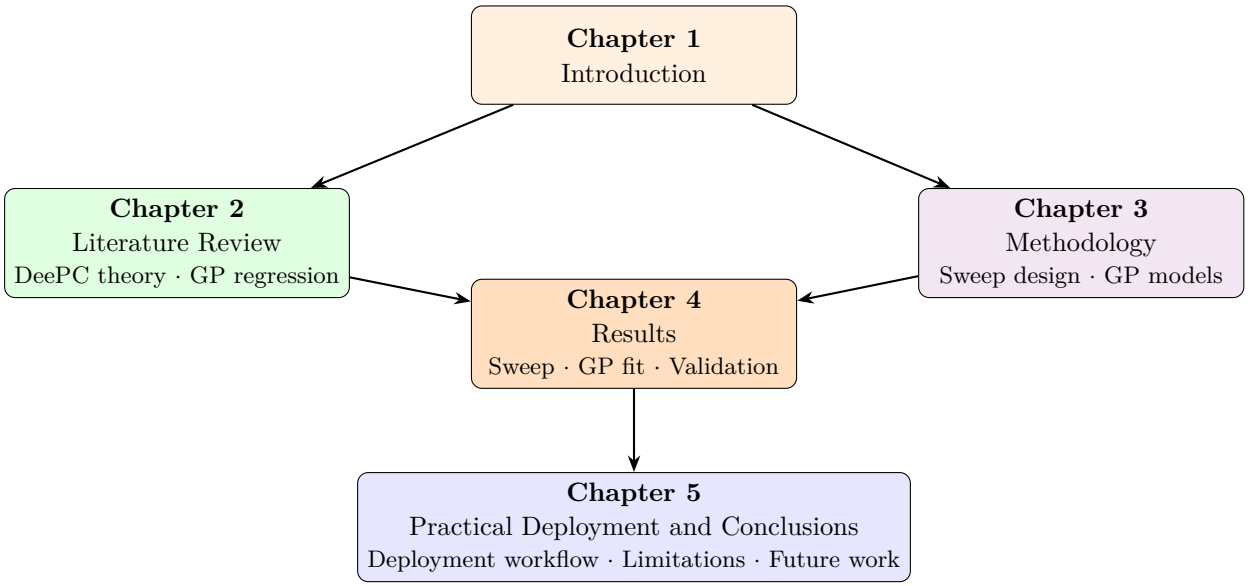


Figure 1-3: Thesis structure.

Chapter 2

Literature Review

This chapter reviews the existing literature on Data-Enabled Predictive Control (DeePC) and its application to building energy systems, with a focus on the hyperparameter tuning problem that motivates this thesis. Section 2-1 introduces the DeePC framework and develops the theory of its regularization parameters. Section 2-2 surveys how DeePC has been applied to building energy systems and what the empirical evidence reveals about tuning difficulty. Section 2-3 reviews existing automated tuning methods and identifies their limitations. Section 2-4 introduces Gaussian Process (GP) regression and reviews its application to controller tuning problems. Section 2-5 states the research gap addressed by this thesis.

2-1 Data-Enabled Predictive Control

2-1-1 The DeePC Framework

The theoretical foundation of DeePC is Willems' Fundamental Lemma [8], which states that for a controllable Linear Time-Invariant (LTI) system driven by a persistently exciting input, the columns of the Hankel matrix constructed from the measured trajectories span the complete set of trajectories the system can produce [8]. The Hankel matrix of depth L for a measured input signal $u \in \mathbb{R}^{mT}$ is defined as [7]:

$$\mathcal{H}_L(u) := \begin{bmatrix} u_1 & u_2 & \cdots & u_{T-L+1} \\ u_2 & u_3 & \cdots & u_{T-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ u_L & u_{L+1} & \cdots & u_T \end{bmatrix}, \tag{2-1}$$

where each column corresponds to one length-L segment of the recorded trajectory. The output Hankel matrix $\mathcal{H}_L(y)$ is defined analogously. Setting $L = T_{\text{ini}} + N$, where T_{ini} is the initialization horizon and N is the prediction horizon, the Hankel matrices are partitioned into past and future blocks [7]:

$$\begin{pmatrix} U_p \\ U_f \end{pmatrix} := \mathcal{H}_{T_{\text{ini}}+N}(u^d), \qquad \begin{pmatrix} Y_p \\ Y_f \end{pmatrix} := \mathcal{H}_{T_{\text{ini}}+N}(y^d). \tag{2-2}$$

The matrices (U_p, Y_p) collect the first T_{ini} block rows and are used to initialize the controller with the most recent measured inputs and outputs. The matrices (U_f, Y_f) collect the remaining N block rows and are used to generate predictions over the control horizon.

By the Fundamental Lemma, any trajectory compatible with the system dynamics can be expressed as a linear combination of the Hankel columns. There exists a coefficient vector g such that [7]:

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix}. \tag{2-3}$$

This is the representation previewed in Eq. (1-1), now derived from the Hankel construction in Eq. (2-1) and Eq. (2-2). The vector g selects a combination of stored trajectories that matches the most recent measured data $(u_{\text{ini}}, y_{\text{ini}})$ and predicts the future trajectory (u, y), replacing both the state estimate and the parametric model used in Model Predictive Control (MPC) [7]. Substituting into a receding-horizon cost yields the basic DeePC optimization problem [7]:

$$\min_{g,\,u,\,y} \sum_{k=0}^{N-1} \left(\|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 \right) \quad \text{s.t.} \quad \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix},\ u_k \in \mathcal{U},\ y_k \in \mathcal{Y}. \tag{2-4}$$

Under the conditions of the Fundamental Lemma, Eq. (2-4) produces the same closed-loop trajectories as an MPC controller with a perfectly identified model [7]. System identification, state estimation, and control are handled within a single optimization problem, without requiring knowledge of any parametric model [7].

2-1-2 Theory of Regularization

In practice, measured trajectories contain sensor noise and the system is nonlinear and time-varying. Under these conditions, the equality constraint in Eq. (2-4) may become infeasible, and the implicit inversion of the Hankel matrices can amplify noise in poorly excited data directions, producing erratic predictions [7, 9]. To address these problems, the authors in [7] introduce a slack variable σ_y that relaxes the past-output consistency constraint, together with one-norm regularization penalties on g and σ_y. Quadratic variants of these penalties, which replace the one-norms with squared two-norms, are analyzed in [9] and shown to admit a robustness interpretation; the quadratic form is the one adopted in this thesis and previewed in Eq. (1-2). With quadratic penalties, the regularized problem reads [9]:

$$\min_{g,\,u,\,y,\,\sigma_y} \sum_{k=0}^{N-1} \left(\|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 \right) + \lambda_g \|g\|_2^2 + \lambda_y \|\sigma_y\|_2^2 \tag{2-5}$$

subject to

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} + \sigma_y \\ u \\ y \end{bmatrix}, \quad u_k \in \mathcal{U},\ y_k \in \mathcal{Y},\ k = 0, \ldots, N-1,$$

where $\lambda_g > 0$ and $\lambda_y > 0$ are the regularization weights and $\sigma_y \in \mathbb{R}^{T_{\text{ini}}p}$ is the slack variable that absorbs inconsistencies between the measured past outputs and the Hankel-based representation [9].

Role of λ_g

The term $\lambda_g \|g\|_2^2$ penalizes large values of the coefficient vector g. The vector g selects a linear combination of Hankel columns to match the current initialization data and predict the future trajectory. When λ_g is small, the optimizer is free to assign large weights to many Hankel columns, including those corresponding to poorly excited or noise-dominated directions in the data. By penalizing large g, the regularization restricts the optimizer to combinations of Hankel columns with smaller, more balanced weights. This suppresses the contribution of unreliable data directions and stabilizes the implicit Hankel inversion [7, 9]. This restriction also shapes the control actions. A small g norm limits how far the predicted trajectory can deviate from the behaviors represented in the recorded data, which has been associated with more cautious control actions [4].

The study in [16] analyzes what the regularization on g does to the predictions themselves. It shows that when λ_g is sufficiently large, the predictions produced by regularized DeePC coincide with those of a linear regression model estimated from the same data [16]. It also shows that increasing λ_g beyond this point distorts the control criterion and degrades closed-loop performance, because the regularization term begins to dominate the objective [16]. In this regime, the controller loses the ability to respond to economic signals such as time-varying electricity prices, and closed-loop cost increases [16, 11]. When λ_g is small, the predictions are free to deviate from the estimated-model behavior [16]; the empirical studies reviewed in Section 2-2-2 show that under measurement noise this freedom manifests as noise amplification and unreliable predictions [11, 12, 10]. Between these two extremes lies an effective intermediate range in which the regularization suppresses noise-dominated Hankel directions while retaining informative ones [12, 11, 4]. The location of this range is not fixed across systems. It shifts with the noise level and conditioning of the Hankel matrices, which depend on the specific building and on the offline data collection procedure [11, 4].

Role of λ_y

The term $\lambda_y \|\sigma_y\|_2^2$ controls how strictly the past-output consistency constraint is enforced. The slack variable σ_y absorbs the mismatch between the measured past outputs y_{ini} and the outputs predicted by the Hankel-based representation for the same initialization window. Under noise-free conditions this mismatch is zero, but in practice sensor noise and nonlinearity make exact consistency infeasible [7, 9].

A small λ_y allows the slack to absorb a large portion of the mismatch, relaxing the consistency requirement. This improves feasibility under noisy conditions but may cause the optimizer to ignore the measured past outputs when selecting g, which degrades the implicit state estimate that the initialization data provide [9, 4]. A large λ_y forces the predicted past outputs to match the measurements closely, which sharpens the implicit state estimate but can propagate measurement noise directly into the prediction of future outputs, introducing numerical stiffness and degrading prediction quality [9, 4]. The sensitivity analysis in [7]

shows that varying λ_y over several orders of magnitude while fixing λ_g produces significant changes in both closed-loop cost and constraint violation duration, confirming that λ_y has an independent and material effect on controller performance.

Joint Tuning and Parameter Interaction

The two regularization weights do not act independently. The study in [9] develops a robust DeePC framework in which the Hankel matrices are treated as uncertain, modeled as bounded perturbations of their true values, and a min-max optimization is solved over the resulting uncertainty set. The inner maximization identifies the worst-case data perturbation for a given g; the outer minimization finds the g that minimizes cost under that worst case [9]. A key result of this framework is that the performance guarantee holds when the consistency weights are jointly chosen to be sufficiently large, with the required magnitudes depending on the cost weighting and on a bound determined by the underlying system [9]. These system-dependent quantities vary from building to building, which means the joint requirement on the weights changes with the building being controlled [9, 4]. A single pair chosen for one building therefore carries no guarantee for a different building [4].

Empirical evidence on the joint behavior of the two weights is limited. The sensitivity study in [7] varies each weight while the other is held at a single fixed value, establishing that both weights materially affect closed-loop cost and constraint violations, but not how the effective range of one weight moves with the value of the other. The study in [11] evaluates λ_g jointly with the initialization horizon T_{ini} and shows that the closed-loop effect of λ_g changes with T_{ini}, confirming that the regularization weights must be tuned jointly with the structural parameters of the DeePC problem, not in isolation [11, 4].

Implications for Tuning

The theoretical literature has direct consequences for how the two regularization weights must be selected in practice. The weight λ_g governs a trade-off between noise amplification, informative prediction, and loss of economic responsiveness, established across theoretical and empirical studies, and the boundaries of this trade-off shift with the noise level and conditioning of the building's Hankel matrices [16, 11, 12]. The weight λ_y governs a related but distinct trade-off between the quality of the implicit state estimate and the propagation of measurement noise into the prediction [9]. The two weights cannot be tuned independently. The joint requirement is established theoretically [9], both weights are shown empirically to have a material effect on closed-loop performance [7], and the effective range of λ_g shifts with structural parameters such as the initialization horizon [11]. No analytical formula that maps measurable building characteristics to suitable values of (λ_g, λ_y) currently exists [9, 16, 4].

2-2 DeePC in Building Energy Systems

2-2-1 Applications to Building Energy Hubs

DeePC has been applied to a growing number of building energy systems and energy hub configurations. These studies share a common motivation. Building dynamics are slow,

thermal coupling between subsystems is strong, and collecting sensor data is easier than developing and maintaining a physical model [4]. Each application also shows how λ_g and λ_y are tuned in practice, and what this reveals about the difficulty of selecting suitable values.

The first application of DeePC to a building energy hub using the Building Resistance-Capacitance Modelling (BRCM) and Energy Hub Component Modelling (EHCM) toolboxes is in [10]. The system consists of a multi-zone office building, a heat pump, and a battery, with the control objective of minimizing electricity cost subject to thermal comfort constraints. In the noise-free case, the regularization weight λ_g is identified through a grid search over ten values from 5×10^{-4} to 5×10^{5}, while λ_y is held fixed; the two smallest values of λ_g render the optimization infeasible due to numerical problems, and the selected value $\lambda_g = 3 \times 10^{-2}$ achieves a cost reduction of around 10% relative to a rule-based Bang-Bang baseline [10]. When measurement noise is added, both λ_g and λ_y are searched jointly over a ten-by-ten grid of 100 combinations, and the resulting parameters still fail to produce satisfactory closed-loop behavior. The identified weight is too small, the algorithm overfits the noise, and the study concludes that the number of simulations was not large enough to find the parameters robustly [10]. The study identifies systematic hyperparameter tuning as the primary direction for future work [10].

The study in [6] applies DeePC to a five-zone building and an energy hub comprising an electric heat pump and a battery, in a one-year simulation that explicitly accounts for battery degradation alongside comfort and cost. Compared to a rule-based strategy, the rule-based controller violates the comfort constraints up to 5.5% of the time, whereas DeePC violates them up to 2.8% of the time and with smaller violations, while achieving a two-fold decrease in battery degradation, without any parametric model of the building or hub dynamics [6]. The regularization weight is fixed at $\lambda_g = 1000$, introduced to avoid overfitting to noise in the data; no sensitivity analysis over this weight is reported [6]. Because the weight was selected for this specific simulation environment, the study offers no guidance on choosing it for a building with different insulation or zone count [4].

The study in [17] applies data-driven predictive control to the bilinear Heating, Ventilation, and Air-Conditioning (HVAC) dynamics of a real meeting room at the Bosch research campus. Constraint adaptation is used to compensate for plant-model mismatch that would otherwise cause constraint violations [17]. In the two reported simulation scenarios, the conventional benchmark controller incurs 19.9% and 60.4% higher cost than the data-driven controller respectively [17]. The study does not report a sensitivity analysis of the regularization weights, so it offers no guidance on their selection for other buildings [17, 4].

2-2-2 Parameter Sensitivity Studies

Several studies conduct structured sensitivity analyses of λ_g to understand how closed-loop performance changes with the regularization weight. These studies provide the most direct empirical evidence for the system-specificity of the tuning problem.

The study in [11] applies DeePC to a five-zone office building HVAC system modeled in EnergyPlus and systematically varies λ_g over $\{1, 10, 30, 100\}$ jointly with $T_{\text{ini}} \in \{2, 4, 6, 8\}$ and the prediction horizon $N \in \{6, 12, 18, 24\}$. The results show that closed-loop cost depends on λ_g without any monotonic trend of improvement or degradation across the evaluated values [11]. The value $\lambda_g = 10$ is adopted for the main simulations, and the closed-loop

effect of λ_g is shown to change with the initialization horizon, so the two parameters cannot be selected independently [11]. The analysis is restricted to one-day simulations to keep the number of evaluations manageable [11]. The identified values are specific to the studied building and conditions, and the study provides no mechanism for transferring them to a different building or season [11, 4].

The study in [12] applies DeePC to combined temperature and humidity control in a museum building, where strict climate requirements and persistent sensor noise make regularization particularly critical. The regularization weight is varied over nine values from 10^0 to 10^4, with prediction quality assessed using the Normalized Mean Squared Error between predicted and measured outputs. Values in the range $\lambda_g \in [10^1, 10^2]$ consistently yield the lowest prediction error for both temperature and relative humidity. Below this range, the predictor becomes sensitive to noise and produces unstable trajectories. Above it, the Hankel-based representation is overly restricted and the controller underreacts to temperature disturbances [12]. Once λ_g is within the effective range, the prediction horizon N has minimal influence on prediction quality, indicating that regularization is the dominant factor governing prediction stability in this setting [12]. The study fixes $\lambda_y = 10^6$ throughout, reflecting the practical choice of keeping the slack penalty large enough that the slack variable remains small [12].

The study in [18] takes a different approach by separating the prediction and control objectives into two levels, so that λ_g can be tuned offline on historical data rather than through closed-loop simulation. A sensitivity analysis on data from the Polydome building at EPFL varies λ_g over a grid and finds that a wide range of values spanning several orders of magnitude produces consistent prediction accuracy, with degradation occurring only when λ_g is too large [18]. The wider effective range compared to standard closed-loop DeePC is attributed directly to the decoupling of prediction from control. In the standard formulation, λ_g simultaneously affects prediction quality and the control criterion, which narrows the range of acceptable values [18]. Even with this reduced sensitivity, λ_g is still identified empirically on a per-building validation set, confirming that no transferable rule for its selection exists even in the most favorable formulation [18].

Across all reviewed studies, the same qualitative picture emerges [4]. Insufficient regularization leads to noise amplification and unreliable predictions. Excessive regularization overly constrains the Hankel-based predictor and degrades control responsiveness. The best performance is consistently obtained within a system-dependent intermediate range of λ_g. The location of this range, however, differs substantially across studies. The selected values of λ_g span more than three orders of magnitude, from 3×10^{-2} in [10] to values in $[10^1, 10^2]$ in [12], all within building thermal control applications [4]. Every reviewed study also evaluates a single building under a single operating period. None reports how a building's effective range shifts across seasons, and none reports parameter values found on one building and applied to another [4]. No study reports a method for predicting where the effective range lies for a new building without conducting closed-loop experiments on that building [4].

2-3 Existing Hyperparameter Tuning Methods

The evidence in Section 2-2 establishes that λ_g and λ_y must be identified separately for each building through closed-loop simulation, but the evaluation cost is high and the results do not

transfer. This section reviews the methods proposed in the literature to address this tuning problem.

The dominant approach across all reviewed studies is manual tuning through grid search over a discrete set of candidate values. In DeePC, unlike the cost matrices of MPC, λ_g and λ_y simultaneously affect the numerical conditioning of the optimization and the effective control criterion, which makes standard design rules inapplicable [16, 9]. Practitioners are left with no analytical starting point and must evaluate candidate combinations directly. In building energy hub applications, each evaluation requires a closed-loop simulation over several days, and the results give no information about suitable values for a different building [10, 11, 4].

2-3-1 Automated Per-System Methods

Gradient-Based Tuning

The study in [19] proposes DeePC-Hunt, a method that formulates regularization parameter selection as a differentiable optimization problem. When the DeePC optimization is formulated as a differentiable optimization problem, the optimal control sequence is a differentiable function of λ_g and λ_y for a fixed dataset. The gradient of the closed-loop cost with respect to the regularization weights can then be computed by backpropagating through the solver, and the weights are updated by gradient descent [19]. Rather than requiring direct interaction with the real system, DeePC-Hunt uses an approximate model of the system dynamics to simulate the closed-loop response during the optimization, making it safer than manual guess-and-check methods that must evaluate parameters on the actual system [19]. The approach is demonstrated on a nonlinear rocket landing stabilization task and achieves higher success rates than a model-based MPC controller under model mismatch [19].

The fundamental limitation of DeePC-Hunt is transferability. The gradient is computed from a fixed dataset collected from a specific system, so the weights it converges to are tied to that system's dynamics and noise characteristics. Applying the method to a new building requires collecting a new dataset from that building and re-running the full optimization, which requires either an approximate model of the new building or direct experimentation on it [19, 4].

Reinforcement Learning

The study in [20] formulates the online selection of λ_g as a Markov Decision Process and solves it using the SARSA reinforcement learning algorithm. At each time step, the method observes the current input-output behavior of the system and selects a value of λ_g from a discrete candidate set based on a learned action-value function, which is updated online using the observed closed-loop reward [20]. Over ten independent experimental trials, the approach reduces the mean absolute error in temperature tracking by 72.5% compared to baseline fixed-parameter DeePC [20].

The action-value function learned by SARSA encodes which values of λ_g produce good performance given the observed behavior of the specific system on which it was trained. This knowledge does not carry over to a building with different thermal dynamics and noise properties. Training a useful policy for a new building requires collecting closed-loop data from

that building, which means that no recommendation can be produced before the controller is deployed and operated [20, 4].

2-4 Gaussian Process Regression for Controller Tuning

2-4-1 Gaussian Process Regression

A GP is a probability distribution over functions, fully specified by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$ [13]. A function f follows a GP prior if any finite collection of function values has a joint Gaussian distribution [13]:

$$f \sim \mathcal{GP}\big(m(\mathbf{x}),\ k(\mathbf{x}, \mathbf{x}')\big)\,. \tag{2-6}$$

The mean function encodes prior beliefs about the expected function value and is commonly set to zero [13]. The kernel encodes the assumed smoothness. Input points that are close together are expected to produce similar function values, with the degree of similarity governed by the kernel parameters [13].

Given n noisy observations $\hat{f}(\mathbf{x}_i) = f(\mathbf{x}_i) + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, \sigma_n^2)$, the posterior distribution at a new test point $\mathbf{x}^*$ is Gaussian with mean [13]:

$$\mu(\mathbf{x}^*) = \mathbf{k}(\mathbf{x}^*)^\top \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \hat{\mathbf{f}}, \tag{2-7}$$

and variance [13]:

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^\top \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{k}(\mathbf{x}^*), \tag{2-8}$$

where $\hat{\mathbf{f}} \in \mathbb{R}^n$ collects the observations, $\mathbf{K} \in \mathbb{R}^{n \times n}$ has entries $[\mathbf{K}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{k}(\mathbf{x}^*)$ collects the kernel evaluations between $\mathbf{x}^*$ and each training point [13]. The posterior mean $\mu(\mathbf{x}^*)$ is the prediction at $\mathbf{x}^*$ and $\sigma^2(\mathbf{x}^*)$ is the associated uncertainty. The posterior variance grows as $\mathbf{x}^*$ moves away from the training data, providing a calibrated measure of extrapolation uncertainty [13].

The choice of kernel determines the smoothness structure of the functions the GP can represent. The squared-exponential kernel,

$$k_{\mathrm{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\!\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\ell^2}\right), \tag{2-9}$$

assumes infinitely differentiable functions and encodes a strong smoothness prior [13]. The exponential kernel,

$$k_{\mathrm{Exp}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\!\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|}{\ell}\right), \tag{2-10}$$

assumes functions that are continuous but not necessarily differentiable, making it appropriate when the modeled function may exhibit sharper transitions between regions [13].

The kernel hyperparameters σ_f, ℓ, and σ_n^2 are learned from data by maximizing the log marginal likelihood [13]:

$$\log p(\hat{\mathbf{f}} \mid \mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2}\hat{\mathbf{f}}^\top \left(\mathbf{K}_\theta + \sigma_n^2 \mathbf{I}\right)^{-1} \hat{\mathbf{f}} - \frac{1}{2}\log\det\!\left(\mathbf{K}_\theta + \sigma_n^2 \mathbf{I}\right) - \frac{n}{2}\log 2\pi, \tag{2-11}$$

where $\boldsymbol{\theta}$ collects the kernel hyperparameters and $\mathbf{K}_\theta$ is the corresponding kernel matrix [13]. On small datasets, marginal likelihood maximization can converge to local optima that interpolate the training data closely but generalize poorly to new inputs [13]. Leave-One-Out (LOO) cross-validation provides an alternative that directly measures predictive accuracy on held-out points [13]:

$$\text{LOO-Root Mean Square Error (RMSE)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{f}(\mathbf{x}_i) - \mu_{-i}(\mathbf{x}_i)\right)^2}, \tag{2-12}$$

where $\mu_{-i}(\mathbf{x}_i)$ is the posterior mean at $\mathbf{x}_i$ trained on all observations except the i-th.

2-4-2 GP Regression for Controller Tuning

Safe Tuning with GP Surrogates

The study in [15] develops a framework for safe Bayesian optimization that generalizes the SafeOpt algorithm to handle multiple safety constraints decoupled from the performance objective. A GP surrogate is trained on closed-loop evaluations of the controller performance and the safety constraints. The posterior uncertainty is used to define a safe region in parameter space that expands as more evaluations are collected, ensuring that unsafe parameter combinations are never evaluated during the learning process [15]. The approach is demonstrated on a quadrotor vehicle and achieves safe and automatic tuning of control parameters with few evaluations [15]. The separation of the performance objective from the constraint measures, each modeled through its own surrogate, is structurally similar to the approach used in this thesis, which trains one GP for comfort violation and one for energy cost, with the constraint enforced through the selection rule. The same framework is extended to a contextual setting, in which an external variable is included in the GP kernel alongside the controller parameters, so that correlations encoded by the joint kernel allow a safe parameter recommendation to be transferred to a context that has not yet been evaluated [15]. In the reported experiments, the context represents the flying speed of a single quadrotor, so the transfer occurs across different operating conditions of one fixed vehicle, not across different vehicles with different physical properties [15].

The study in [21] reformulates the SafeOpt algorithm as a series of optimization problems rather than an exhaustive grid search, reducing the computational cost while preserving the safety guarantees. The method is demonstrated on a controller tuning application and shown to achieve better computational efficiency than the original SafeOpt formulation [21]. The study in [14] provides experimental validation on an industrial-grade precision motion system, demonstrating 30% better tracking than a state-of-the-art safe learning algorithm and a nearly sevenfold reduction in computational cost relative to grid-based safe learning [14]. Together, these results confirm that GP-based surrogate models can guide safe parameter tuning with far fewer evaluations than grid search.

GP Surrogates for MPC and Related Tuning

The study in [22] applies Bayesian optimization with a GP surrogate to tune MPC parameters for lithium-ion battery fast charging. A hierarchical framework is used in which the GP

surrogate optimizes the global closed-loop performance objective while the MPC handles short-term constraint satisfaction [22]. The GP is initialized with a small number of closed-loop evaluations and updated iteratively until near-optimal parameters are found, achieving fast charging while keeping the battery terminal voltage within safe limits [22]. The study demonstrates that GP surrogates are sample-efficient for controller tuning problems where each evaluation is expensive [22].

The study in [23] applies constrained GP surrogates to the simulation-based optimization of a solar process heat system. Shape constraints, namely boundedness, monotonicity, and concavity, are imposed on the surrogate itself, and an active learning strategy selects which simulations to run [23]. Adding the shape constraints reduces the amount of training data and time required to reach a given surrogate quality, compared to an unconstrained GP [23]. The study demonstrates that encoding prior structural knowledge in the surrogate reduces the number of expensive simulations needed, a consideration that applies directly to closed-loop controller tuning, where each evaluation is a multi-day simulation [23, 4].

2-4-3 The Cross-System Gap

All GP-based tuning methods reviewed in this section train the surrogate on evaluations of a single target system and use it to optimize parameters for that same system. A new building, battery, or process would require retraining the surrogate from scratch using evaluations collected from the new system [22, 23]. The contextual extension in [15] is the closest existing mechanism to a cross-system formulation. It also augments the GP kernel with an external variable to transfer knowledge to conditions that have not yet been evaluated. The context variable in [15] is trajectory speed, an operating condition of a single quadrotor. Transferring across buildings instead requires the GP to generalize across different physical systems, not just across operating conditions of one system, and no existing application of contextual Bayesian optimization uses a system's physical construction parameters as the context variable. Transfer learning in building energy modeling targets a related but different problem, reusing a model trained on one building to predict energy use in a different building rather than transferring controller hyperparameters [24]. No existing method expresses DeePC regularization weights as a function of measurable building characteristics, learned through GP regression trained across multiple energy hubs. Such a formulation would allow a new hub to be assigned hyperparameters directly from its physical characteristics, through the posterior mean of a trained GP, without requiring any closed-loop evaluation of that hub. This formulation has not been implemented. No closed-loop data have been collected across multiple hubs, no GP has been trained, and no validation against unseen buildings has been reported. This thesis provides the first implementation and empirical validation, training a pair of GP regression models across simulated buildings with varying physical properties and evaluating the resulting recommendations through both leave-one-out cross-validation and closed-loop performance on unseen buildings (Chapter 3).

2-5 Research Gap

The literature reviewed in this chapter establishes a clear and specific gap in the existing body of work on DeePC for building energy systems. The gap has four distinct dimensions

that together motivate the approach taken in this thesis.

The first dimension concerns the absence of an analytical tuning rule. The theoretical studies in [16] and [9] show that the optimal values of λ_g and λ_y depend on the noise level, the conditioning of the data, and system-dependent quantities, all of which vary across buildings. The performance guarantee of robust DeePC [9] requires the consistency weights to be sufficiently large relative to a bound determined by the underlying system, but that bound is not accessible before the controller is deployed on a new building. No expression that maps measurable building characteristics such as insulation level, zone count, or thermal time constant to suitable regularization values has been established in the literature [4]. Without such a rule, every new building requires its own independent tuning campaign.

The second dimension concerns the system-specificity of all existing tuning results. Every study reviewed in Section 2-2 identifies its regularization parameters through manual grid search, heuristic selection, or online adaptation, and the resulting values are specific to the building on which they were found. The preferred values of λ_g span more than three orders of magnitude across building studies, from 3×10^{-2} in [10] to values in $[10^1, 10^2]$ in [12]. The study in [11] shows that even within a single building, the effect of λ_g changes with the initialization horizon, and the study in [18] confirms that per-building empirical selection is still required even when the tuning problem is simplified by decoupling prediction from control. No study reports parameter values that were found on one building and applied successfully to a different building without re-tuning [4]. Nor does any study evaluate the same building across multiple seasons, leaving the seasonal dimension of the tuning problem uncharacterized [4].

The third dimension concerns the evaluation cost. Each closed-loop evaluation of a building energy hub requires a simulation spanning multiple days to capture the slow thermal dynamics and the effect of the time-varying electricity tariff [10, 11]. The study in [11] had to restrict its sensitivity analysis to one-day periods to keep the number of evaluations manageable, and the study in [10] reports that identifying suitable parameters under measurement noise required searching a ten-by-ten grid of 100 combinations, and even then failed to find values that performed robustly. As the number of buildings grows, the total evaluation cost scales proportionally, making exhaustive grid search impractical as a general deployment strategy [4]. This cost also makes it difficult to apply automated single-system tuning methods such as Bayesian optimization or gradient-based search directly, since each evaluation of a new combination requires a multi-day simulation [19, 4].

The fourth dimension concerns the joint tuning of λ_g and λ_y across systems. Most empirical studies focus exclusively on λ_g and fix λ_y at a large value without studying the interaction between the two weights [12, 6, 11]. The theoretical result of [9] ties the joint requirement on the weights to system-dependent quantities, and the sensitivity results of [7] show that both weights materially affect closed-loop cost and constraint satisfaction. No existing study characterizes this joint dependence across multiple buildings or uses it to derive transferable recommendations [4].

None of the automated tuning methods reviewed in Section 2-3 close any of these four gaps. Gradient-based tuning [19], reinforcement learning [20], and online feedback optimization [25] all operate on a single system and produce parameters that are tied to that system's dynamics and data. The GP-based surrogate methods reviewed in Section 2-4 are sample-efficient and produce calibrated uncertainty estimates, but all existing applications train and evaluate on

a single target system without including the building's physical characteristics as inputs to the surrogate [22, 23, 15, 4].

The gap addressed by this thesis is therefore the following. No method currently exists that can predict suitable values of λ_g and λ_y for a new building energy hub from its measurable physical characteristics, without requiring any closed-loop evaluation of that building. This thesis proposes to close this gap by training a pair of GP regression models across multiple simulated buildings with varying physical properties, including the building characteristics alongside the candidate parameter values as inputs to the surrogate. The physical properties of the building determine the thermal dynamics, the noise characteristics of the Hankel matrices, and the effective range of the regularization weights, so a model that takes those properties as inputs can learn how the optimal pair (λ_g, λ_y) shifts across buildings and generalize to an unseen building from its measurable characteristics alone. The proposed approach and its evaluation are described in Chapter 3 and Chapter 4.

Chapter 3

Methodology

This chapter describes the simulation environment, the learning problem, and the procedures used to train and evaluate the Gaussian Process (GP) tuning models. Section 3-1 describes the building energy hub simulation environment, including the building model, the energy hub components, and the Data-Enabled Predictive Control (DeePC) formulation specific to this problem. Section 3-2 defines the building feature vector, the performance metrics, and the GP learning problem. Section 3-3 covers the design of the parameter sweep used to generate training data. Section 3-4 describes the GP models, including the kernel, normalization, and hyperparameter optimization. Section 3-5 presents the selection rule used to recommend a hyperparameter pair for a new building. Section 3-6 defines the evaluation procedure used to assess the quality of the learned tuning rules.

3-1 Simulation Environment

All simulations in this thesis are conducted using two MATLAB-based toolboxes, the Building Resistance-Capacitance Modelling (BRCM) Toolbox for building dynamics and the Energy Hub Component Modelling (EHCM) Toolbox for energy hub management. These toolboxes are chosen because together they provide the only established open simulation environment in which DeePC has previously been applied to building energy hubs [10, 6], which allows the results of this thesis to be compared against prior work on the same foundation. This section describes the building model, the energy hub components, the DeePC formulation used for control, and the disturbance and tariff signals.

3-1-1 Building Model

The BRCM Toolbox [5] generates physics-based multi-zone office building models from construction parameters. Each zone corresponds to one room. The toolbox represents the building as a resistance-capacitance network in which the thermal states correspond to the air

temperatures of each room and the temperatures of the layers of each building element (floors, ceilings, inner walls, and outer walls). The continuous-time building model takes the form [6]:

$$\dot{x}_s(t) = A_c x_s(t) + B_u u_s(t) + B_v v_s(t) + \sum_i \left(B_{vu,i}\, v_s(t) + B_{xu,i}\, x_s(t)\right) u_{s,i}(t), \tag{3-1}$$

where $x_s \in \mathbb{R}^{n_x}$ is the state vector of zone and wall-layer temperatures, $u_s \in \mathbb{R}^{n_u}$ is the control input vector (radiator heating powers and blind positions), $v_s \in \mathbb{R}^{n_v}$ is the disturbance vector (ambient temperature, ground temperature, global solar radiation on each facade, and internal gains from occupancy and equipment), and the matrices A_c, B_u, B_v, $B_{vu,i}$, $B_{xu,i}$, C_c are generated by the toolbox from the building construction parameters [5, 6].

The model in Eq. (3-1) is bilinear because the blind positions appear in products with both the states and the disturbances. For the purposes of this thesis, the blind positions are fixed at their fully open setting and excluded from the control inputs, so that the building model reduces to a discrete-time Linear Time-Invariant (LTI) system with sampling time $T_s = 1$ h [6, 10]:

$$x_{t+1} = A x_t + B_u u_t + B_v v_t, \qquad y_t = C x_t, \tag{3-2}$$

where $y_t \in \mathbb{R}^{n_y}$ collects the room air temperatures. This LTI reduction satisfies the conditions required by Willems' Fundamental Lemma and is the standard approach used in prior DeePC studies with the BRCM framework [10, 6]. The sampling time of one hour is appropriate because the thermal dynamics of building envelopes evolve over timescales of several hours, so a finer discretization would increase the size of the Hankel matrices and the online optimization without capturing additional relevant dynamics [2, 4]. The state dimension n_x varies across hubs depending on the number of zones and wall layers; for the hubs used in this thesis it ranges from 88 to 139.

The comfort constraints imposed on the room temperatures are:

$$18\,^\circ\mathrm{C} \le y_{t,j} \le 22\,^\circ\mathrm{C} \quad \text{during occupied hours}, \qquad 10\,^\circ\mathrm{C} \le y_{t,j} \le 30\,^\circ\mathrm{C} \quad \text{otherwise}, \tag{3-3}$$

for each room j. Occupied hours are defined as 08:00 to 18:00 on weekdays. The tight occupied-hours band reflects the comfort requirement of an office workspace, while the loose band outside occupied hours gives the controller the freedom to let the building drift and recover, which is the source of the cost savings that predictive control can achieve [2]. The radiator heating power for each zone is constrained to $u_{\mathrm{rad},j} \in [0,\, u_{\mathrm{rad}}^{\max}]$, where $u_{\mathrm{rad}}^{\max}$ is the rated radiator capacity per unit floor area. The rated capacity is set per simulated season to match its heating demand. It is 70 W/m^2 for the March and September simulations, 30 W/m^2 for June, and 90 W/m^2 for December.

3-1-2 Energy Hub Components

The EHCM Toolbox [26] couples the building with an electric heat pump and a lithium-ion battery to form a building energy hub. The heat pump is modeled as a static device following the Coefficient of Performance (COP) relation used in [6], with the coefficient of performance fixed at COP = 3 in this thesis. The heat delivered to the building is three times the electrical power consumed by the pump at each time step. A static COP is a simplification, since the efficiency of a real heat pump varies with the temperature lift, but it keeps the hub model linear and is the standard treatment in prior DeePC studies on this environment [6, 10].

The battery is modeled as a discrete-time linear system with dynamics

$$\mathrm{SoC}_{t+1} = \mathrm{SoC}_t + \frac{T_s}{E_{\max}} \left(\eta_c \, P_c^t - \frac{1}{\eta_d} P_d^t \right), \tag{3-4}$$

where $\mathrm{SoC}_t \in [0, 1]$ is the state of charge, $P_c^t \geq 0$ and $P_d^t \geq 0$ are the charging and discharging powers, η_c and η_d are the charging and discharging efficiencies, and $E_{\max}$ is the nominal capacity [26]. The battery state of charge is constrained to $\mathrm{SoC}_t \in [0.2, 0.9]$ to prevent deep discharge and overcharge.

The energy hub imposes an electricity balance constraint at each time step:

$$p_t = P_{\mathrm{rad},t} + P_{\mathrm{hp},t} + P_{c,t} - P_{d,t}, \tag{3-5}$$

where p_t is the power purchased from the grid, $P_{\mathrm{rad},t}$ is the total radiator power across all zones, $P_{\mathrm{hp},t}$ is the heat pump electrical consumption, and $P_{c,t}$, $P_{d,t}$ are the battery charging and discharging powers [26].

3-1-3 DeePC Formulation for the Energy Hub

The EHCM Toolbox exports the combined building-hub system in a form compatible with the DeePC framework. The hub outputs (room temperatures and battery state of charge) are collected into a single output vector y_t, the hub inputs (radiator powers and battery charging and discharging commands) are collected into a single input vector u_t, and the predicted disturbances (weather, occupancy, electricity prices) are provided as a feedforward signal [26, 10].

The DeePC optimization problem solved at each time step is motivated by the robust formulation of [9] and adapted to the energy hub setting. The control objective is to minimize the total electricity cost over the prediction horizon rather than a tracking error, since the energy hub has no fixed reference trajectory but instead optimizes against a time-varying electricity tariff [6]. The formulation used in this thesis is:

$$\min_{g,\, u,\, y,\, \sigma_y,\, \sigma} \; c^\top u + Q_s \|\sigma\|_1 + \lambda_g \|g\|_2^2 + \lambda_y \|\sigma_y\|_2^2 \tag{3-6}$$

subject to

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\mathrm{ini}} \\ y_{\mathrm{ini}} + \sigma_y \\ u \\ y \end{bmatrix}, \quad y_k \in \mathcal{Y}_t + \sigma_k, \; u_k \in \mathcal{U}, \; k = 0, \ldots, N-1,$$

where $c \in \mathbb{R}^{n_u N}$ is the vector of electricity prices over the horizon, stacked to match the input dimension, $Q_s > 0$ is a fixed comfort violation penalty weight, $\sigma \in \mathbb{R}_{\geq 0}^{n_y N}$ is a slack vector that relaxes the comfort constraint $\mathcal{Y}_t$ by σ_k at each step, and $\sigma_y \in \mathbb{R}^{T_{\mathrm{ini}} n_y}$ relaxes the past-output consistency constraint [6, 9]. The ℓ_1 penalty on σ penalizes comfort violations sparsely, so that the controller is strongly incentivized to satisfy the comfort constraint but can do so softly when infeasibility would otherwise result.

Compared to the general regularized formulation reviewed in Chapter 2, formulation Eq. (3-6) differs in two respects. First, the stage cost replaces the quadratic tracking error with

the linear electricity cost $c^\top u$, reflecting that the energy hub objective is economic rather than reference-tracking. Second, a separate slack variable σ is introduced specifically for the comfort constraint, penalized with a fixed weight Q_s that is not part of the tuning problem. The two regularization weights λ_g and λ_y retain their role from the general formulation and are the quantities whose transferable tuning rules this thesis aims to learn [9, 4].

The DeePC structural parameters are fixed throughout all experiments at $T_{\text{ini}} = 30$ h and $N = 48$ h. The initialization horizon of 30 hours exceeds one full day of operation, so that the initial condition trajectory $(u_{\text{ini}}, y_{\text{ini}})$ always contains at least one complete daily cycle of occupancy and tariff variation, which fixes the slow thermal state of the building implicitly. The prediction horizon of 48 hours spans two full daily tariff cycles, so that the controller always has visibility of at least one complete off-peak period ahead and can schedule pre-heating and battery charging accordingly [4]. Fixing these parameters isolates the tuning problem to the two regularization weights. The study in [11] shows that the effect of λ_g changes with T_{ini}, so holding the structural parameters constant is necessary for the sweep results to be attributable to the regularization weights alone. The same initialization horizon of 30 hours is used in the prior DeePC studies on this simulation environment [10, 6]. The comfort penalty weight is fixed at $Q_s = 10^4$, chosen to be large enough that comfort violations are strongly discouraged relative to the electricity cost savings achievable through optimal scheduling. With this weight, one degree-hour of violation costs the optimizer more than any realistic saving from letting the temperature drift outside the band, so the slack activates only when the constraint cannot be met.

3-1-4 Disturbance and Tariff Signals

The disturbance vector v_t in Eq. (3-2) consists of the ambient air temperature, the ground temperature, the global solar radiation on each building facade, and the internal heat gains due to occupancy, lighting, and equipment in each zone. These signals are taken from historical weather data representative of central Europe, with peak solar gains occurring in June and July and minimum ambient temperatures in December and January. The Netherlands follows the same seasonal pattern. Internal gains follow a fixed occupancy schedule with peak loads during working hours on weekdays and zero gains on weekends. Using historical rather than synthetic weather data means the simulations contain realistic day-to-day variability, including warm spells and cold snaps within a season; the closed-loop trajectories in Chapter 4 show that these events shape the controller behavior visibly.

The electricity tariff c_t used in the cost function of Eq. (3-6) is a time-of-use tariff with a high price during peak hours (07:00–21:00) and a low price during off-peak hours (21:00–07:00). This time-varying structure provides the economic incentive for the controller to pre-heat the building during cheap periods, exploit battery storage to shift loads, and coordinate the heat pump and radiators to minimize total electricity cost while maintaining comfort [26, 6].

3-2 Problem Formulation

This section formally defines the hyperparameter tuning problem, the building feature vector, the closed-loop performance metrics, and the GP learning problem that forms the core of this thesis.

3-2-1 The Hyperparameter Tuning Problem

The DeePC formulation in Eq. (3-6) requires two regularization parameters, $\lambda_g > 0$ and $\lambda_y > 0$, to be fixed before the controller is deployed on a building. As established in Section 2-1-2, their optimal values depend on the physical properties of the building, the noise level of the offline data, and the conditioning of the Hankel matrices, none of which can be computed analytically for a new building. The standard approach is to evaluate a finite set of candidate combinations through closed-loop simulation and select the combination that best satisfies the performance objectives.

Let $\Lambda = \{(\lambda_g^{(i)}, \lambda_y^{(i)})\}_{i=1}^{M}$ denote the candidate grid. For a given building H and a given combination $(\lambda_g, \lambda_y) \in \Lambda$, a closed-loop simulation of duration T_{sim} is run, producing two scalar performance metrics, the comfort violation $v(H, \lambda_g, \lambda_y)$ and the energy cost $c(H, \lambda_g, \lambda_y)$, defined formally in Section 3-2-3. The per-hub tuning problem is then to find the combination in Λ that minimizes energy cost subject to an acceptable level of comfort violation.

The fundamental limitation of this per-hub approach is that each closed-loop simulation of a building energy hub requires T_{sim} hours of computation, which in the energy hub setting is several days to capture representative seasonal behavior [10, 11]. Evaluating all M combinations for a single building therefore requires $M \times T_{\text{sim}}$ hours of simulation, and the optimal combination found for one building does not transfer to a building with different physical characteristics [4]. Each new building requires its own independent evaluation campaign.

The tuning problem addressed in this thesis removes this dependence on per-building evaluation. Given a set of N_{train} training buildings $\{H_1, \ldots, H_{N_{\text{train}}}\}$ for which closed-loop performance data have been collected across all combinations in Λ, and given a new building H_{new} for which no closed-loop data are available, the goal is to find a combination $(\hat{\lambda}_g, \hat{\lambda}_y) \in \Lambda$ that is expected to achieve satisfactory closed-loop performance on H_{new}, using only the measurable physical properties of H_{new} and the performance data collected on the training buildings.

This cross-building learning formulation requires two components. The first is a building descriptor that characterizes each hub by its measurable physical properties. The second is a surrogate model that learns how the performance metrics depend jointly on those properties and on the candidate parameter combination. These are defined in the following subsections.

3-2-2 Building Feature Vector

Each building energy hub is described by a four-dimensional physical feature vector:

$$x_H = \begin{bmatrix} d_{\text{wall}} & d_{\text{roof}} & A_{\text{tot}} & \tau \end{bmatrix}^\top \in \mathbb{R}^4, \tag{3-7}$$

where d_{wall} (m) is the outer wall insulation thickness, d_{roof} (m) is the roof insulation thickness, A_{tot} (m^2) is the total conditioned floor area, and τ (h) is the dominant thermal time constant of the building.

These four features are chosen because they are measurable from building construction documentation without requiring any closed-loop experiment, and because they jointly determine the thermal response of the building to control inputs and external disturbances. Thicker

insulation increases the thermal resistance of the envelope, slowing the building's response to outdoor temperature changes and increasing the thermal time constant. A larger floor area increases the total heating load and the number of zones that must be coordinated. The thermal time constant τ is the most direct summary of how quickly the building responds to changes in heating power or ambient conditions, and is computed from the dominant eigenvalue of the matrix A in Eq. (3-2) as $\tau = -T_s / \log|\lambda_1(A)|$, where $\lambda_1(A)$ is the eigenvalue with the largest magnitude [5]. Measurability is the decisive criterion here. Quantities such as the noise level or the conditioning of the Hankel matrices, which the theory in Chapter 2 identifies as the direct determinants of the effective regularization range, are only available after data collection on the building, and using them as features would defeat the purpose of recommending parameters before any experiment is run.

The feature vector x_H captures the physical diversity across buildings in a low-dimensional and interpretable form. The zone count is not included as a separate feature because it is already partially captured by A_{tot}. Buildings with more zones at the same total area have smaller per-zone areas and different coupling patterns, but the total area and insulation thickness together summarize the dominant thermal behavior relevant for regularization [4]. Whether this omission is justified is tested directly in the room-count extrapolation experiments of Chapter 4.

3-2-3 Performance Metrics

For each building H and each hyperparameter combination $(\lambda_g, \lambda_y) \in \Lambda$, a closed-loop simulation of duration $T_{\text{sim}} = 2160$ h is run and two performance metrics are recorded.

The comfort violation metric v measures the cumulative temperature excess above the upper comfort bound and below the lower comfort bound during occupied hours, normalized by the number of zones and the simulation duration:

$$v = \frac{1}{n_y T_{\text{occ}}} \sum_{j=1}^{n_y} \sum_{t \in \mathcal{T}_{\text{occ}}} \max(0,\ y_{t,j} - 22) + \max(0,\ 18 - y_{t,j})\,, \tag{3-8}$$

where n_y is the number of zones, $\mathcal{T}_{\text{occ}}$ is the set of occupied time steps, and $T_{\text{occ}} = |\mathcal{T}_{\text{occ}}|$ is the total number of occupied hours in the simulation window. The violation metric v has units of °C and represents the average temperature excess per zone per occupied hour. The normalization by zone count and occupied hours is what makes v comparable across buildings with different numbers of zones and across seasons, which is essential for a training set that pools data from multiple buildings.

The energy cost metric c is the total electricity expenditure over the simulation window:

$$c = \sum_{t=1}^{T_{\text{sim}}} c_t \cdot p_t \cdot T_s, \tag{3-9}$$

where c_t is the electricity price at time t, p_t is the grid power purchased at time t, and $T_s = 1$ h is the sampling interval. The cost metric c has units of EUR and reflects the total electricity bill over the 90-day simulation. Unlike the violation metric, the cost metric is not normalized by floor area, because the selection rule in Section 3-5 only ever compares costs of different

parameter combinations on the same building, where the area is constant. Where costs are compared between buildings in Chapter 4, they are normalized by floor area.

Both metrics depend jointly on the building H and the hyperparameter combination (λ_g, λ_y). They capture different aspects of controller performance that are not always aligned. The comfort violation v measures constraint satisfaction, which is a primary requirement for any practical building controller, while the energy cost c measures economic efficiency, which is the primary optimization objective of the DeePC formulation in Eq. (3-6). A suitable parameter combination must achieve low comfort violation while also achieving competitive energy cost. The relationship between the two objectives is generally non-trivial. Parameter combinations that achieve very low violation may do so at the expense of higher energy cost, and combinations that minimize cost may tolerate larger comfort deviations. The selection rule in Section 3-5 resolves this trade-off by treating comfort satisfaction as a primary constraint and minimizing cost within the set of comfort-satisfying combinations.

3-2-4 The GP Learning Problem

The GP learning problem is to fit two surrogate models from the sweep data. Let $\mathcal{D} = \{(z_i, v_i, c_i)\}_{i=1}^{n}$ denote the training dataset, where each entry consists of the combined input vector

$$z_i = \begin{bmatrix} x_{H_i} \\ \log_{10}(\lambda_{g,i}) \\ \log_{10}(\lambda_{y,i}) \end{bmatrix} \in \mathbb{R}^6, \tag{3-10}$$

the observed comfort violation v_i, and the observed energy cost c_i from the simulation of building H_i at hyperparameter combination $(\lambda_{g,i}, \lambda_{y,i})$.

The log transformation of λ_g and λ_y is applied because the candidate grid Λ is logarithmically spaced and the performance metrics vary more smoothly as a function of $\log_{10}(\lambda)$ than of λ directly [4]. A GP is chosen as the surrogate model class for two reasons. First, the training set is small by construction, since every data point costs one 90-day closed-loop simulation, and GP regression is well suited to small datasets because it makes maximal use of every observation [13]. Second, the GP posterior variance provides a calibrated uncertainty estimate that grows with distance from the training data [13]; for a method whose purpose is to make recommendations for buildings it has never seen, knowing when the recommendation is uncertain is as important as the recommendation itself. The two GP models are:

$$f_v(z) \sim \mathcal{GP}\big(m_v(z),\ k_v(z, z')\big)\,, \quad \text{trained to predict } \log(v), \tag{3-11}$$

$$f_c(z) \sim \mathcal{GP}\big(m_c(z),\ k_c(z, z')\big)\,, \quad \text{trained to predict } \log(c). \tag{3-12}$$

The log transformation of the outputs is applied because both v and c are strictly positive and vary over several orders of magnitude across the training data, so that the log-transformed outputs are more nearly homoscedastic and better suited to GP regression with a stationary kernel [13]. Predictions of v and c for a new input z^* are recovered by exponentiating the posterior means: $\hat{v}(z^*) = \exp(\mu_{f_v}(z^*))$ and $\hat{c}(z^*) = \exp(\mu_{f_c}(z^*))$.

3-3 Parameter Sweep Design

The GP models are trained on simulation data collected from a set of training buildings. For each training building, DeePC is run with every combination of regularization weights in a candidate grid, and the resulting comfort violation and energy cost are recorded. This section describes the training buildings, the candidate grid, the data collection procedure, and the seasonal structure of the sweep.

3-3-1 Training Buildings

Three energy hub models are used as training buildings, referred to as Building 1, Building 2, and Building 3 throughout this thesis. Each model is generated using the BRCM toolbox [5] and the EHCM toolbox [26], following the setup described in Section 3-1. The buildings differ in the number of thermal zones, wall and roof insulation thickness, total floor area, and thermal time constant. The physical features of the three training buildings are listed in Table 3-1.

Table 3-1: Physical features of the training buildings. d_{wall}: outer wall insulation thickness; d_{roof}: roof insulation thickness; A_{tot}: total conditioned floor area; τ: dominant thermal time constant.

Building	Zones	d_{wall} (m)	d_{roof} (m)	A_{tot} (m^2)	τ (h)
Building 1	6	0.150	0.020	504	113.48
Building 2	5	0.100	0.015	420	94.60
Building 3	4	0.110	0.013	288	86.17

The three buildings are chosen to span a range of insulation levels, floor areas, and zone counts representative of typical office buildings. Building 1 is the largest and best-insulated building, with six zones and the highest thermal time constant. Building 3 is the smallest, with four zones, the thinnest roof insulation, and the lowest time constant. Building 2 has the thinnest wall insulation of the three, while its roof insulation, floor area, zone count, and time constant lie between those of the other two buildings. This selection places the training points at the extremes and the interior of each feature dimension rather than clustering them, so that the training data cover a meaningful portion of the feature space. The wall insulation ranges from 0.100 m to 0.150 m, the roof insulation from 0.013 m to 0.020 m, the thermal time constant from 86.17 h to 113.48 h, and the floor area from 288 m^2 to 504 m^2. Three buildings is the minimum number that provides an interior point in addition to the two extremes of each feature range, and it keeps the total simulation budget of the four-season sweep at 300 closed-loop simulations of 90 days each, which is the practical limit of the computational budget of this thesis.

3-3-2 Candidate Hyperparameter Grid

The candidate grid Λ consists of all combinations of λ_g and λ_y from the discrete set $\{10^1, 10^2, 10^3, 10^4, 10^5\}$. This gives $|\Lambda| = 25$ combinations arranged on a 5×5 logarithmic grid. The grid spans four orders of magnitude in each parameter and covers the range of values found to be relevant for building energy applications in the literature. The effective values reported for λ_g in building

studies lie in $[10^1, 10^2]$ [12] and around 10^1–10^3 [11, 6], while values of λ_y up to 10^5–10^6 are used to keep the slack variable small [7, 12]. The lower bound of the grid is set at 10^1 because values below 10^0 are reported to cause severe performance deterioration [12] and numerical infeasibility [10] in building applications. For each training building, DeePC is run once for each of the 25 combinations, yielding 25 data points per building and 75 data points per season before any point removal.

The choice of a logarithmic grid is motivated by the non-monotonic dependence of closed-loop performance on both parameters. The relevant range spans multiple orders of magnitude, and performance changes more smoothly when the parameters are varied on a log scale [11, 4]. Equal spacing in log-space ensures that the training data are distributed uniformly across the entire candidate range. A 5×5 resolution is the coarsest grid that still provides interior points in both dimensions, and it bounds the per-season simulation budget at 75 runs; a finer grid would improve the resolution of the learned surface at a directly proportional increase in simulation cost.

3-3-3 Data Collection and Simulation Setup

The simulation for each training building and hyperparameter combination follows a two-phase procedure.

In the first phase, a Pseudo-Random Binary Sequence (PRBS) signal is applied to all control inputs of the building hub for $T_{\text{collect}} = 1600$ h to collect the input-output data used to construct the Hankel matrices. The PRBS signal is designed to be persistently exciting of sufficiently high order for the building dynamics, as required by Willems' Fundamental Lemma (Section 2-1-1) [8, 7]. The collection length of 1600 h provides a number of Hankel columns far in excess of the minimum required for the window length used here, which matters because surplus columns average out noise in the data-driven representation [7, 4]. During data collection, the disturbance inputs (weather data and occupancy) are applied in open loop with their nominal values; no feedback is active. The Hankel matrices U_p, Y_p, U_f, Y_f are constructed from the collected trajectories using the partition in Eq. (2-2), with window length $L = T_{\text{ini}} + N = 30 + 48 = 78$ h. Only the output rows corresponding to room temperatures and the battery state of charge are included in the Hankel output matrices, following the hub output definition of [10]; internal wall temperatures are excluded because they are not directly relevant to the control objectives and their inclusion would increase the problem dimension without improving prediction accuracy.

In the second phase, DeePC is run in closed loop using the collected Hankel matrices and the optimization problem in Eq. (3-6) for a simulation window of $T_{\text{sim}} = 2160$ h, corresponding to 90 days. A 90-day window is long enough to average over the day-to-day weather variability of the historical data and over many tariff cycles, so that the recorded metrics reflect sustained closed-loop behavior rather than a favorable or unfavorable stretch of weather. At each time step, the DeePC optimization is solved using the Operator Splitting Quadratic Program (OSQP) solver [27], with absolute and relative tolerances of 10^{-4}. A solution is considered infeasible if the solver reports an infeasible or inaccurate status; in that case the previous control input is held constant for the current time step. The Hankel matrix construction, the fixed portions of the quadratic program, and the initial state are computed once before the sweep loop to minimize computational overhead.

3-3-4 Seasonal Sweep Structure

The optimal hyperparameter combination varies between seasons because the thermal load on the building changes with outdoor temperature and solar irradiation [4]. In summer, the building requires less heating and solar gains increase the risk of overheating; the DeePC controller must respond more aggressively to price signals to avoid unnecessary heating costs. In winter, the building requires sustained heating to maintain comfort, and the optimal regularization may differ from summer because the signal-to-noise ratio in the Hankel matrices changes with the operating regime. To capture this seasonal dependence, the sweep is run separately for four seasons, one per quarter of the year, namely March, June, September, and December. The March simulation starts at day 60 of the year, corresponding to the beginning of March. The June simulation starts at day 151, corresponding to the beginning of June. The September simulation starts at day 242, corresponding to the beginning of September. The December simulation starts at day 333, corresponding to the beginning of December, with the window extending past the turn of the calendar year. These four starting points are chosen so that each 90-day window captures a distinct thermal regime. March covers the transition out of winter, June covers the summer minimum of heating demand, September covers the transition into autumn, and December covers the winter peak. Figure 3-1 shows the ambient temperature and solar radiation for the full year, with the four simulated windows highlighted, confirming that each window falls in a distinct part of the annual cycle.

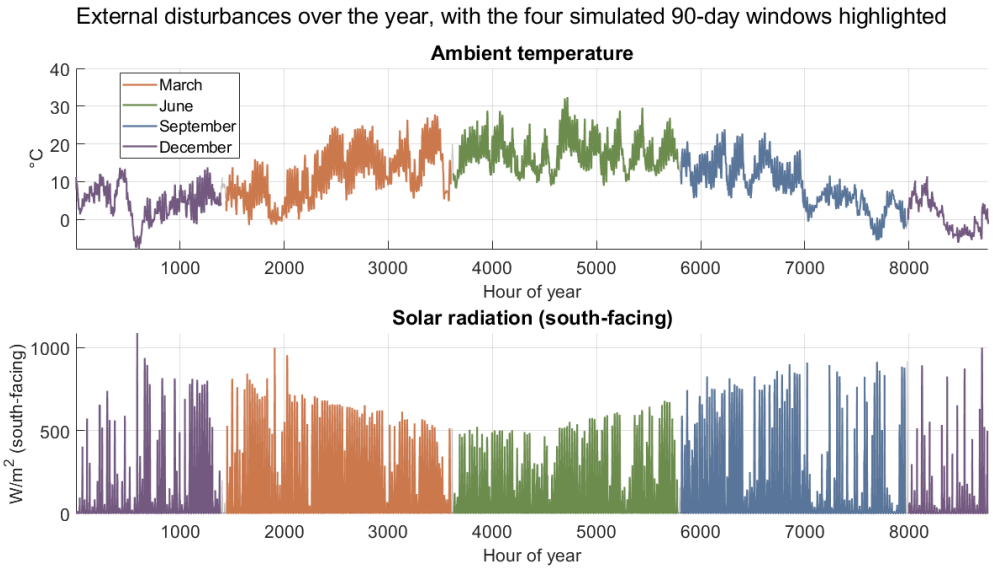


Figure 3-1: Ambient temperature and south-facing solar radiation over the full year. The four highlighted windows are the 90-day periods simulated for each season.

A separate pair of GP models (one for violation and one for cost) is fitted for each season, so that the tuning rules reflect the thermal and climatic conditions of the target operating period. When a recommendation is required for a new building in a given season, the GP models trained on that season's data are queried.

3-4 Gaussian Process Models

Two GP models are fitted to the sweep data of each season: one for comfort violation and one for energy cost, as defined in Eq. (3-11) and Eq. (3-12). This section describes the kernel and basis function selection, the normalization procedure, the hyperparameter optimization strategy, and the point removal step.

3-4-1 Kernel and Basis Function

Both GP models use an exponential kernel with a constant basis function. The exponential kernel between two inputs z and z' is:

$$k(z, z') = \sigma_f^2 \exp\left(-\frac{\|z - z'\|}{\ell}\right), \tag{3-13}$$

where $\sigma_f^2 > 0$ is the signal variance and $\ell > 0$ is the length scale. The exponential kernel is the Matérn-$\frac{1}{2}$ kernel and models functions that are continuous but not necessarily differentiable [13].

The choice of the exponential over the squared-exponential kernel is motivated by the observed behavior of the performance metrics across the candidate grid. The squared-exponential kernel assumes infinitely differentiable functions, which is appropriate when the output varies smoothly and gradually with the input. In the sweep data, the violation metric v changes abruptly as λ_g passes below a threshold where noise amplification becomes dominant, and the cost metric c exhibits a plateau followed by a sharp rise when λ_g becomes too large. These transitions are better captured by the exponential kernel, which allows sharper changes in the posterior while still encoding a smoothness prior through the length scale [13, 4].

The constant basis function adds a learnable mean offset to the GP prior. This is necessary because the log-transformed outputs have a non-zero mean that varies across buildings, and a zero-mean prior would introduce a systematic bias in predictions for buildings whose performance levels differ from the training mean [13].

3-4-2 Normalization

All input dimensions are normalized to zero mean and unit variance before fitting the GP models. Without normalization, the physical features (d_{wall} in meters, A_{tot} in m^2, τ in hours) have very different scales, which causes the single length scale ℓ in the exponential kernel to be dominated by the largest-magnitude dimension and to be uninformative for the others [13]. Normalization places all six dimensions on the same scale before computing pairwise distances, so that the length scale reflects the smoothness of the function jointly in all input directions [13].

The normalization parameters (mean μ_d and standard deviation s_d for each dimension d) are computed per season from that season's training data after point removal, and applied identically to training inputs and to any new building queried at inference time. The normalization statistics for the March season are listed in Table 3-2.

Table 3-2: Normalization statistics for the GP input vector, computed from the March training data after point removal.

Feature	Mean μ_d	Std. deviation s_d
d_{wall} (m)	0.1193	0.0216
d_{roof} (m)	0.0159	0.0029
A_{tot} (m^2)	402.8333	88.7508
τ (h)	97.3483	10.6976
$\log_{10}(\lambda_g)$	3.0694	1.4075
$\log_{10}(\lambda_y)$	2.9167	1.3916

On the full 25-point grid, the means of $\log_{10}(\lambda_g)$ and $\log_{10}(\lambda_y)$ are exactly 3.0, because the grid Λ is symmetric around 10^3, and their standard deviations are approximately $\sqrt{2} \approx 1.41$, as expected for five equally spaced values on a log scale with step size 1. After point removal, the statistics deviate slightly from these grid values, since the removed points are not distributed uniformly over the grid.

3-4-3 Hyperparameter Optimization

The GP kernel hyperparameters are the signal variance σ_f^2, the length scale ℓ, and the noise variance σ_n^2. Both models are fitted with **MATLAB!** (**MATLAB!**)'s `fitrgp` function. The length scale and the signal standard deviation are obtained from the internal maximization of the log marginal likelihood Eq. (2-11) performed by `fitrgp` [13]. The noise standard deviation σ_n is selected separately through Bayesian optimization, using the expected-improvement-plus acquisition function with 30 objective evaluations. The noise level is treated separately because marginal likelihood optimization on small datasets is sensitive to local minima and may require informed initialization or regularization of the hyperparameter search [13]. The violation and cost GP models are fitted separately under this procedure, and each converges to its own kernel hyperparameters; the fitted values per season are reported in Chapter 4. The cost GP consistently converges to a substantially larger length scale than the violation GP, reflecting that energy cost varies more smoothly across the regularization grid than comfort violation does.

3-4-4 Point Removal

Before fitting the GP models, each season's training set is screened for points with extreme comfort violation. These points correspond to hyperparameter combinations for which the DeePC controller fails outright. Including such extreme points in the training data distorts the GP posterior in two ways. First, the log transformation applied to the violation output amplifies large violation values, producing outliers that disproportionately influence the fitted length scale and pull the GP toward a short correlation length. Second, the GP expends modeling capacity on representing the steep gradient from the acceptable region to the failure region, at the expense of accuracy within the practically relevant range of violations. Removing these points trades coverage of the failure region for accuracy in the region where recommendations are actually made. This is acceptable because the purpose of the model is to identify good combinations, not to predict the precise magnitude of failures. Points

were removed from the March and June training sets, and no points were removed from the September or December training sets. The criterion used to identify these points is not the same in every season, since a single rule did not fit the data well across all four. The criterion used in each season, and the reasoning behind it, is given alongside the corresponding fit in Chapter 4.

3-5 Selection Rule

Given a new building H_{new} with feature vector $x_{H_{\text{new}}}$, the trained GP models are queried at all 25 combinations in Λ to produce predicted violations $\hat{v}_i$ and predicted costs $\hat{c}_i$ for $i = 1, \ldots, 25$. The selection rule then proceeds in two steps.

In the first step, the minimum predicted violation over the grid is identified:

$$\hat{v}_{\min} = \min_{i \in \Lambda} \hat{v}_i. \tag{3-14}$$

In the second step, a feasible set $\mathcal{C}$ is formed from all combinations whose predicted violation is within a tolerance ε of $\hat{v}_{\min}$:

$$\mathcal{C} = \{\, i \in \Lambda : \hat{v}_i \leq (1 + \varepsilon)\, \hat{v}_{\min} \,\}. \tag{3-15}$$

The recommended combination is the element of $\mathcal{C}$ with the lowest predicted cost:

$$i^* = \text{argmin}_{i \in \mathcal{C}}\ \hat{c}_i. \tag{3-16}$$

A tie in predicted cost within $\mathcal{C}$ would leave i^* undefined by Eq. (3-16) alone. If it did occur, the tied combinations would already be comfort-equivalent by construction and cost-equivalent by the tie itself, so any of them would be an equally valid choice. The rule breaks such a tie by selecting the combination with the lower grid index, so that i^* remains well defined.

The rationale for this two-step rule is that comfort and cost are not equally important objectives. The comfort constraint must be satisfied before any cost optimization is meaningful. The first step therefore identifies the comfort-optimal region of the parameter grid, and the second step selects the cheapest option within that region. The tolerance ε prevents the rule from over-committing to a single comfort-optimal combination when the predicted differences within the comfort-optimal set are too small to be meaningful. A value of $\varepsilon = 0.05$ is used throughout this thesis, meaning that any combination whose predicted violation is within 5% of the minimum is treated as comfort-equivalent and the cheapest among them is selected. This value is a design choice fixed a priori. A 5% difference in mean comfort violation is small compared to the variation of the violation metric across the grid, which spans one to two orders of magnitude within a single building, so combinations within this band are practically indistinguishable in comfort terms. The consequences of fixing this tolerance, including two cases in which the fixed threshold interacts with prediction error, are examined in Chapter 4, and an uncertainty-scaled alternative is identified as future work in Chapter 5.

The same rule is also applied directly to the measured sweep data of each training building, in which case the predicted violations and costs in Eq. (3-14)–Eq. (3-16) are replaced by the

measured values. This defines the recommended combination of each training building from its own sweep, and allows the GP recommendations and the measured recommendations to be compared on equal terms in Chapter 4. All simulations were run in **MATLAB!** on a laptop with an Intel(R) Core(TM) i7-8750H CPU at 2.20 GHz and 16 GB of RAM, running Windows 11 Home. The full selection procedure requires exactly 50 GP posterior mean evaluations (25 for violation, 25 for cost) and completes in under one second on this hardware.

3-6 Evaluation Procedure

The quality of the learned tuning rules is assessed in three stages. The first stage measures the predictive accuracy of the fitted models on their own training grid using Leave-One-Out (LOO) cross-validation. The second stage checks that each season's model recovers the measured recommended combination of each training building. The third stage evaluates the recommendations in closed loop on buildings that were not used for training.

3-6-1 Leave-One-Out Cross-Validation

LOO cross-validation is used to assess the predictive accuracy of the fitted GP models [13]. For each training point, the model predicts the held-out value using all remaining points, and the LOO Root Mean Square Error (RMSE) is computed over the training set after point removal:

$$\mathrm{RMSE}_{\mathrm{LOO}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{v}_i^{(-i)} - v_i\right)^2}, \tag{3-17}$$

where $\hat{v}_i^{(-i)}$ is the predicted violation at point i from a model trained on all points except the i-th, v_i is the observed violation, and n is the number of training points after point removal. A low LOO RMSE indicates that the fitted surface is consistent across the training grid and not dominated by individual points. This metric measures interpolation quality on the training distribution; it does not by itself establish generalization to unseen buildings, which is the purpose of the closed-loop validation in Section 3-6-3.

3-6-2 Training-Building Consistency Check

As a second check, each season's trained model is queried at the exact feature vector of each training building, and the selection rule of Section 3-5 is applied to the predictions. The resulting recommendation is compared against the recommended combination obtained by applying the same rule to that building's measured sweep data. Exact recovery indicates that the model faithfully reproduces the data it was trained on at the points where ground truth is available. Where recovery is inexact, the measured gap between the model's recommendation and the sweep recommendation quantifies the practical consequence. This check is deliberately weak as a test of generalization, since the queried points are in the training set; its purpose is to establish a baseline of internal consistency before the harder tests that follow.

3-6-3 Validation on Buildings Outside the Training Set

The GP tuning rules are evaluated on thirteen building feature vectors that were not used for training, constructed as controlled variants of the training buildings so that the cause of any prediction error can be attributed to a specific feature change. Nine interpolation variants remain within the training feature range. For each training building, a Wall variant perturbs the wall insulation, a Roof variant perturbs the roof insulation, and a Wall+Roof variant perturbs both, with the thermal time constant left to react naturally to the insulation change. Two extrapolation variants move the roof insulation of Building 3 outside the training range, one above and one below. Two further buildings extend the room count one room below and one room above the training range, with insulation matched exactly to the nearest training building so that room count is the only substantive difference. Controlled variants are used instead of arbitrary new buildings because they isolate one feature change at a time; an arbitrary building that differed in every feature simultaneously would make it impossible to attribute a prediction error to its source.

For each of the thirteen feature vectors, the selection rule in Section 3-5 is applied using the trained March models to recommend a hyperparameter combination, and DeePC is then run in closed loop with the recommended combination for $T_{\text{sim}} = 2160$ h under the same two-phase procedure described in Section 3-3-3. Running the full 25-combination sweep on every validation building would require 13×25 additional 90-day simulations and is outside the computational budget of this thesis; the validation therefore evaluates the recommended combination only, and assesses it against three criteria. First, the achieved comfort violation is classified using the same tiers used throughout Chapter 4, with the green tier (violation below 0.010 °C) representing comfort performance comparable to the training buildings' own recommended combinations. Second, the achieved violation is compared against the 95% confidence interval predicted by the violation GP at the recommended combination; an achieved value inside the interval indicates that the model's uncertainty quantification remains valid at that point. Third, the relative error between predicted and achieved violation measures point-prediction accuracy, with the predicted value as denominator. Validation vectors that lie far from the training data are expected to produce wider confidence intervals, so the comparison between interval width and normalized feature-space distance provides a check on whether the GP uncertainty grows appropriately with extrapolation distance.

The closed-loop validation is conducted with the March models. To establish whether the observed behavior is specific to March or a property of the method, the June, September, and December models are additionally queried at the same thirteen feature vectors, and the resulting recommendations and confidence intervals are analyzed; no additional closed-loop simulations are run for these seasons, and this limitation is discussed explicitly where the cross-seasonal results are presented in Chapter 4.

Chapter 4

Results

This chapter evaluates the learning-based tuning approach introduced in Chapter 3. Section 4-1 presents the parameter sweep results for the three training buildings across all four seasons considered in this thesis, along with the Gaussian process models fitted to that data, one per season. For each season, this includes the measured violation and cost landscape across the regularization grid, the fitted model's quality, and a check of whether the model recovers each training building's own known optimum. This addresses how Data-Enabled Predictive Control (DeePC) closed-loop performance varies with the regularization parameters across buildings and seasons, and how accurately a Gaussian Process (GP) regression model trained on a small set of buildings learns the mapping from building properties and regularization parameters to closed-loop performance. Section 4-2 evaluates these trained models on buildings not used for training, both inside the training feature range (interpolation) and outside it (extrapolation). This addresses SQ3. Section 4-3 combines these results together and discusses what they imply for the tuning problem overall. The overall procedure spans several sections of Chapter 3.

Figure 4-1 summarizes it as a single sequence, from data collection to a recommended hyperparameter combination for a new building. The figure separates the procedure into two phases: an offline training phase, run once per season, in which data is collected from the three training buildings and used to fit a pair of Gaussian process models; and a deployment phase, run whenever a new building needs a recommendation, in which those already-trained models are queried to select a regularization combination for that building.

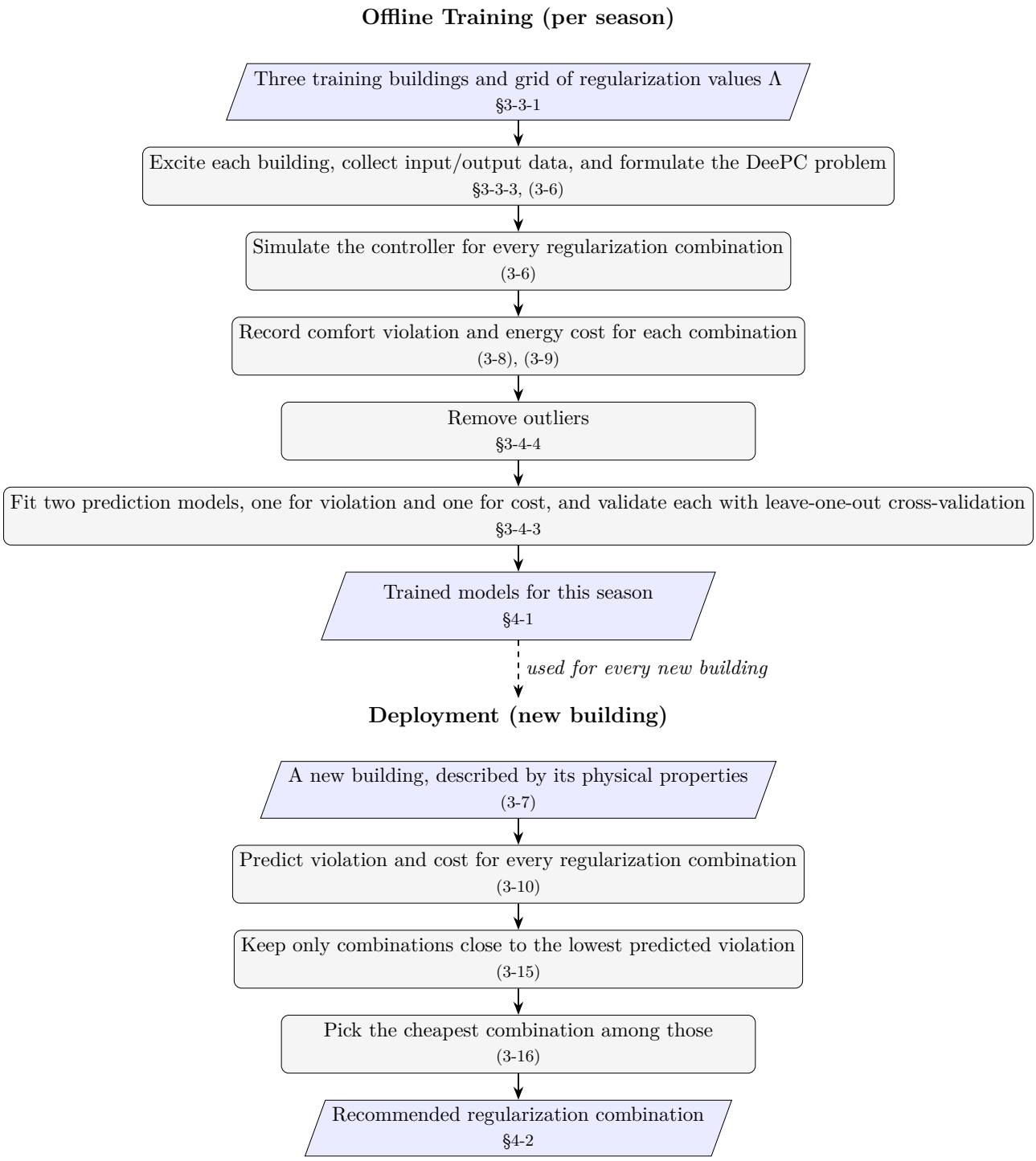


Figure 4-1: Overall approach: offline training produces one pair of prediction models per season; deployment uses those models to recommend a regularization combination for any new building.

4-1 Model Training and Fit

Three buildings, referred to as Building 1, Building 2, and Building 3, were used to train a separate Gaussian process model for each of the four seasons considered in this thesis: March, June, September, and December. The selection of these three buildings is described in §3-3-1. For each season, this section presents the parameter sweep results for each training building, the resulting Gaussian process fit, and a comparison of the fit across seasons. Table 4-1 lists the concrete experimental parameters used to produce these results.

Table 4-1: Experimental parameters used to produce the results in this chapter.

Fixed across all seasons	
DeePC initial-condition horizon T_{ini}	30 h
DeePC prediction horizon N	48 h
Comfort penalty weight Q_s	10^4
Pseudo-Random Binary Sequence (PRBS) data collection duration T_{collect}	1600 h
Closed-loop simulation duration T_{sim}	2160 h (90 days)
Heat pump coefficient of performance (COP)	3
Battery state-of-charge bounds	[0.2, 0.9]
Selection rule tolerance ε	0.05
Season-dependent	
Radiator power limit, March	70 W/m^2
Radiator power limit, June	30 W/m^2
Radiator power limit, September	70 W/m^2
Radiator power limit, December	90 W/m^2

Throughout this chapter, the recommended combination for a training building is defined by applying the selection rule of §3-5 directly to the measured sweep data. The combination with the lowest measured violation defines the minimum, all combinations within the $\varepsilon = 0.05$ tolerance band form the candidate set, and the cheapest candidate is selected.

The tolerance band exists because comfort violation alone is not a reliable enough signal to pick a single winner outright. Small differences in measured violation between two combinations can reflect noise in a single 90-day simulation run rather than a difference in tuning quality. Treating any combination within 5% of the grid minimum as equally good in comfort terms avoids selecting a combination whose apparent advantage is too small to trust, and lets cost break the tie among combinations that are functionally equivalent on the primary criterion. The same rule is later applied to the Gaussian process predictions, so the measured and predicted recommendations can be compared on equal terms.

4-1-1 Predictor Validation

Before presenting the parameter sweep results, the accuracy of the DeePC predictor itself is verified. Because DeePC makes predictions directly from the Hankel matrices built from collected input/output data, rather than from an explicit model of the building, this check

verifies that the data-driven Hankel representation accurately captures the dynamics of the true underlying building model. The predicted output $Y_f g$ is formed from the Hankel partition Eq. (2-2) and the decision variable g obtained by solving the DeePC optimization problem Eq. (3-6) at each time step. At every timestep of a closed-loop simulation, the predicted output $Y_f g$ is compared against the output subsequently realized by the true building model. The one-step-ahead prediction error is defined as

$$e_t = y_{\text{actual},t} - Y_f g \tag{4-1}$$

where $y_{\text{actual},t}$ is the room temperature realized by the true building model at time t and $Y_f g$ is the corresponding one-step-ahead prediction from the DeePC problem Eq. (3-6) solved at the previous time step. This error should remain small relative to the comfort bounds. A small error confirms that differences in comfort violation and cost observed across the regularization grid arise from tuning effects rather than from model mismatch.

This check is illustrated using Building 3, March, at its recommended combination $(\lambda_g, \lambda_y) = (10^4, 10^4)$, initialized inside the comfort band rather than from a cold start.

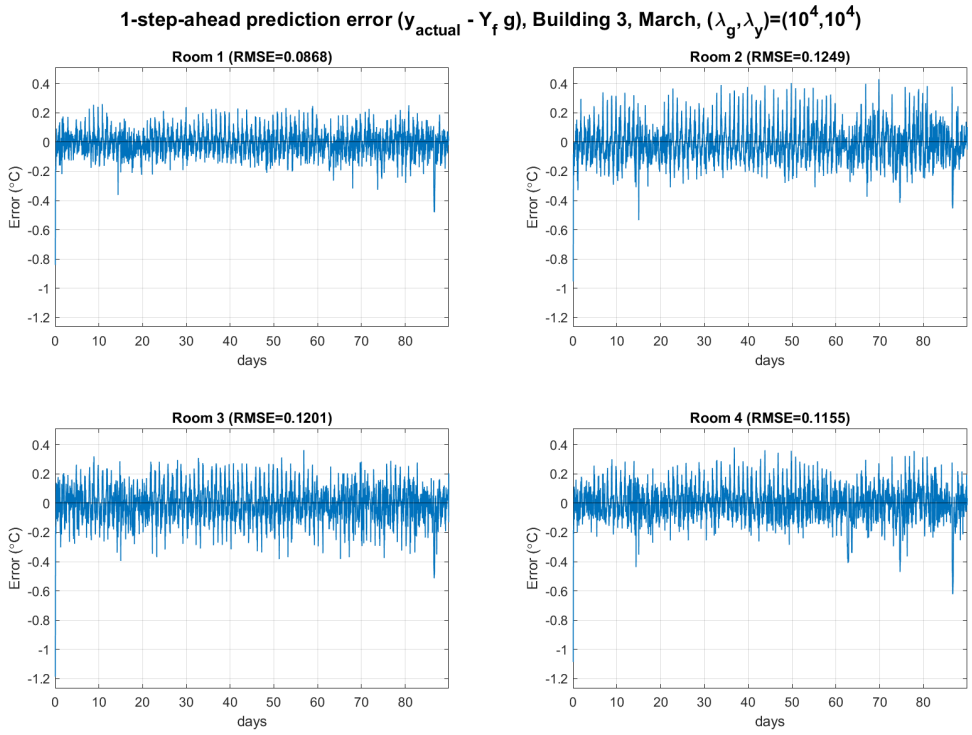


Figure 4-2: One-step-ahead prediction error, $y_{\text{actual}} - Y_f g$, per room, Building 3, March, recommended combination.

Figure 4-2 shows that the one-step-ahead prediction error remains small throughout the 90-day run, in all four rooms, consistent with the root-mean-square errors of 0.09–0.12 °C reported in Table 4-2. The one exception is a brief initial transient, during which the error is larger than during the remainder of the simulation before quickly settling close to zero. This transient is an expected consequence of how the simulation is initialized. The building's

state is set directly inside the occupied comfort constraints, but the initial condition trajectory $(u_{\text{ini}}, y_{\text{ini}})$ supplied to the DeePC problem comes from the last part of the historical data collected during the PRBS excitation phase. This historical data does not match the comfort-band starting point that was chosen. The predictor therefore assumes a recent history that does not match the true recent history of the simulated building, and the one-step-ahead prediction error is larger at the start as a direct result. As the closed-loop simulation continues, the sliding T_{ini}-length window is populated with actual input/output pairs from the running simulation itself, so the mismatch resolves and the prediction error drops toward zero. For the remainder of the 90-day run, the error oscillates around zero with small deviations from the actual temperatures in each of the rooms of Building 3.

Table 4-2 reports the root-mean-square prediction error per room over the full 90-day simulation.

Table 4-2: One-step-ahead prediction RMSE per room, Building 3, March, recommended combination $(\lambda_g, \lambda_y) = (10^4, 10^4)$.

Room	RMSE (°C)
Z1	0.0868
Z2	0.1249
Z3	0.1201
Z4	0.1155

Despite the initial transient discussed above, the root-mean-square error over the full 90-day run remains below 0.13°C in every room. This is less than 4% of the occupied-hours comfort range. This confirms that the Hankel-matrix representation accurately captures the true building dynamics. This root-mean-square value is not directly comparable to the mean violation metric Eq. (3-8) used elsewhere in this chapter. This root-mean-square value is calculated from the one-step-ahead prediction error e_t defined in (4-1). For each room z, it is calculated as the square root of the mean squared error across every hour of the 90-day run :

$$\text{RMSE}_z = \sqrt{\frac{1}{T_{\text{sim}}} \sum_{t=1}^{T_{\text{sim}}} e_{z,t}^2}, \tag{4-2}$$

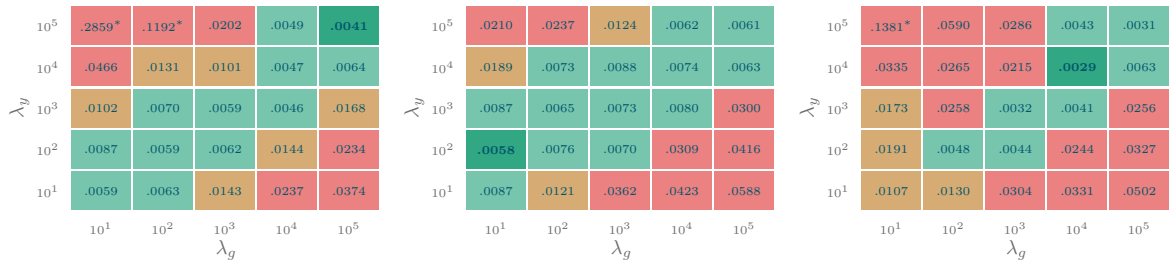
Squaring each error before averaging means that larger deviations, such as the initial transient, contribute more to RMSE_z than the same total error spread evenly across many hours would. An RMSE below 0.13°C in every room therefore means that, on a typical hour, the model's one-step-ahead prediction differs from the true room temperature by well under 0.13°C, and even the transient at the start of the simulation is not large or long enough to push this average close to that value. This confirms that the Hankel-matrix representation accurately captures the true building dynamics.

Three buildings, referred to as Building 1, Building 2, and Building 3, were used to train a separate Gaussian process model for each of the four seasons considered in this thesis: March, June, September, and December. The selection of these three buildings is described in §3-3-1. For each season, this section presents the parameter sweep results for each training building, the resulting Gaussian process fit, and a comparison of the fit across seasons. Beginning with

June, each season's discussion notes only the aspects that differ from the seasons already presented, rather than repeating unchanged content. Table 4-1 lists the concrete experimental parameters used to produce these results.

4-1-2 March

Training data for March consisted of 75 points: the full grid of 25 regularization combinations, evaluated in closed-loop simulation for each of the three training buildings. Building specifications are given in §3-3-1. Figure 4-3 shows the mean comfort violation, calculated using Eq. (3-8), across the grid for each building.



(a) Building 1: $(\lambda_g, \lambda_y) = (10^5, 10^5)$.

(b) Building 2: $(\lambda_g, \lambda_y) = (10, 10^2)$.

(c) Building 3: $(\lambda_g, \lambda_y) = (10^4, 10^4)$.

■ <0.010 ■ 0.010–0.020 ■ ≥0.020 * removed as an outlier before fitting

Figure 4-3: Mean comfort violation ($°C \cdot$ room/hr) across the 5×5 regularization grid for the three training buildings, March. Bolded cell in each panel marks the recommended combination for that building. A cell at exactly 0.020 belongs to the red tier. Asterisked cells were removed as outliers before fitting the GP models.

Effects on Tuning The three buildings exhibit different tuning landscapes, yet all three share one common pattern. In every building, the diagonal of the grid, where λ_g and λ_y are close in magnitude, forms a well-behaved region of low violation. Building 1 and Building 2 both show more low-violation combinations than high-violation ones. Building 1 contains twelve green-tier combinations against seven red-tier combinations, and Building 2 contains fourteen green-tier combinations against eight red-tier combinations. Building 3 shows a different pattern. Thirteen of its twenty-five combinations fall in the red tier, a much larger fraction than in either of the other two buildings. This contrast illustrates how differently the three buildings respond to the same regularization grid.

A direct comparison across buildings makes this difference concrete. Building 2's own recommended combination, $(10, 10^2)$, produces an amber-tier violation of $0.0191\,°C \cdot$ room/hr when applied to Building 3. The same combination produces a green-tier violation of $0.0087\,°C \cdot$ room/hr when applied to Building 1. This value is still more than double Building 1's own recommended violation of $0.0041\,°C \cdot$ room/hr. A combination that performs well for one building can therefore perform noticeably worse on another, even when the result still falls in the same comfort tier. This is the core problem this thesis addresses: there is no single combination that works well for every building, so each building requires its own tuning.

This difference matters when a combination is applied to a different building than it was obtained for. Building 2's combination produces a violation more than double Building 1's own recommended value, and an amber-tier result on Building 3 against Building 3's own green-tier value of $0.0029\,^{\circ}\mathrm{C}\cdot\mathrm{room/hr}$. In both cases, Building 2's combination is not close to the best available for the other two buildings.

The reverse comparison shows a different pattern. Building 1's combination, $(10^5, 10^5)$, produces a violation of $0.0061\,^{\circ}\mathrm{C}\cdot\mathrm{room/hr}$ in Building 2 (Figure 4-3b), close to Building 2's own recommended violation of $0.0058\,^{\circ}\mathrm{C}\cdot\mathrm{room/hr}$. Building 3's combination, $(10^4, 10^4)$, produces a violation of $0.0074\,^{\circ}\mathrm{C}\cdot\mathrm{room/hr}$ in Building 2, also close to Building 2's own recommended value. Both combinations remain in the green tier for Building 2, only marginally worse than its own recommended combination.

This asymmetry has a direct consequence for the tuning problem. Applying Building 1's or Building 3's combination to Building 2 still gives acceptable performance, remaining in the green tier. Applying Building 2's combination to Building 1 or Building 3 does not: it more than doubles Building 1's violation and pushes Building 3 into the amber tier. A single fixed combination, chosen to be safe for one building, is not safe for all three. This is the practical reason a per-building tuning approach is needed, rather than one combination applied to every building regardless of its physical properties.

Table 4-3: Recommended regularization combination per building, March, $\varepsilon = 0.05$ selection rule applied to the measured sweep data.

Building	λ_g	λ_y	Mean violation ($^{\circ}\mathrm{C}\cdot\mathrm{room/hr}$)	Cost (EUR)
Building 1	10^5	10^5	0.0041	1484.6
Building 2	10	10^2	0.0058	1342.7
Building 3	10^4	10^4	0.0029	925.6

Table 4-3 also reports the realized 90-day energy cost at each building's recommended combination. The raw cost increases with building size: Building 3 ($288\,\mathrm{m}^2$) costs 925.6 EUR, Building 2 ($420\,\mathrm{m}^2$) costs 1342.7 EUR, and Building 1 ($504\,\mathrm{m}^2$) costs 1484.6 EUR.

Effects on the Gaussian Process These same 75 points are also the training data for this season's Gaussian process models. The optimal combination depends on more than a building's physical description. As stated in §3-2-1, the best values of λ_g and λ_y also depend on how well-conditioned a building's Hankel matrices are. This property comes from how the data collection was performed for that specific building, not from the building's physical description. Two buildings with the same floor area and time constant can still require different regularization, if their collected data differ in how well it excites the building's dynamics.

Building 2 illustrates this directly. Its physical features sit mostly between Building 1's and Building 3's (Table 3-1), while its recommended regularization is the most extreme of the three. A mapping from physical features to regularization that varied smoothly with size or insulation would place Building 2's optimum between Building 1's and Building 3's. The actual function instead departs from that pattern at the point where Building 2 sits. This gap, between what a building's physical description would suggest and what its data requires, is the reason this thesis learns the λ_g, λ_y mapping from measured data rather than deriving it from a formula based on physical features alone.

The heatmaps also show that each building's well-behaved diagonal region sits at a different location and falls off differently away from it, most visibly in Building 3's more irregular pattern compared to Building 1's and Building 2's. The Gaussian process fitted to this data reproduces both the shared diagonal structure and each building's individual departure from it, as shown by the fit quality reported in §4-1-2. Section 4-2 extends this check to buildings outside the training set.

Controller Behavior at Recommended Combinations

Figures 4-4–4-6 show the zone temperature and radiator input trajectories for the three training buildings, each at its own recommended combination, over the full 90-day simulation.

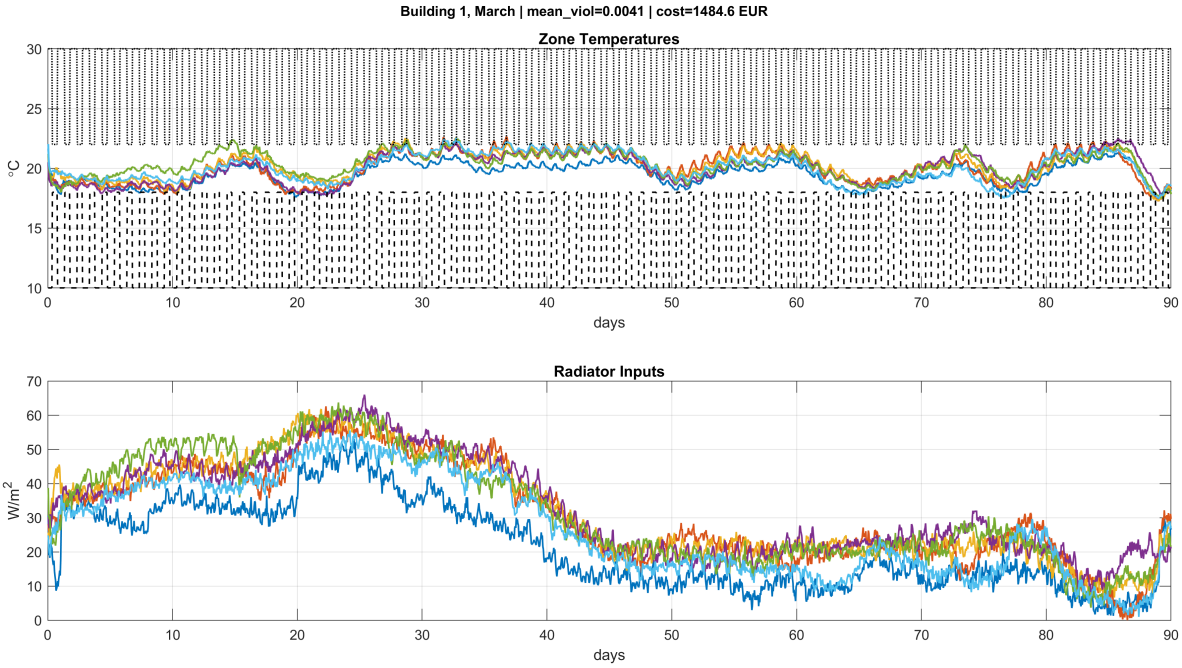


Figure 4-4: Zone temperatures and radiator inputs, Building 1, March, at the recommended combination $(\lambda_g, \lambda_y) = (10^5, 10^5)$. Zones are shown in MATLAB's default color order: Z1 blue, Z2 orange, Z3 yellow, Z4 purple, Z5 green, Z6 cyan.

Building 1 holds zone temperatures closely within the occupied-hours comfort band throughout the 90-day simulation (Figure 4-4), consistent with its low mean violation reported in Table 4-3. Temperatures remain close to the tighter daytime band even during the loose overnight window (10°C–30°C), rather than drifting toward its cheaper extremes. Radiator input rises to a peak around day 20–25 and recedes to a trough around day 45–60, tracking the building's seasonal heating demand. When zone temperatures approach the upper bound, radiator input visibly starts to decrease, but the high regularization on λ_g prevents it from dropping aggressively. The resulting radiator trajectory is smoother in trend than a combination with lower regularization would produce, though it is not free of hour-to-hour noise.

This noise is consistent with the structure of the DeePC problem Eq. (3-6) itself. The equality constraint $U_p g = u_{\text{ini}}$ is enforced exactly at every solve, but g has far more entries than the

building's true degrees of freedom, so many different g vectors satisfy this constraint for any given u_{ini}. At high λ_g, the $\|g\|^2$ term dominates the objective, so the solver effectively selects the minimum-norm g satisfying the constraints rather than one shaped primarily by the comfort and cost terms. Because u_{ini} is built from the real, previously applied input and real measurements, it changes somewhat from one hour to the next, and a small change in this constraint can shift which minimum-norm solution satisfies it by more than the change itself would suggest. At low λ_g, this pressure toward a minimum-norm solution is weak, so the comfort and cost terms, which evolve smoothly and on a fixed daily schedule, dominate the choice of g instead. This is offered as a plausible mechanism grounded in the structure of the optimization problem, not one directly verified against the solver's intermediate output. One zone, Z1 (blue), runs consistently cooler than the other five and requires consistently less radiator input, visible throughout roughly days 5–40.

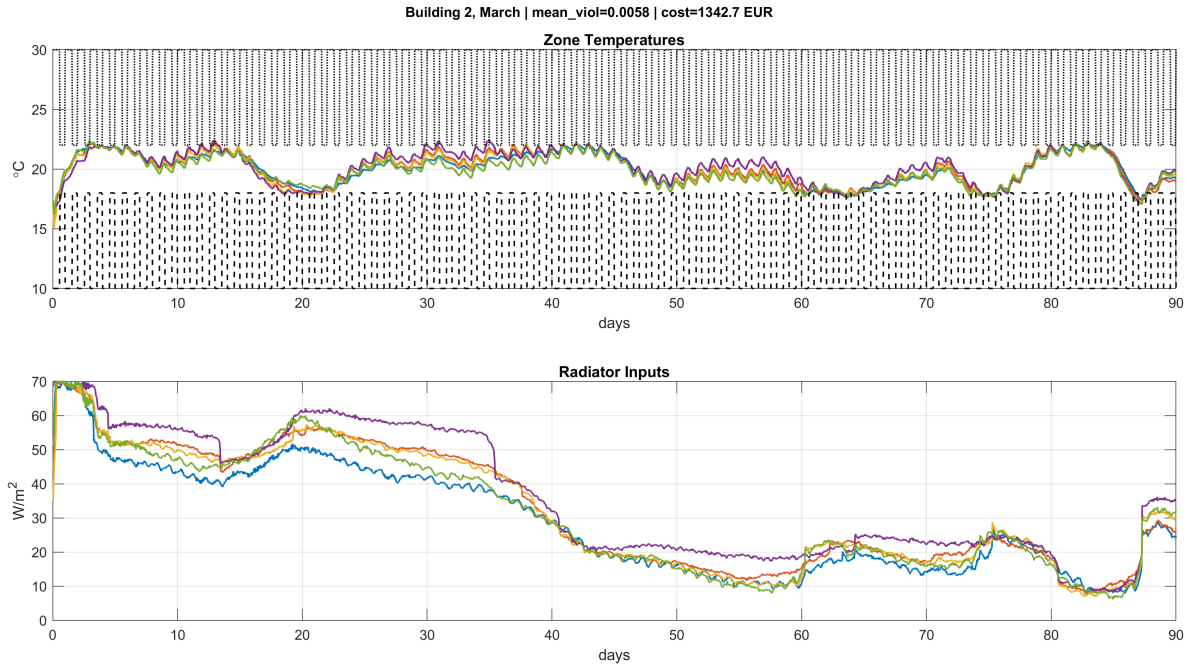


Figure 4-5: Zone temperatures and radiator inputs, Building 2, March, at the recommended combination $(\lambda_g, \lambda_y) = (10, 10^2)$. Zones are shown in MATLAB's default color order: Z1 blue, Z2 orange, Z3 yellow, Z4 purple, Z5 green.

Building 2 also holds zone temperatures closely within the occupied-hours comfort band (Figure 4-5), again remaining close to the tighter daytime band overnight rather than drifting toward the loose bounds. Radiator input follows the same seasonal heating curve as Building 1, peaking around day 20–25 and receding around day 45–60, and saturates at the 70 W/m^2 March radiator limit for the first four days.

Building 2's regularization on λ_g is far lower than Building 1's, and this is visible directly in the radiator input. Rather than the continuous hour-to-hour noise seen in Building 1, the input here is calmer between transitions but changes sharply at the transitions themselves, with pronounced step-like drops and rises rather than a gradual climb or descent. This is consistent with the mechanism proposed above: at low λ_g, the pressure toward a minimum-norm g is weak, so the comfort and cost terms dominate the choice of input instead. Because the energy price switches on a fixed day/night schedule rather than varying continuously,

the input can track that schedule directly, producing sharp changes exactly where the price regime changes rather than smooth continuous variation.

Zone temperatures are also visibly more tightly grouped across the five zones than in Building 1, where the six zones spread out more from one another over the same period. One zone, Z4 (purple), runs consistently slightly warmer than the other four and takes consistently slightly more radiator input, a small but persistent offset visible across the full 90 days.

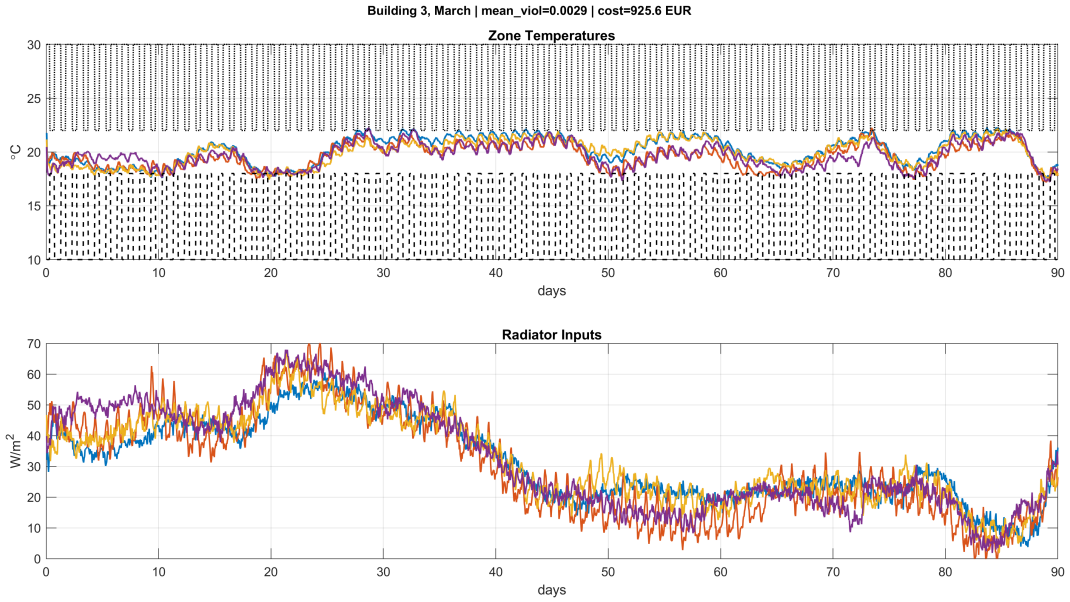


Figure 4-6: Zone temperatures and radiator inputs, Building 3, March, at the recommended combination $(\lambda_g, \lambda_y) = (10^4, 10^4)$. Zones are shown in MATLAB's default color order: Z1 blue, Z2 orange, Z3 yellow, Z4 purple.

Building 3 shows the same comfort-band behavior as the other two buildings (Figure 4-6), holding temperatures close to the daytime band even overnight. Radiator input follows the same seasonal curve, peaking around day 20–25 and receding around day 45–60. Despite using an intermediate regularization ($\lambda_g = 10^4$) between Building 1's (10^5) and Building 2's (10), radiator input here shows continuous high-frequency variation at least as pronounced as Building 1's, not intermediate in smoothness between the two. This is a third data point consistent with the earlier finding that radiator input smoothness does not follow a simple monotonic relationship with regularization strength.

Comparing the three buildings highlights one finding that is not visible from any single trajectory alone. Radiator input smoothness does not follow a simple monotonic relationship with regularization strength. Building 2 uses the lowest regularization of the three, yet its radiator input is the smoothest of the three buildings. Its input consists of long, flat segments separated by a small number of sharp, zone-synchronized step changes. Building 1 and Building 3, despite using higher regularization, show continuous high-frequency variation instead, with Building 3's input at least as pronounced as Building 1's even though its regularization sits between Building 1's and Building 2's. This is the opposite of what the smoothing role attributed to λ_g in Chapter 2 would suggest.

This is consistent with the mechanism discussed above for Building 1: smoothness depends on which side of a threshold in λ_g a building's regularization falls on, not on where it sits

numerically between two other buildings. Building 1 and Building 3 both use λ_g high enough to fall on the minimum-norm side of this threshold, which is why Building 3's input is not intermediate in smoothness between the other two.

Gaussian Process Fit

Outlier Removal Three of the 75 training points exhibited mean comfort violation that deteriorated the performance of the Gaussian Process GP. Two are from Building 1, at $(\lambda_g, \lambda_y) = (10, 10^5)$ and $(10^2, 10^5)$, and one is from Building 3, at $(10, 10^5)$. Building 2 contributed no points. These points are marked with an asterisk in Figure 4-3.

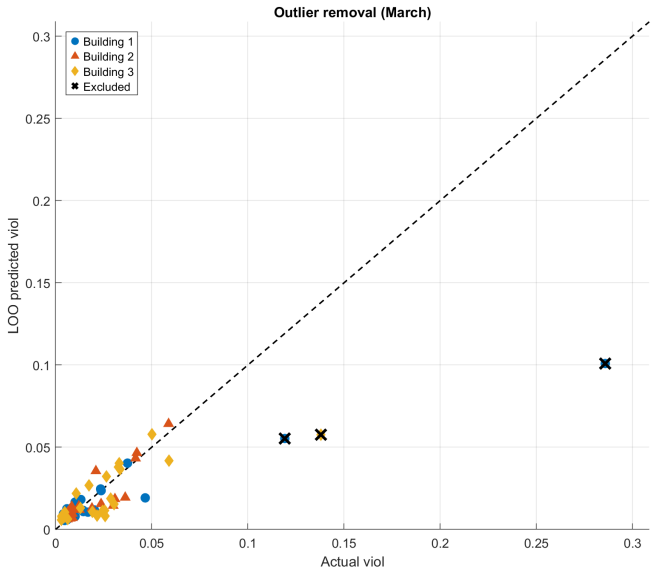


Figure 4-7: Leave-one-out predicted vs. actual mean violation, March, all 75 points. The three excluded outliers are marked with an X.

Figure 4-7 shows all 75 points together, before any point was removed. The three points later excluded are visibly separated from the rest of the grid: they sit at actual violations from roughly 0.12 to 0.29 °C · room/hr, while every other point falls below 0.06 °C · room/hr. This separation is what motivated removing them and checking whether the fit improved.

Table 4-4 confirms that it does.

Table 4-4: Leave-one-out RMSE for the March violation Gaussian process, with and without the three outlier points.

	Points	LOO RMSE (°C · room/hr)
With outliers	75	0.0253
Without outliers	72	0.0078

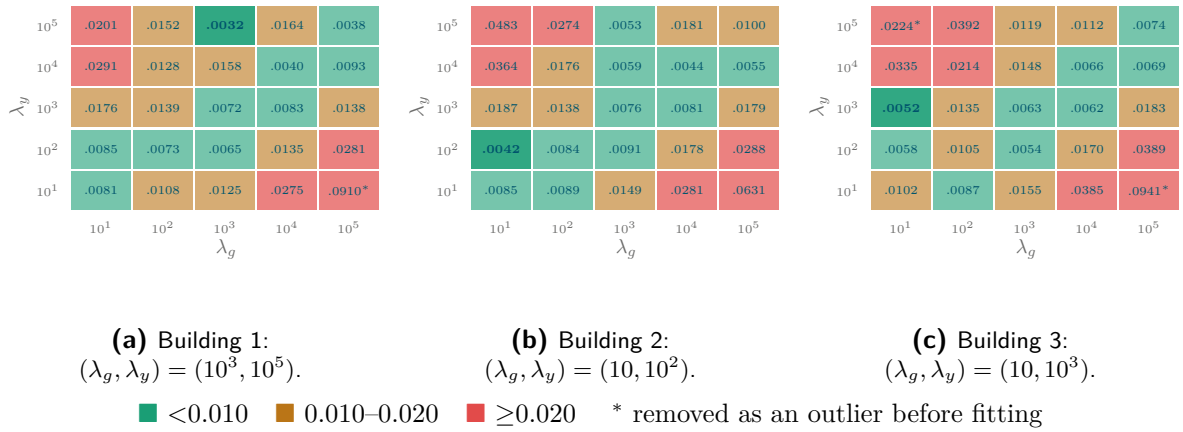
Whether an RMSE of 0.0253 is acceptable depends on what the model is used for. The comfort tiers used throughout this thesis are 0.010 °C · room/hr wide for amber (0.010–0.020 °C ·

room/hr) and defined by a red threshold at $0.020\,°\text{C} \cdot \text{room/hr}$. An RMSE of 0.0253 is larger than the entire amber tier and comparable to the red threshold itself. A model with error of this size cannot reliably tell a green-tier combination from an amber or red one, which undermines the selection rule's ability to rank combinations correctly. With the three points removed, the RMSE drops to 0.0078, comfortably below the narrowest tier width, so tier classification becomes something the fitted model can be trusted for.

With outliers removed, the March violation Gaussian process achieves a leave-one-out RMSE of 0.0078 on 72 training points. The fitted exponential kernel has lengthscale 2.2958, signal standard deviation 1.0157, and noise standard deviation 0.0307. The cost Gaussian process, used as a tiebreaker within the selection rule described in §3-5, achieves a leave-one-out RMSE of 30.87 EUR on the same 72 points, with fitted exponential kernel lengthscale 79.4907, signal standard deviation 0.4455, and noise standard deviation 0.0001. The cost Gaussian process's larger lengthscale indicates that energy cost varies far more smoothly across the regularization grid than comfort violation does. This is consistent with the selection rule treating violation as the primary criterion and cost only as a tiebreaker among near-tied candidates.

4-1-3 June

Training data for June again consisted of 75 points, 25 regularization combinations evaluated on each of the three training buildings.



(a) Building 1: $(\lambda_g, \lambda_y) = (10^3, 10^5)$.

(b) Building 2: $(\lambda_g, \lambda_y) = (10, 10^2)$.

(c) Building 3: $(\lambda_g, \lambda_y) = (10, 10^3)$.

■ <0.010 ■ 0.010–0.020 ■ ≥0.020 * removed as an outlier before fitting

Figure 4-8: Mean comfort violation ($°\text{C} \cdot \text{room/hr}$) across the 5×5 regularization grid for the three training buildings, June. Bolded cell in each panel marks the recommended combination for that building. A cell at exactly 0.020 belongs to the red tier. Asterisked cells were removed as outliers before fitting the GP models.

Effects on Tuning The three buildings' June landscapes differ from March, though one pattern continues unchanged. As in March, the diagonal of the grid, where λ_g and λ_y are close in magnitude, remains a well-behaved region of low violation in all three buildings, where almost every diagonal combination stays in the green tier for Building 1 and Building 2, and all but the two lowest-regularization diagonal points remain green for Building 3.

Building 2's recommended combination is unchanged from March, still $(10, 10^2)$. It is the only one of the three buildings whose recommendation does not shift between these two

seasons. Building 1's recommendation moves from the highest-regularization corner in March, $(10^5, 10^5)$, to $(10^3, 10^5)$: λ_y remains maximal, but the required λ_g drops by two orders of magnitude. Building 3 shows the largest shift of the three: its recommendation drops from $(10^4, 10^4)$ in March to $(10, 10^3)$ in June, needing significantly lower regularization on both parameters.

The overall number of red-tier combinations also falls from March to June. Building 1 drops from seven red-tier combinations to five, Building 2 from eight to six, and Building 3 from thirteen to seven. Combined across all three buildings, red-tier combinations fall from twenty-eight of seventy-five in March to eighteen of seventy-five in June. This matters for how much a wrong recommendation costs. When more of the grid falls in the red tier, choosing the wrong combination is more likely to produce abad result. When fewer combinations are red, as in June, a less-than-optimal recommendation is more likely to still land in the green or amber tier rather than fail outright.

Table 4-5: Recommended regularization combination per building, June, $\varepsilon = 0.05$ selection rule applied to the measured sweep data.

Building	λ_g	λ_y	Mean violation (°C · room/hr)	Cost (EUR)
Building 1	10^3	10^5	0.0032	481.8
Building 2	10	10^2	0.0042	400.7
Building 3	10	10^3	0.0052	315.1

Table 4-5 also shows a drop in energy cost compared to March, for all three buildings: 481.8 EUR versus 1484.6 EUR for Building 1, 400.7 EUR versus 1342.7 EUR for Building 2, and 315.1 EUR versus 925.6 EUR for Building 3. This makes sense given that June is a warmer period than March, requiring less heating and therefore less energy expenditure.

This drop in overall cost changes the stakes of the selection rule's cost tiebreaker (§3-5). The tiebreaker chooses the cheapest candidate among combinations with near-tied predicted violation, so it operates on whatever absolute cost scale the season happens to have. At June's much lower cost level, the EUR difference between two near-tied candidates is smaller in absolute terms than the same relative difference would be in March, even before considering whether the candidates are otherwise identical. A tiebreaker decision therefore carries lower absolute financial consequence in a low-cost season like June than in a higher-cost season like March, independent of how accurate the underlying GP predictions are.

Effects on the Gaussian Process Building 3's June landscape remains, as in March, visibly scattered rather than forming a clean low-violation band. Its recommended combination sits in a narrow region between two considerably rougher neighboring combinations, rather than at the center of a broad well-behaved area the way Building 1's and Building 2's recommendations do. Since these same 75 points are the training data for June's Gaussian process models, this narrow, sharply-bounded optimum is part of what the fitted model has to reproduce for Building 3, a harder target than a broad, gently-varying region would be.

The fitted kernel reflects this directly (§4-1-3). June's violation Gaussian process has a shorter lengthscale, 1.7010, and a substantially higher noise standard deviation, 0.2038, than March's 2.2958 and 0.0307 respectively. A shorter lengthscale means the fitted function is allowed to

vary more sharply between neighboring grid points, and higher noise means the model treats more of the observed variation as unexplained rather than signal. Both are consistent with June's violation surface being intrinsically less smooth than March's, independent of the point-removal step performed beforehand.

Building 3's narrow optimum shows up directly in the model's own recommendation check (§4-1-3). The model recommends $(10, 10^2)$ rather than the true optimum $(10, 10^3)$, a gap of 11.7%. Querying both candidates directly gives predicted violations of 0.006170 and 0.006125. Both values fall within the selection rule's 5% tolerance band of the grid's predicted minimum, so the two combinations are a near-tie in the underlying data. The cost tiebreaker activates as designed, correctly choosing the cheaper of the two, 298.23 EUR against 315.09 EUR. The consequence is small, a difference of only 0.0006 °C·room/hr in realized violation and a saving of about 17 EUR over the 90-day horizon, but the case illustrates concretely how a narrow, sharply-bounded region in the training landscape can produce a close call in the model's own recommendation, resolved correctly here by the tiebreaker rather than by the violation prediction alone.

Controller Behavior at Recommended Combinations

As discussed in §4-1-3, June's recommended combinations use lower regularization on λ_g than March across all three buildings, most sharply for Building 3. This lower regularization is visible directly in the trajectories that follow. June also uses a lower radiator power limit than March, 30 W/m^2 rather than 70 W/m^2 (Table 4-1), reflecting the lower heating demand of the warmer month. Figures 4-9–4-11 show the zone temperature and radiator input trajectories for the three training buildings, each at its own recommended combination, over the full 90-day simulation.

Building 1 holds zone temperatures closely within the occupied-hours comfort band (Figure 4-9), consistent with the low mean violation reported in Table 4-5. Radiator input drops to near zero for an extended period, approximately days 43–58, consistent with a stretch of warm weather in the simulated June data reducing heating demand to almost nothing. Radiator input shows sharp, step-like transitions at the start and end of this quiet period, rather than a gradual rise and fall.

Building 1's regularization on λ_g is two orders of magnitude lower in June (10^3) than in March (10^5), and this is visible in the radiator input. Within each of the four periods separated by the sharp transitions, the six zones settle into distinct, roughly flat bands, each zone holding a consistent position relative to the others, rather than the continuously crossing, harder-to-separate lines seen for this same building in March. Small hour-to-hour changes are still visible within each band, so the input has not become as smooth as Building 2's in March, but it sits between March's two extremes, consistent with $\lambda_g = 10^3$ falling between Building 2's 10 and Building 1's own March value of 10^5 on the same minimum-norm mechanism discussed for Building 1 in §4-1-2, grounded in the equality constraint $U_p g = u_{\text{ini}}$ within Eq. (3-6).

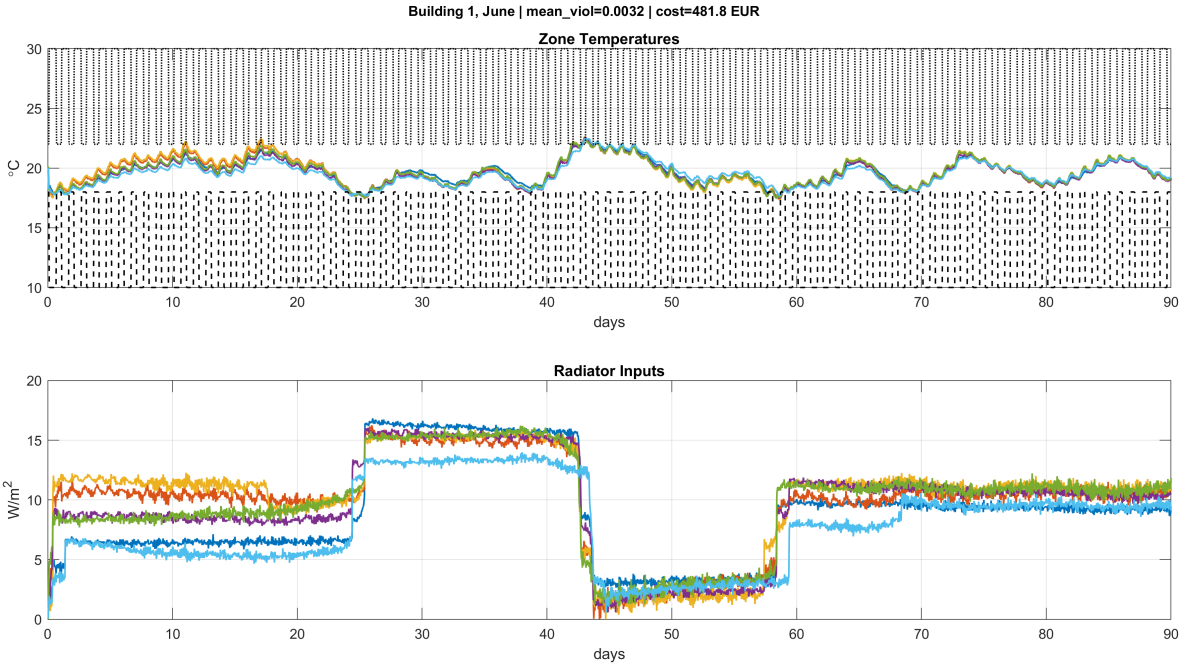


Figure 4-9: Zone temperatures and radiator inputs, Building 1, June, at the recommended combination $(\lambda_g, \lambda_y) = (10^3, 10^5)$.

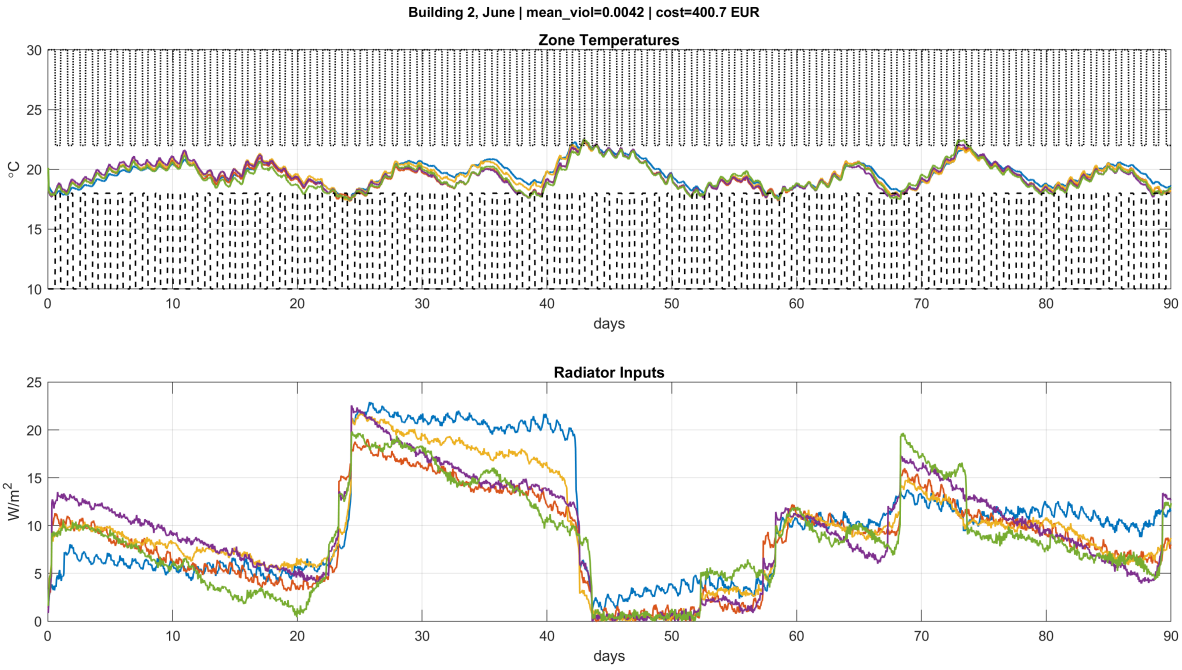


Figure 4-10: Zone temperatures and radiator inputs, Building 2, June, at the recommended combination $(\lambda_g, \lambda_y) = (10, 10^2)$.

Building 2 also holds zone temperatures closely within the occupied-hours comfort band (Figure 4-10), and shows the same near-zero radiator input during a quiet period, approximately days 43–52, somewhat shorter than Building 1's. As in Building 1, the transitions into and

out of this quiet period are sharp and step-like rather than gradual.

Building 2's recommended combination is unchanged between March and June, $(10, 10^2)$ in both seasons, yet the radiator input's profile is completely different from March. Since the regularization is identical, this difference comes from the disturbances the controller is responding to, not from the tuning. During this stretch, ambient temperature and solar gains are high enough that passive heat entering the building keeps zone temperatures close to the upper comfort bound on their own, with little or no radiator input needed to stay within bounds. Once passive gains are enough on their own, the cost-minimizing choice is to heat as little as possible, which is consistent with input dropping toward zero rather than easing off gradually. The same combination therefore produces a near-flat, low-cost input profile in this warm stretch and the more active profile seen in the surrounding days, purely as a function of the weather the controller is reacting to.

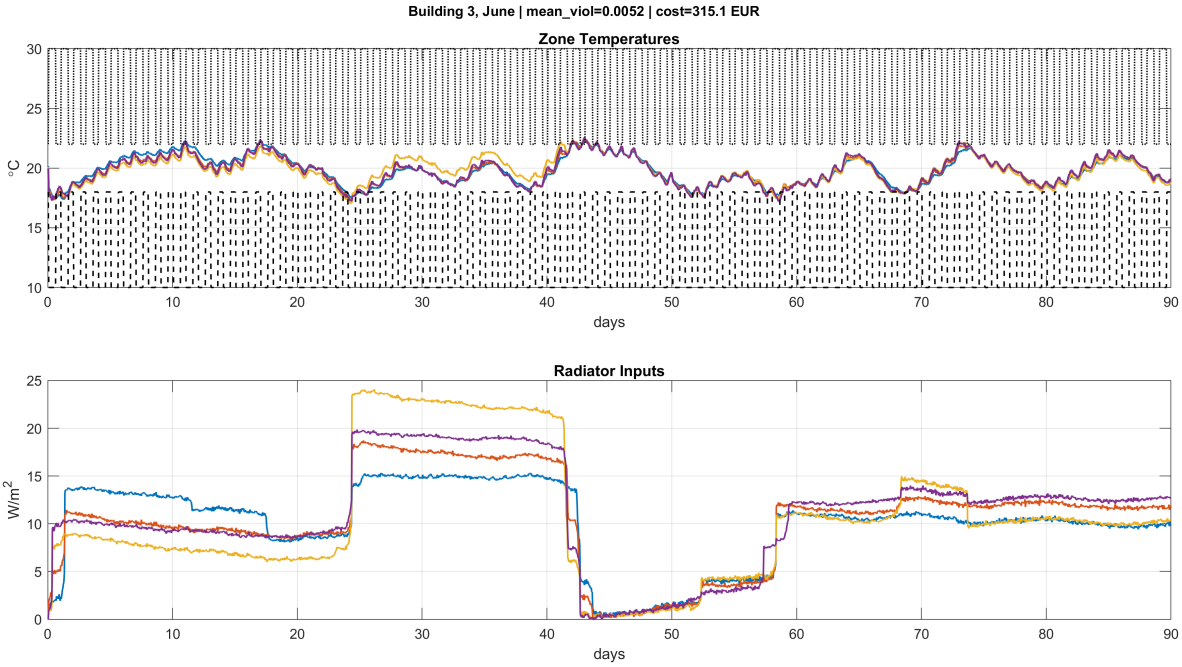


Figure 4-11: Zone temperatures and radiator inputs, Building 3, June, at the recommended combination $(\lambda_g, \lambda_y) = (10, 10^3)$.

Building 3 shows the same comfort-band behavior and the same near-zero radiator period, approximately days 43–52, matching Building 2 rather than Building 1's slightly longer trough (Figure 4-11). As in the other two buildings, the transitions into and out of the quiet period are sharp rather than gradual.

The near-zero radiator period is therefore a shared feature not present in March, though its exact duration differs slightly by building. All three buildings show sharp, synchronized step transitions at the boundaries of this period, despite using very different regularization strengths (10^3, 10, and 10 for Building 1, Building 2, and Building 3 respectively). Since this pattern appears regardless of regularization strength, it more plausibly reflects a shared feature of the underlying June weather data, such as an abrupt onset and end of a warm spell, than an effect of the controller's tuning. This has not been verified against the underlying weather data.

Gaussian Process Fit

Point Removal No point in the June dataset reached the extreme raw violation values seen in March. Three points diverge visibly from the diagonal in an initial leave-one-out fit on all 75 points, shown in Figure 4-12 and marked with an asterisk in Figure 4-8: $(10^5, 10)$ for Building 1 (actual 0.0910), $(10, 10^5)$ for Building 3 (actual 0.0224), and $(10^5, 10)$ for Building 3 (actual 0.0941). Two of these, Building 1 at 0.0910 and Building 3 at 0.0941, are also the two highest raw violation values recorded anywhere in the June dataset, though both are well below March's most extreme value of 0.286. The third, Building 3 at 0.0224, is not itself an elevated raw value; it stands out only because the model's prediction for it, roughly 0.055, diverges sharply from the true value.

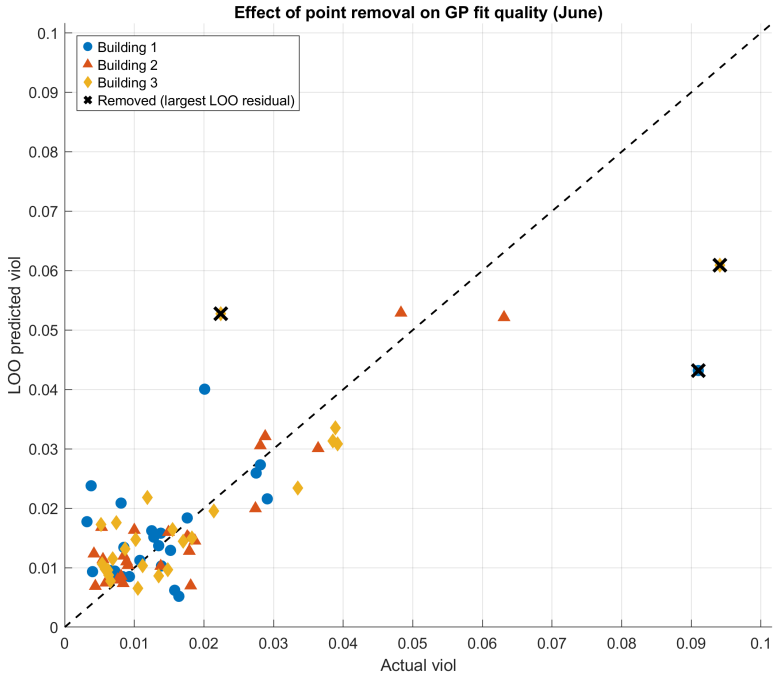


Figure 4-12: Leave-one-out predicted vs. actual mean violation, June, all 75 points. The three points later removed, identified by the largest leave-one-out residual under an initial fit, are marked with an X.

All three sit at the extreme off-diagonal corners of the grid, where λ_g and λ_y are as far apart as possible on the log scale. This is consistent with the diagonal pattern ($\lambda_g \approx \lambda_y$) observed as the well-behaved region across every building and season in this thesis: at these extreme corners, one regularization weight is essentially unconstrained while the other completely dominates, an imbalance not present anywhere else on the grid. The resulting closed-loop behavior at these points is less consistent with the smoother trend followed by their neighbors, producing a violation value the exponential kernel cannot represent as well as it does elsewhere, and inflating the leave-one-out residual. These three points were removed and the model refit on the remaining 72.

The way these points were identified differs from how March's outliers were found. In March, the three removed points were separated from the rest of the grid by their raw violation values alone, visible directly without needing to fit a model first. Here, only two of the three points are elevated in raw value, and neither is separated from the rest of the grid as clearly as March's outliers were; the third is not elevated in raw value at all. All three were instead identified by the leave-one-out residual of an initial fit, a criterion that depends on already having a fitted model. This criterion is, by construction, guaranteed to improve the reported leave-one-out error, since the removed points are precisely those the metric penalizes most heavily.

Table 4-6 shows the effect: removing these three points reduces the leave-one-out RMSE from 0.0101 to 0.0067 °C · room/hr. Both values are small in absolute temperature terms, but the relevant comparison is to the size of the violation values themselves, which range from about 0.003 to 0.09 °C · room/hr across the training grid. A prediction error of 0.0101 °C · room/hr is therefore large relative to the differences that separate a good combination from a poor

one, even though it looks small on its own. This matters because the selection rule ranks combinations by predicted violation: if the model's error is comparable in size to the true gap between two candidates, the model can rank them incorrectly, even while every individual prediction stays numerically close to the true value. Removing these three points reduces this risk. This improvement should be read as a description of how much these three specific points were degrading the fit, rather than as independent evidence of overall model quality. Removing these three points also does not correct the one recommendation that is inexact in June (Building 3, discussed below). The two issues occur in different regions of the grid and are unrelated.

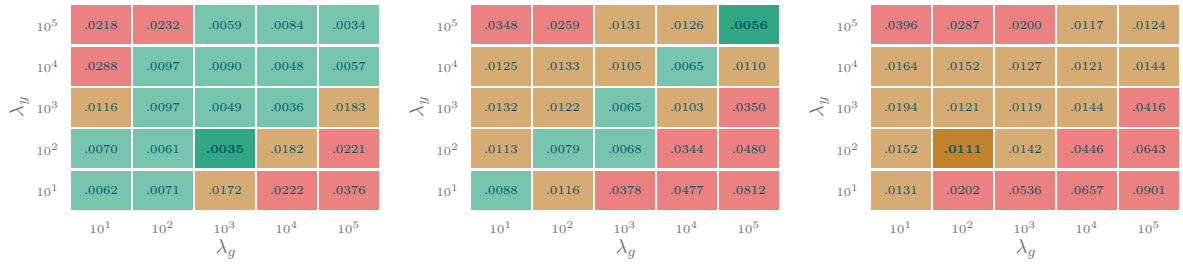
Table 4-6: Leave-one-out RMSE for the June violation Gaussian process, before and after removal of the three points.

	Points	LOO RMSE (°C · room/hr)
All points	75	0.0101
After removal	72	0.0067

With these three points removed, the June violation Gaussian process achieves a leave-one-out RMSE of 0.0067 on 72 training points. The fitted exponential kernel has lengthscale 1.7010, signal standard deviation 0.7782, and noise standard deviation 0.2038. Both the shorter lengthscale and the higher noise standard deviation, compared to March's 2.2958 and 0.0307 respectively, indicate that June's violation surface is intrinsically less smooth than March's, independent of the point-removal decision above. The cost Gaussian process achieves a leave-one-out RMSE of 33.60 EUR on the same 72 points, with fitted exponential kernel lengthscale 20.3403, signal standard deviation 0.4501, and noise standard deviation 0.0001.

4-1-4 September

Training data for September again consisted of 75 points, 25 regularization combinations evaluated on each of the three training buildings.



(a) Building 1: $(\lambda_g, \lambda_y) = (10^3, 10^2)$.

(b) Building 2: $(\lambda_g, \lambda_y) = (10^5, 10^5)$.

(c) Building 3: $(\lambda_g, \lambda_y) = (10^2, 10^2)$.

≤0.010 0.010–0.020 ≥0.020

Figure 4-13: Mean comfort violation (°C · room/hr) across the 5×5 regularization grid for the three training buildings, September. Bolded cell in each panel marks the recommended combination for that building. A cell at exactly 0.010 belongs to the green tier; a cell at exactly 0.020 belongs to the red tier.

Effects on Tuning Because the September simulation window spans both late-summer and late-autumn conditions, a landscape resembling some blend of March and June might be expected. The actual landscape is instead different from both. March's simulation window ends in late spring, and June's sits entirely in summer, but September's window runs through to the end of November, well into late autumn. This is consistent with Building 1 and Building 2's radiator input trending upward through the second half of the simulation and approaching the $70\,\mathrm{W/m^2}$ limit by around day 80 (§4-1-4): the simulated period grows ly cold by its end, unlike March or June.

Building 2 and Building 3 both show a sharp increase in amber-tier combinations compared to March: from 3 to 11 for Building 2, and from 4 to 15 for Building 3. This comes almost entirely at the expense of the green tier rather than a shift into the red tier. The red-tier count itself changes little, from 8 to 8 for Building 2 and from 13 to 10 for Building 3. Building 3's green tier disappears entirely: no combination in its 25-point grid falls below $0.010\,^\circ\mathrm{C}\cdot\mathrm{room/hr}$. Its best achievable violation, 0.0111, sits just above the green threshold, at the good end of the amber tier. Checking the four grid neighbors of this optimum shows no evidence that a better combination lies outside the tested range. Violation is worse in every adjacent direction. The two directions in which violation does trend downward toward a grid edge, low λ_g at $\lambda_y = 10$ and high λ_g at high λ_y, both bottom out at values still worse than the interior optimum. The grid therefore contains Building 3's best achievable result for September, even though no tested combination reaches the green tier.

Building 3's September grid also has the narrowest spread between its best and worst outcome of any training building this season. The ratio between its highest and lowest violation is approximately 8.1× (0.0901 at $(10^5, 10)$ versus 0.0111 at its optimum), compared to 11.1× for Building 1 and 14.5× for Building 2. This narrow spread is a further expression of the same absence of a green-tier combination: no combination in the grid performs substantially better than the rest, so tuning has correspondingly less room to improve outcomes for this building in September than it does for Building 1 or Building 2.

This shift toward the amber tier is consistent with September's simulation window covering two different conditions rather than one, as discussed in §4-1-4: radiator demand in Building 1 and Building 2 rises through the second half of the window, reaching the heating limit by around day 80. A single combination has to work reasonably well in both the milder first half and the colder second half. Fewer combinations achieve the low violation needed for green under these two conditions, even though most still avoid the poor performance of the red tier.

Building 2's own pattern reverses entirely from March and June. Both prior seasons recommended combination was low regularization, $(10, 10^2)$, but September's recommended combination is $(10^5, 10^5)$, the maximum available on both parameters. Applying Building 2's March/June combination directly in September gives a violation of $0.0113\,^\circ\mathrm{C}\cdot\mathrm{room/hr}$, amber rather than green, confirming this is a reversal rather than a marginal shift. This also fits the same pattern: a low-regularization combination tuned for one steady climate does not have one steady climate to fit in September. Higher regularization gives smoother, more cautious control, which does not track either half of the window as tightly, but performs more evenly across both.

September's cost does not fall between June's and March's, as a simple reading of the season names might suggest. All three buildings incur a higher cost in September than in March: 1822.5 EUR versus 1484.6 EUR for Building 1, 1661.7 EUR versus 1342.7 EUR for Building 2,

Table 4-7: Recommended regularization combination per building, September, $\varepsilon = 0.05$ selection rule applied to the measured sweep data.

Building	λ_g	λ_y	Mean violation (°C · room/hr)	Cost (EUR)
Building 1	10^3	10^2	0.0035	1822.5
Building 2	10^5	10^5	0.0056	1661.7
Building 3	10^2	10^2	0.0111	1158.9

and 1158.9 EUR versus 925.6 EUR for Building 3. This is consistent with September functioning as a transition month rather than a uniformly mild one. As discussed in §4-1-4, radiator demand in Building 1 and Building 2 trends upward through the second half of the simulation, reaching the 70 W/m^2 limit by around day 80, indicating the simulated period grows colder as it progresses into early autumn. March's steady, settled cold produces a lower total heating cost than September's mix of a milder start and a colder finish.

This transition within a single simulation window also bears on the tuning problem itself. Each building's recommended combination is selected using one mean violation value computed over the full 90-day window. If conditions change sharply within that window, one combination has to work well across very different conditions, not just one steady condition. March's own radiator input also shows some change over its window, rising to a peak around day 20–25 and receding to a trough around day 45–60 (§4-1-2), so March is not perfectly steady either. September's shift is more pronounced: rather than rising and partly receding within the window, radiator demand in Building 1 and Building 2 trends upward through the entire second half, reaching the heating limit by around day 80 (§4-1-4), a larger and more one-directional change than March's rise and partial fall. A single (λ_g, λ_y) pair recommended for September must perform reasonably well across this larger shift in conditions, rather than being matched to a narrower range of conditions the way March's recommendation is. This offers a plausible reason why September's violation landscape is rougher than March's and why Building 3 has no green-tier combination at all (§4-1-4): a combination well suited to the early, milder half of the window may perform worse once conditions turn colder, and a combination well suited to the colder second half may perform worse during the earlier mild stretch, so no single combination fits the whole window as well as one fits a window with less change in conditions.

Effects on the Gaussian Process A narrow spread between best and worst violation also has a consequence for the fitted model. With the true differences between neighboring combinations smaller in absolute terms, those differences are also smaller relative to whatever noise or fitting error the Gaussian process carries. This makes it comparatively harder for the model to correctly rank which combination is truly best, since a given amount of prediction error represents a larger fraction of the actual gap between candidates than it would in a grid with a wider spread.

Building 1's own recommendation check shows this difficulty directly (§4-1-4). The model recommends $(10^5, 10^5)$ rather than the true optimum $(10^3, 10^2)$, predicting a gap of roughly 11% between the two candidates when the real gap in the measured data is only about 4.5%. This is not the model ranking the true optimum as clearly worse; both candidates' predicted violations are close, but the model's own uncertainty overstates how different they

are, which keeps the cost tiebreaker from activating on a comparison that the real data would have allowed. The consequence is small in absolute terms, since $(10^5, 10^5)$'s actual violation, $0.0034\,°\text{C} \cdot \text{room/hr}$, is marginally lower than $(10^3, 10^2)$'s, $0.0035\,°\text{C} \cdot \text{room/hr}$, at a cost of 1872.5 EUR against 1822.5 EUR. The mechanism, however, is a direct example of the ranking difficulty just described: with true gaps this narrow, the model's uncertainty is large enough to obscure which candidate isbetter.

Despite this, the fitted model's overall accuracy in September is not worse than in the other seasons. Its leave-one-out RMSE, 0.0070, is close to March's 0.0078 and June's 0.0067 (§4-1-4), achieved without removing any points, unlike June. The ranking difficulty described above affects specific close comparisons within the grid rather than the model's general ability to predict violation across it.

Controller Behavior at Recommended Combinations

As discussed in §4-1-4, September's simulation window spans a transition from milder conditions into early autumn, rather than one settled climate. This is reflected directly in the trajectories that follow. Figures 4-14–4-16 show the zone temperature and radiator input trajectories for the three training buildings, each at its own recommended combination, over the full 90-day simulation.

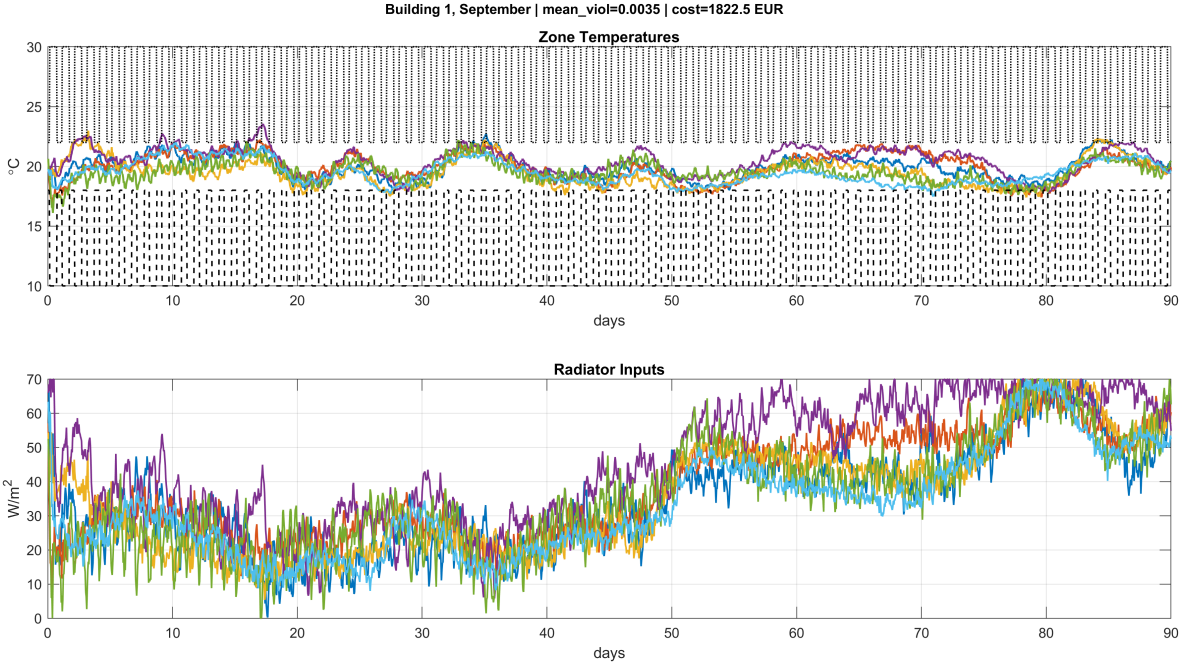


Figure 4-14: Zone temperatures and radiator inputs, Building 1, September, at the recommended combination $(\lambda_g, \lambda_y) = (10^3, 10^2)$.

Building 1 holds zone temperatures within the occupied-hours comfort band (Figure 4-14), consistent with the violation reported in Table 4-7. Unlike March and June, where radiator input traced a single seasonal hump peaking mid-simulation, radiator input here instead trends upward through the second half of the 90-day window, reaching the $70\,\text{W/m}^2$ limit

by around day 80. One zone, Z4 (purple), separates briefly from the other five at several points in the first third of the simulation, then becomes consistently the warmest zone, with correspondingly higher radiator input, through roughly the middle third of the window before the pattern loosens again near the end. Toward the end of the simulation, several zones approach the radiator limit at the same time rather than just one, indicating the building is operating close to its full heating capacity as the window ends, with little remaining headroom had the simulated period continued into colder conditions.

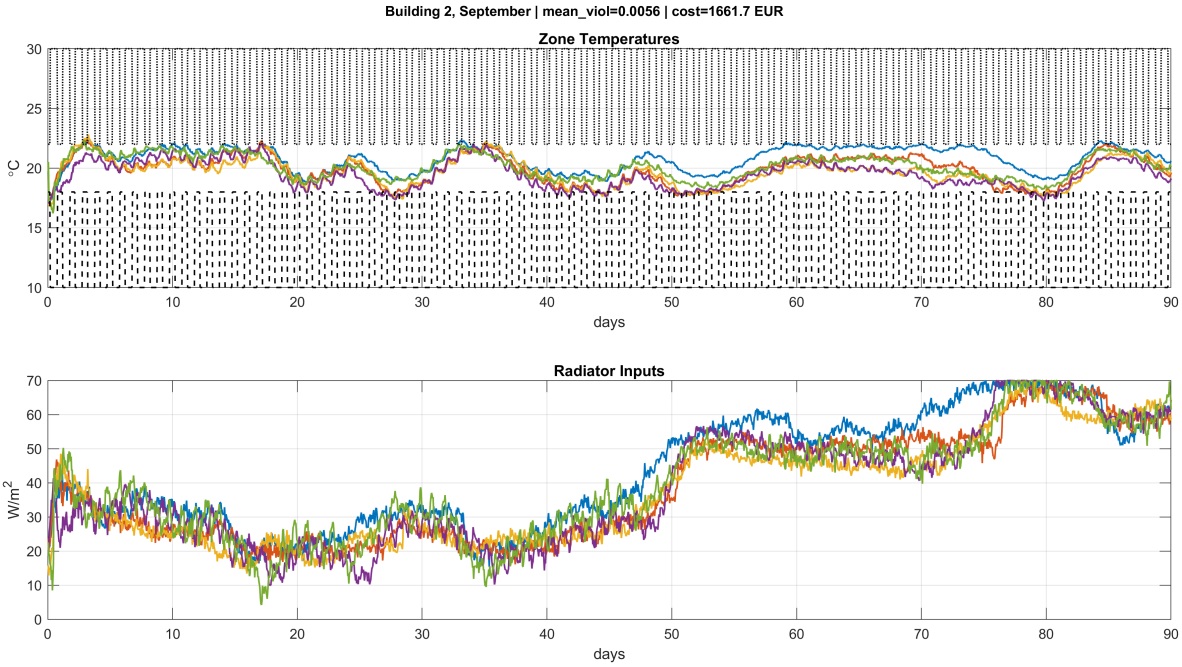


Figure 4-15: Zone temperatures and radiator inputs, Building 2, September, at the recommended combination $(\lambda_g, \lambda_y) = (10^5, 10^5)$.

Building 2 also holds zone temperatures within the comfort band (Figure 4-15), and shows the same second-half upward trend as Building 1, consistent with September transitioning from warmer conditions into early autumn as the simulated period progresses. The five zones do not rise together, however. One zone, Z1 (blue), separates from the others first, climbing from around day 47 and reaching the 70 $\mathrm{W/m^2}$ limit by around day 75, ahead of the rest. Past this point, Z1's input eases back down to roughly 50–60 $\mathrm{W/m^2}$, while the remaining four zones, still climbing through this period, only reach the limit later, around day 80–90, and remain there through the end of the window. Rather than all five zones converging on the limit at once, one zone leads the rise and recedes first, while the others catch up and are the ones still at the limit as the simulation ends.

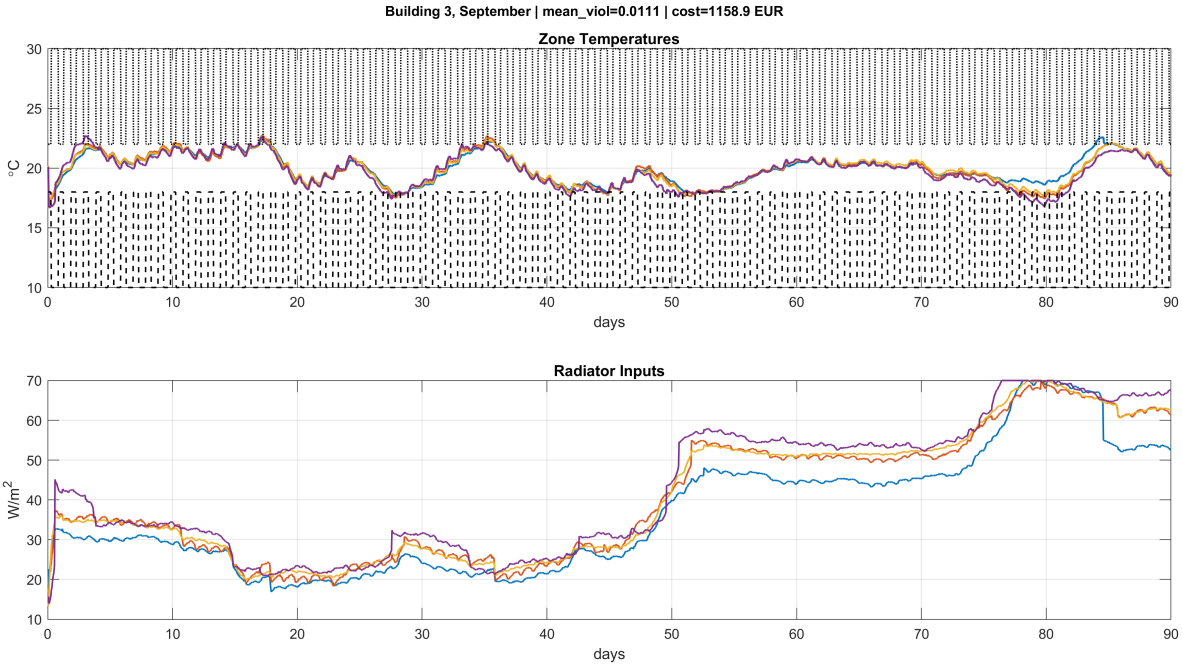


Figure 4-16: Zone temperatures and radiator inputs, Building 3, September, at the recommended combination $(\lambda_g, \lambda_y) = (10^2, 10^2)$.

Building 3 also holds zone temperatures within the comfort band (Figure 4-16). Its radiator input shows a similar overall rise to the other two buildings, but in discrete steps rather than a smooth trend, with plateaus around days 20–50 and 52–76 before a further rise near day 78. In the final days of the simulation, around day 85–90, one zone's radiator input, Z1, drops visibly below the other three, which remain near the limit.

The upward-trending radiator pattern common to all three buildings contrasts with the single mid-simulation hump seen in March and June. This is consistent with September capturing a transition from warmer to colder conditions within the simulation window itself, rather than a single settled season. The three buildings differ, however, in how that rise takes shape. Building 1 and Building 2 rise smoothly, while Building 3 rises in discrete plateaus. This repeats the same distinction already noted in its violation landscape (§4-1-4) between a settled pattern and a more irregular one.

This distinction is also consistent with the regularization each building uses. Building 3's recommended λ_g, 10^2, is the lowest of the three, below Building 1's 10^3 and well below Building 2's 10^5. This matches the mechanism proposed for Building 2's smooth, step-like March input (§4-1-2): at lower λ_g, the pressure toward a minimum-norm solution is weaker, so the comfort and cost terms have more influence over the input, and these terms change on a fixed daily schedule rather than continuously. Building 3's discrete plateaus in September are consistent with the same effect. Building 1 and Building 2 both use higher λ_g, consistent with a continuously rising input rather than one that moves in steps.

Toward the end of the simulation, both Building 1 and Building 2 have multiple zones reaching the $70\,\mathrm{W/m^2}$ radiator limit at once, with little headroom remaining as the window closes. This motivates using a higher radiator limit, $90\,\mathrm{W/m^2}$, for December (Table 4-1), since December's steady cold climate places demand on the buildings for the entire simulation, not only its final

days, and would be more likely to saturate a lower limit for an extended period rather than briefly at the end.

Gaussian Process Fit

Unlike June, no points were removed from the September training set. An initial fit on all 75 points showed no residuals as disproportionately large as those identified in June. Figure 4-17 shows the resulting fit.

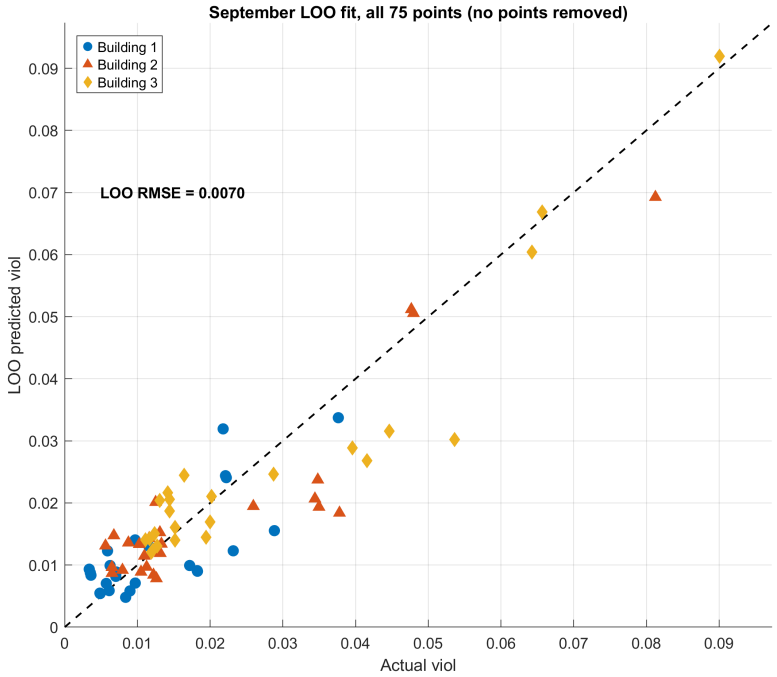


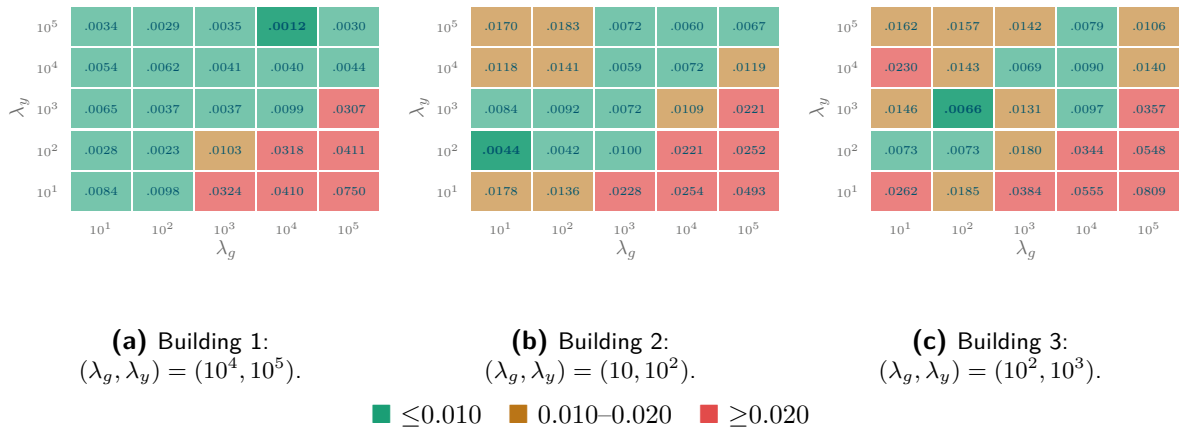
Figure 4-17: Leave-one-out predicted vs. actual mean violation, September, all 75 points. No points were removed.

The shape of the residual distribution itself differs between the two seasons. In June, three points stood out clearly above the rest, separated from the bulk of the distribution by a visible gap. In September, residuals decline smoothly from largest to smallest, with no such gap separating any subset of points from the rest, visible directly in Figure 4-17: every point sits at some distance from the diagonal, but no small group is set apart from the others. Any cutoff applied to a smoothly declining distribution is necessarily arbitrary, since there is no natural boundary between anomalous points and points that are somewhat harder to predict.

This distinction matters because removing training points is guaranteed to improve the reported leave-one-out RMSE by construction: a removed point can no longer contribute its own error to the metric. Applying June's removal criterion here, in the absence of a clear gap in the residual distribution, would have improved September's reported fit quality without a principled basis for selecting which points to remove. For this reason, the June removal is treated as a response to a specific feature of that season's data, and the same diagnostic led to no intervention in September.

4-1-5 December

Training data for December again consisted of 75 points, 25 regularization combinations evaluated on each of the three training buildings.



(a) Building 1: $(\lambda_g, \lambda_y) = (10^4, 10^5)$.

(b) Building 2: $(\lambda_g, \lambda_y) = (10, 10^2)$.

(c) Building 3: $(\lambda_g, \lambda_y) = (10^2, 10^3)$.

Figure 4-18: Mean comfort violation ($^\circ$C · room/hr) across the 5×5 regularization grid for the three training buildings, December. Bolded cell in each panel marks the recommended combination for that building. A cell at exactly 0.010 belongs to the green tier; a cell at exactly 0.020 belongs to the red tier.

Effects on Tuning The diagonal trend observed in every prior season mostly continues in December, but with one exception. Building 1's diagonal remains entirely in the green tier, and Building 2's diagonal is green at every point except its lowest-regularization corner. Building 3's diagonal, by contrast, is mixed: red at $(10, 10)$, green at $(10^2, 10^2)$, amber at $(10^3, 10^3)$, green at $(10^4, 10^4)$, and amber again at $(10^5, 10^5)$. December is the first season in which the diagonal does not represent a broadly reliable region for Building 3.

Tier composition shifts across all four seasons, not just between March and December. Building 1 improves fairly steadily: twelve green-tier and seven red-tier combinations in March, ten green and five red in June, fifteen green and six red in September, and eighteen green and six red in December. Building 2 and Building 3 instead follow a different path: both worsen through June, reach their lowest point in September, then partially recover in December without returning to March's level. Building 2 falls from fourteen green-tier combinations in March to eleven in June and only six in September, its worst season, before recovering to eleven in December. Building 3 rises from eight green-tier combinations in March to nine in June and zero in September, its worst season by a wide margin, before recovering to seven in December. September, therefore has the fewest green-tier combinations for both of these buildings, and December's low violations represent a recovery from that low point rather than a steady improvement from March onward.

December does not achieve the lowest violation of every season for every building, despite the very low figures reported in Table 4-8. Building 1's recommended violation, 0.0012, is the lowest recorded for that building across all four seasons, roughly an order of magnitude below its March value. Building 2's December value, 0.0044, is marginally higher than its June value, 0.0042; June, not December, is Building 2's best season overall. Building 3's December value,

0.0066, is likewise higher than its March value, 0.0029, which remains Building 3's best season overall. December's very low violations are therefore a feature of Building 1 specifically, not a universal property of the season.

Building 2's recommended combination returns to $(10, 10^2)$, the same combination found in March and June, after September's recommended combination was high regularization, $(10^5, 10^5)$. The building's regularization pattern is therefore not fixed across seasons but appears to depend on season-specific conditions.

Building 3's recommended combination, $(10^2, 10^3)$, sits between Building 1's and Building 2's on λ_y in every season except September, where it ties Building 1's value at the lower end rather than falling strictly between. On λ_g, this middle position holds only in March and December. In June, Building 3 ties Building 2 at the lowest value on the grid, and in September, Building 3's λ_g is lower than both of the other buildings, breaking the pattern entirely rather than sitting between them.

Table 4-8: Recommended regularization combination per building, December, $\varepsilon = 0.05$ selection rule applied to the measured sweep data.

Building	λ_g	λ_y	Mean violation (°C · room/hr)	Cost (EUR)
Building 1	10^4	10^5	0.0012	2747.4
Building 2	10	10^2	0.0044	2276.7
Building 3	10^2	10^3	0.0066	1742.4

December's raw energy cost is the highest of any season for every building: 2747.4 EUR versus September's 1822.5 EUR for Building 1, 2276.7 EUR versus 1661.7 EUR for Building 2, and 1742.4 EUR versus 1158.9 EUR for Building 3. This is consistent with December being the coldest of the four simulated months, requiring the most heating and therefore the highest energy expenditure.

Effects on the Gaussian Process Building 3's landscape becomes mixed along the diagonal for the first time in December, and its recommended combination breaks the positional pattern relative to Building 1 and Building 2 seen in most other seasons. Despite this, December's fitted Gaussian process is the most accurate of any season (§4-1-5). Its leave-one-out RMSE, 0.0058, is the lowest recorded in this thesis, and its fitted noise standard deviation, 0.0010, is also the lowest of any season, an order of magnitude below March's 0.0307. A rougher landscape for one building does not, in this case, translate into a worse overall fit.

Point removal required no intervention here either (§4-1-5): the largest residuals under an initial fit were concentrated systematically in Building 1, at combinations pairing high λ_g with low λ_y, rather than appearing as isolated anomalies, and this systematic pattern did not affect the recommendation check reported in §4-1-5.

Controller Behavior at Recommended Combinations

As discussed in §4-1-5, December is the coldest of the four simulated months, with steady, high heating demand rather than a transition between conditions. This is reflected directly in the trajectories that follow. Figures 4-19–4-21 show the zone temperature and radiator input

trajectories for the three training buildings, each at its own recommended combination, over the full 90-day simulation.

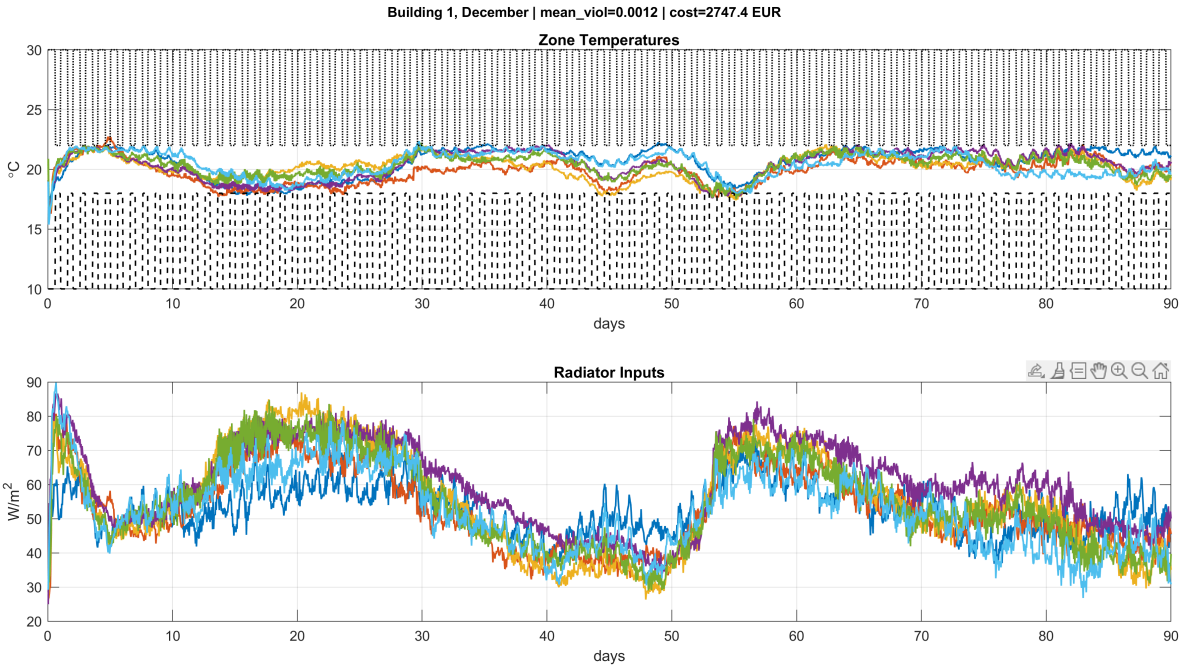


Figure 4-19: Zone temperatures and radiator inputs, Building 1, December, at the recommended combination $(\lambda_g, \lambda_y) = (10^4, 10^5)$.

Building 1 holds zone temperatures within the occupied-hours comfort band (Figure 4-19), consistent with the very low violation reported in Table 4-8. Zone temperatures track noticeably closer to the upper bound than to the lower bound throughout the simulation, unlike any other building–season trajectory presented in this thesis, consistent with its exceptionally low violation figure. Radiator input spikes sharply toward the 90 $\mathrm{W/m^2}$ December radiator limit in the first one to two days, then falls substantially to a trough around day 5–10. It then rises into a broad plateau spanning roughly days 13–28, rather than a brief peak, before declining to a second trough around days 40–50. Input rises sharply again around day 53–55, reaches a peak around day 55–60, and gradually declines, with continued short-term variation, through the remainder of the simulation.

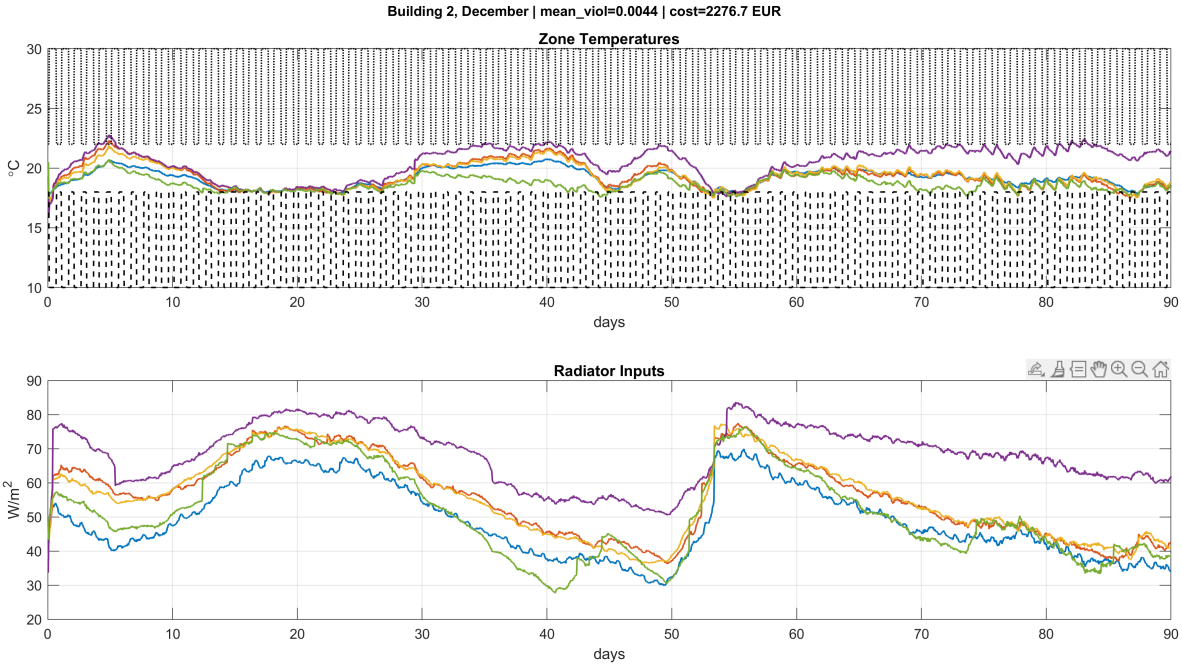


Figure 4-20: Zone temperatures and radiator inputs, Building 2, December, at the recommended combination $(\lambda_g, \lambda_y) = (10, 10^2)$.

Building 2 also holds zone temperatures within the comfort band (Figure 4-20). Radiator input shows a similar shape to Building 1's over the first 20–30 days, an elevated start followed by a partial pullback and a rise to a peak around day 15–20, though the pullback here is shallower, dropping only to around 60 $\mathrm{W/m^2}$ rather than the deep trough seen in Building 1. Input then falls to a trough around days 40–50, rises sharply again around day 53–55, and declines gradually afterward. One zone, Z4, runs consistently warmer and takes consistently more radiator input than the other four across the full 90 days, a persistent offset similar to the one noted for this building in March, though the size of the gap grows substantially from around day 50–55 onward, well beyond the tighter separation visible in the first half of the simulation.

Building 2's recommended combination, $(10, 10^2)$, is unchanged across March, June, and December, the only building–combination pair tested in three different seasons. This makes December a further test of the same point established in June: with tuning held fixed, differences in the realized input come from the weather being responded to, not from the controller. The differences here are substantial. December's radiator input reaches peaks of roughly 80 $\mathrm{W/m^2}$, higher than March's peak near 70 $\mathrm{W/m^2}$ and far above June's range, where the radiator limit itself is only 30 $\mathrm{W/m^2}$ (Table 4-1). December also shows nothing resembling June's extended near-zero quiet period; input stays substantial throughout the full 90 days, consistent with December being the coldest and most steadily demanding of the four simulated months.

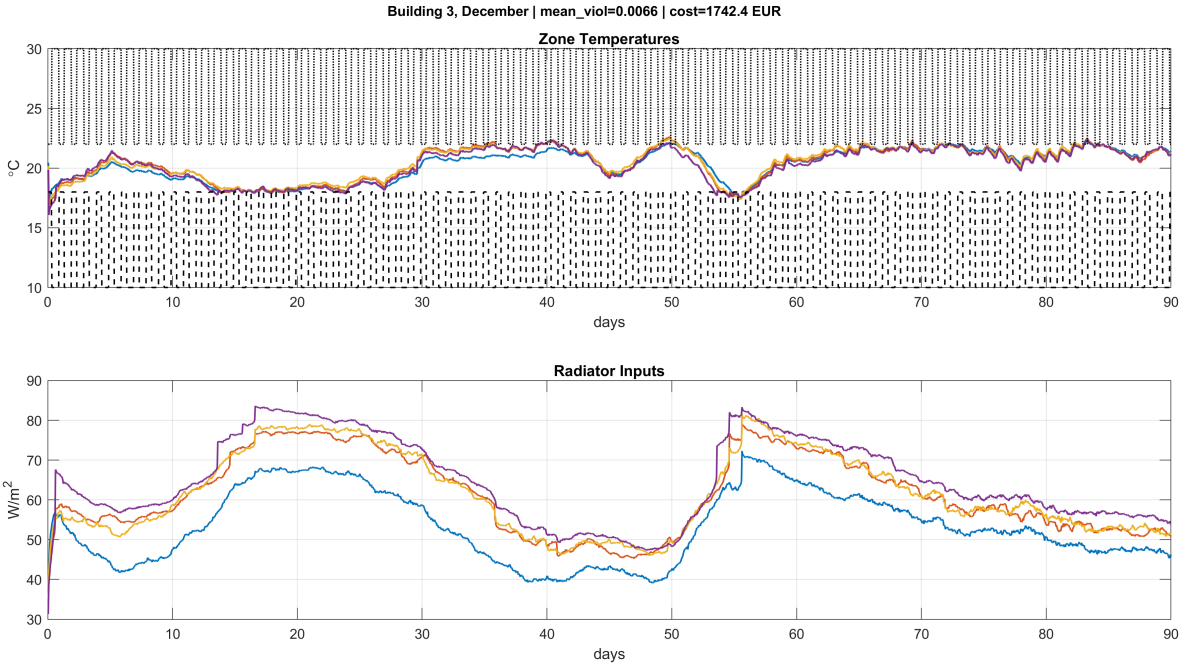


Figure 4-21: Zone temperatures and radiator inputs, Building 3, December, at the recommended combination $(\lambda_g, \lambda_y) = (10^2, 10^3)$.

Building 3 also holds zone temperatures within the comfort band (Figure 4-21). Like the other two buildings, its radiator input shows a brief early spike in the first day, settling back slightly before climbing steadily to a first peak around day 17–20, though this early spike is less pronounced than Building 1's or Building 2's. Input then falls to a trough around days 40–50 and rises sharply again around day 53–55 before declining, matching the later part of the pattern seen in the other two buildings. Unlike Building 1, Building 3's peaks, at both day 17–20 and day 53–55, remain below the 90 $\mathrm{W/m^2}$ December radiator limit, reaching only around 83 $\mathrm{W/m^2}$.

The double-peak structure common to all three buildings is consistent with the December simulation window spanning the turn of the calendar year, plausibly capturing a milder spell in early winter followed by a colder period later in the window, though this has not been independently verified against the underlying weather data. All three buildings show some form of a small early spike before their main first rise, though its size differs by building, most pronounced in Building 1 and Building 2 and comparatively small in Building 3.

Gaussian Process Fit

Point Removal As in September, no points were removed from the December training set. Figure 4-22 shows the resulting fit. The worst-fit residuals under an initial fit on all 75 points were modest (maximum residual 0.0187), well below March's 0.10 outlier threshold and below June's pre-removal residuals, which reached 0.0459. Unlike June, where three points stood out as clear anomalies separated from the rest of the distribution, December's largest residuals are concentrated almost entirely in Building 1 (five of the eight largest), at combinations pairing high λ_g with low λ_y. This concentration reflects a systematic, identifiable region of

the parameter space where the fit is comparatively weaker, rather than a small number of isolated anomalous points. Removing points from a systematic pattern rather than isolated anomalies would not be a principled application of June's removal criterion. Given both the modest residual magnitude and this systematic rather than anomalous character, no points were removed.

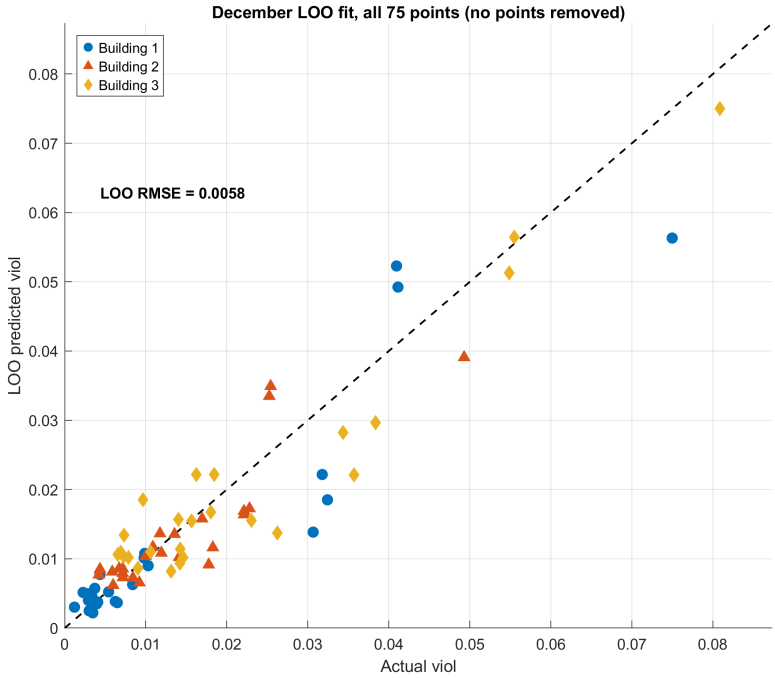


Figure 4-22: Leave-one-out predicted vs. actual mean violation, December, all 75 points. No points were removed.

Fit Quality The December violation Gaussian process, fit on all 75 points, achieves a leave-one-out RMSE of 0.0058, the best of any season in this thesis (March 0.0078, June 0.0067, September 0.0070). The fitted exponential kernel has lengthscale 9.2522, signal standard deviation 1.5763, and noise standard deviation 0.0010. The cost Gaussian process achieves a leave-one-out RMSE of 37.32 EUR, with fitted exponential kernel lengthscale 119.0658, signal standard deviation 0.4283, and noise standard deviation 0.0001.

Cross-Seasonal Comparison

Both the violation and cost GP models are fitted using MATLAB's `fitrgp` function, with an exponential kernel and a constant basis function, matching the kernel defined in Eq. (3-13). Model inputs are pre-normalized following §3-4-2, so `fitrgp`'s own standardization is disabled. The noise standard deviation σ_n is optimized via Bayesian optimization (`expected improvement plus` acquisition function, 30 objective evaluations), while the length scale ℓ and signal standard deviation σ_f are obtained from `fitrgp`'s own internal log marginal likelihood fit. This procedure is applied identically and independently to both the violation GP and the cost GP, for each season. Table 4-9 reports the resulting hyperparameters for all four seasons.

Table 4-9: Fitted Gaussian process kernel hyperparameters by season. All models use the exponential kernel Eq. (3-13) with a constant basis function.

Season	Model	Lengthscale ℓ	Signal std. σ_f	Noise std. σ_n	LOO RMSE
March	Violation	2.2958	1.0157	0.0307	0.0078 °C · room/hr
March	Cost	79.4907	0.4455	0.0001	30.87 EUR
June	Violation	1.7010	0.7782	0.2038	0.0067 °C · room/hr
June	Cost	20.3403	0.4501	0.0001	33.60 EUR
September	Violation	4.4525	1.0340	0.1672	0.0070 °C · room/hr
September	Cost	126.2418	0.4543	0.0006	32.79 EUR
December	Violation	9.2522	1.5763	0.0010	0.0058 °C · room/hr
December	Cost	119.0658	0.4283	0.0001	37.32 EUR

4-2 Validation

4-2-1 Interpolation

Building 1

To evaluate the trained model's behavior on buildings within the training feature range but not used to train it, three controlled variants of Building 1 were constructed, each differing from Building 1 in exactly one or two physical features while holding room count, floor area, and zone layout unchanged. The Wall variant reduces wall insulation from Building 1's 0.150 m to 0.135 m. The Roof variant reduces roof insulation from 0.020 m to 0.0165 m. The Wall+Roof variant applies both changes simultaneously. In each case the thermal time constant was left to react naturally to the insulation change rather than held fixed, since τ is not an independently controllable input to the building model.

Table 4-10: Physical features of the Building 1 interpolation variants, March.

Variant	d_{wall} (m)	d_{roof} (m)	A_{tot} (m^2)	τ (hrs)
Building 1 (original)	0.150	0.020	504	113.48
Wall	0.135	0.020	504	112.66
Roof	0.150	0.0165	504	107.56
Wall+Roof	0.135	0.0165	504	106.79

The Wall and Roof changes produce a combined τ shift in Wall+Roof (113.48 $\rightarrow$ 106.79, a change of -6.69 hrs) that is nearly identical to the sum of their individual effects (-0.82 and -5.92 hrs, summing to -6.74), differing by only 0.05 hrs. This suggests wall and roof insulation act as largely independent contributors to Building 1's thermal time constant over this range of perturbation, rather than interacting nonlinearly.

For each variant, the March model was queried at the variant's true feature vector, and the recommended combination was evaluated with a dedicated closed-loop simulation. Table 4-11 summarizes the results.

All three variants were recommended the same combination as Building 1 itself, $(10^5, 10^5)$, suggesting the region around Building 1 in feature space consistently recommends high regularization regardless of which specific feature is perturbed. All three actual violations

Table 4-11: Interpolation validation results, Building 1 variants, March. GP recommendation, predicted vs. actual violation, relative error of the prediction (with the predicted value as denominator) from Building 1's original point.

Variant	GP Choice	Pred. viol.	Actual viol.	Rel. error
Wall	$(10^5, 10^5)$	0.0060	0.0058	3.3%
Roof	$(10^5, 10^5)$	0.0079	0.0051	35.4%
Wall+Roof	$(10^5, 10^5)$	0.0080	0.0052	35.0%

fell within their respective 95% confidence intervals ([0.0017, 0.0214], [0.0017, 0.0373], and [0.0017, 0.0380] for Wall, Roof, and Wall+Roof respectively). The model's uncertainty quantification therefore remains valid across all three cases, even where the point predictions are less accurate.

The point predictions themselves, however, differ in accuracy. Wall is predicted tightly (3.3% relative error), while both Roof and Wall+Roof show a much larger, consistently pessimistic error of around 35%, in both cases predicting worse comfort performance than was achieved. One contributing factor is that the Roof and Wall+Roof variants are not clean single-feature perturbations. Because τ reacts to the insulation change rather than remaining fixed, and roof insulation is the dominant driver of the thermal time constant for this building, the Roof variant's τ shift is larger than Wall's (a normalized shift of -0.521 compared to -0.072). Roof and Wall+Roof therefore move further from Building 1's original point across two input dimensions simultaneously rather than one. Wall+Roof's actual violation (0.0052) and Roof's (0.0051) are nearly identical despite Wall+Roof combining two feature changes rather than one. The roof change alone appears to account for nearly all of the deviation from Building 1's behavior, with the additional wall change in Wall+Roof contributing little on top of it. This is consistent with roof insulation being the dominant driver of this building's thermal response: once the roof is perturbed, an additional wall perturbation makes comparatively little further difference to closed-loop performance, even though it does still measurably affect the underlying thermal time constant.

Building 2

For Building 2, the same interpolation design was applied: Wall, Roof, and Wall+Roof variants, with insulation shifted by the same normalized magnitude used for Building 1 (§4-2-1), reversed in direction since Building 2's original values already sit near the training range minimum on both features.

Table 4-12: Physical features of the Building 2 interpolation variants, March.

Variant	d_{wall} (m)	d_{roof} (m)	A_{tot} (m^2)	τ (hrs)
Building 2 (original)	0.100	0.015	420	94.60
Wall	0.115	0.015	420	94.96
Roof	0.100	0.0185	420	100.67
Wall+Roof	0.115	0.0185	420	101.04

As with Building 1, the Wall+Roof τ shift is close to additive. The individual Wall and Roof shifts, +0.36 and +6.07 hrs, sum to +6.43, against an observed +6.44 (0.01 hrs off).

Building 2's roof sensitivity (+6.07 hrs) closely matches Building 1's (−5.92 hrs for an equal-magnitude change in the opposite direction), while its wall sensitivity (+0.36 hrs) is smaller than Building 1's (−0.82 hrs). This is consistent with roof insulation acting as a dominant, building-independent driver of thermal response and wall insulation acting in a more building-specific way.

For each variant, the March model was queried at the variant's true feature vector, and the recommended combination was evaluated with a dedicated closed-loop simulation.

Table 4-13: Interpolation validation results, Building 2, March.

Variant	GP choice	Pred. viol.	Actual viol.	Actual cost (EUR)
Wall	$(10^5, 10^5)$	0.0065	0.0053	1325.21
Roof	$(10^5, 10^5)$	0.0090	0.0063	1299.33
Wall+Roof	$(10^5, 10^5)$	0.0083	0.0075	1282.51

All three variants recommend $(10^5, 10^5)$, achieving an actual violation in the Green tier in every case. This is notable because Building 2's own recommended combination is $(10, 10^2)$, the opposite corner of the grid. All three variants shift the recommendation to the high-regularization corner instead.

Inspecting the full predicted ranking for the Wall+Roof variant clarifies what this shift represents. $(10, 10^2)$ ranks fourth of 25 candidate combinations, with a predicted violation only 16.8% above the grid minimum, and is in fact the cheapest predicted option among the top-ranked candidates (1371.98 EUR, versus 1426.12 EUR for the selected $(10^5, 10^5)$). It falls outside the fixed 5% tolerance band used by the selection rule, so the cost tiebreaker never considers it. Both predicted violations, however, fall within the Green comfort tier. The fixed percentage threshold is therefore, in this instance, stricter than the practical comfort distinction it is meant to enforce: a modestly cheaper, comfort-equivalent alternative is excluded by a fixed relative cutoff rather than by a real difference in outcome. This is a characteristic of the selection rule's design, not a proposed change to it.

Building 3

For Building 3, the same interpolation design was applied: Wall, Roof, and Wall+Roof variants, with insulation shifted by the same normalized magnitude used for Building 1, reversed in direction since Building 3's original values already sit near the training range minimum on both features.

Table 4-14: Physical features of the Building 3 interpolation variants, March.

Variant	d_{wall} (m)	d_{roof} (m)	A_{tot} (m^2)	τ (hrs)
Building 3 (original)	0.110	0.013	288	86.17
Wall	0.125	0.013	288	86.36
Roof	0.110	0.0165	288	92.55
Wall+Roof	0.125	0.0165	288	92.77

As with Buildings 1 and 2, the Wall+Roof τ shift is close to additive. The individual Wall and Roof shifts, +0.19 and +6.38 hrs, sum to +6.57, against an observed +6.60 (0.03 hrs off).

This is the third of three buildings confirming near-perfect additivity between wall and roof insulation effects on τ.

Building 3's wall sensitivity (+0.19 hrs) is the smallest of the three training buildings (Building 1: −0.82 hrs; Building 2: +0.36 hrs; Building 3: +0.19 hrs). With only three data points this is noted as an observation rather than a confirmed trend. Building 3's roof sensitivity (+6.38 hrs) is consistent with the other two buildings (−5.92 hrs and +6.07 hrs for Building 1 and Building 2 respectively). This reinforces that roof insulation acts as a dominant, building-independent driver of thermal response, in contrast to wall insulation's comparatively building-specific and weaker influence.

For each variant, the March model was queried at the variant's true feature vector, and the recommended combination was evaluated with a dedicated closed-loop simulation.

Table 4-15: Interpolation validation results, Building 3, March.

Variant	GP combo	Pred. viol.	Actual viol.	Actual cost
Wall	$(10^5, 10^5)$	0.0054	0.0048	935.43
Roof	$(10^5, 10^5)$	0.0071	0.0035	909.38
Wall+Roof	$(10^5, 10^5)$	0.0079	0.0045	912.93

All three variants recommend $(10^5, 10^5)$, achieving an actual violation in the Green tier in every case. Building 3's own recommended combination is $(10^4, 10^4)$, so this represents a smaller shift than was observed for Building 2, whose recommended combination is the opposite corner of the grid. The direction of the shift is the same in both cases: every perturbation of Building 2 or Building 3 moves the recommendation to the high-regularization corner, while the Building 1 variants remain at Building 1's own combination, which already sits in that corner.

With all nine variants across the three training buildings now complete, every predicted and actual violation falls within the Green tier, and the recommended combination in every case is $(10^5, 10^5)$. This consistency, combined with the finding below that $(10^5, 10^5)$ is never far from any training building's own best measured violation, suggests the model has identified a region of the regularization grid with small regret across the training set. Its consistent recommendation of that region for perturbed buildings is then an understandable response to feature-space displacement rather than an arbitrary default. This is examined next.

Interpolation Bias Across Seasons

Across all nine interpolation experiments in March, the model recommended $(10^5, 10^5)$ in every case, regardless of which feature was perturbed, which building served as the base, or the direction of the perturbation. This consistency does not simply repeat any single training building's own recommended combination: only Building 1's own optimum is $(10^5, 10^5)$; Building 2's and Building 3's lie elsewhere in the grid. To understand whether this pattern reflected a property of the model or an artifact of the specific variants tested, a check was performed on how close $(10^5, 10^5)$ is to each training building's own best measured violation, using the real March sweep data directly (Table 4-16).

Table 4-16: Mean comfort violation at $(10^5, 10^5)$ compared to each training building's own recommended combination, March. Gaps are computed from unrounded violation values.

Building	Viol. at $(10^5, 10^5)$	Own recommended value	Gap
Building 1	0.0041	0.0041 (is the recommendation)	0.0%
Building 2	0.0061	0.0058	6.3%
Building 3	0.0031	0.0029	7.7%

Across all three training buildings, $(10^5, 10^5)$ is never more than approximately 8% worse than that building's own best measured violation, even where it is not the winning combination. The concentration of recommendations at this point is therefore not necessarily an error. The combination performs acceptably across the entire training set, which makes it a low-regret choice for a model that is uncertain about a new building's exact behavior.

To check whether this bias is specific to March or holds across seasons, the same nine interpolation points were queried against each season's own trained model. Tables 4-17–4-19 report the recommended combination for each building, variant, and season.

Table 4-17: Recommended combination at Building 1's interpolation variants, by season.

Variant	March	June	September	December
Wall	$(10^5, 10^5)$	$(10^5, 10^5)$	$(10^5, 10^5)$	$(10^4, 10^5)$
Roof	$(10^5, 10^5)$	$(10^4, 10^5)$	$(10^5, 10^5)$	$(10^4, 10^5)$
Wall+Roof	$(10^5, 10^5)$	$(10^4, 10^5)$	$(10^5, 10^5)$	$(10^4, 10^5)$

Table 4-18: Recommended combination at Building 2's interpolation variants, by season.

Variant	March	June	September	December
Wall	$(10^5, 10^5)$	$(10^4, 10^4)$	$(10^5, 10^5)$	$(10^4, 10^5)$
Roof	$(10^5, 10^5)$	$(10^4, 10^4)$	$(10^5, 10^5)$	$(10^4, 10^5)$
Wall+Roof	$(10^5, 10^5)$	$(10^4, 10^5)$	$(10^5, 10^5)$	$(10^4, 10^5)$

Comparing these three tables reveals a consistent asymmetry between buildings. Building 1 and Building 2 each recommend the same combination, or a combination differing by only one grid step, across all three of their own variants in every season. Building 3, by contrast, is internally inconsistent within a season: September recommends three different combinations across its three variants, and December recommends two. Building 3 is the only one of the three buildings whose own variants disagree with each other within a single season. This means a single recommendation for Building 3 is less trustworthy within a season than for Building 1 or Building 2, since which combination the model picks depends more sensitively on which physical feature happens to be perturbed.

The four seasons also do not group in the way a simple time-of-year ordering might suggest. March and September are the most similar to each other for Building 1 and Building 2, both settling on $(10^5, 10^5)$ in every variant, but they diverge sharply for Building 3, fully consistent at $(10^5, 10^5)$ in March against three different combinations in September. December looks distinct from both: Building 1 and Building 2 shift together by exactly one grid step, from $(10^5, 10^5)$ to $(10^4, 10^5)$, in every one of their variants, a small, uniform shift rather

Table 4-19: Recommended combination at Building 3's interpolation variants, by season.

Variant	March	June	September	December
Wall	$(10^5, 10^5)$	$(10^1, 10^2)$	$(10^3, 10^3)$	$(10^3, 10^4)$
Roof	$(10^5, 10^5)$	$(10^1, 10^2)$	$(10^5, 10^5)$	$(10^4, 10^5)$
Wall+Roof	$(10^5, 10^5)$	$(10^1, 10^2)$	$(10^4, 10^5)$	$(10^4, 10^5)$

than a change in kind. June is the clearest outlier of the four. It is the only season in which no combination is shared across all three buildings: Building 1 sits at $(10^4, 10^5)$ or $(10^5, 10^5)$, Building 2 at $(10^4, 10^4)$ or $(10^4, 10^5)$, and Building 3 uniformly at $(10^1, 10^2)$, a low-regularization combination unlike anything recommended for Building 3 in any other season. Every other season has at least two of the three buildings converging on a shared combination; June has none.

Building 3's greater instability in September and December is consistent with its own violation landscape being the most irregular of the three training buildings in both of these seasons, including its complete absence of a green-tier combination in September (§4-1-4). Building 1 and Building 2, whose own landscapes remain comparatively well-behaved in every season, are also the two buildings whose interpolation variants stay most consistent across seasons.

Whether this same building-specific breakdown also appears under extrapolation is addressed in §4-2-2.

4-2-2 Extrapolation

Two features were tested outside the training range: roof insulation, identified in the interpolation analysis as the dominant, building-independent driver of thermal response (§4-2-1), and room count, the one categorical feature the model does not observe directly, entering only indirectly through floor area and τ.

Roof Insulation

Building 3 was used as the base building, since its roof insulation, 0.013 m, sits at the training minimum. Two variants were constructed by varying roof insulation alone, with wall insulation, floor area, and zone layout held at Building 3's own values, and τ left to react naturally. The below-range case decreases roof insulation to 0.01 m, a small step past the training boundary. The above-range case increases roof insulation to 0.023 m. Because Building 3's roof insulation already sits at the training minimum, this is not a small step past the opposite boundary. It instead tests extrapolation across the entire training range, past Building 1's maximum of 0.02 m, rather than a local extrapolation near the upper boundary.

Both variants' observed τ fall slightly below what a linear extrapolation of the interpolation-range roof sensitivity (1822.9 hrs/m, §4-2-1) would predict: 104.40 hrs projected versus 102.52 hrs observed above range, and 80.70 hrs projected versus 79.99 hrs observed below range. Both deviations are small and point in the same direction. This is consistent with the roof–τ relationship becoming mildly sub-linear once ly outside the training range.

Table 4-20: Physical features of the Building 3 roof extrapolation variants, March.

Variant	d_{wall} (m)	d_{roof} (m)	A_{tot} (m^2)	τ (hrs)
Building 3 (original)	0.110	0.013	288	86.17
Above (0.023)	0.110	0.023	288	102.52
Below (0.01)	0.110	0.010	288	79.99

For each variant, the March model was queried at the variant's true feature vector, and the recommended combination was evaluated with a dedicated closed-loop simulation. Table 4-21 summarizes the results.

Table 4-21: Roof extrapolation validation results, Building 3, March. Tiers as elsewhere in this thesis (Green < 0.010, Amber 0.010–0.020, Red $\geq 0.020\,^{\circ}\text{C} \cdot \text{room/hr}$).

Variant	GP choice	Pred. viol.	Actual viol.	CI width
Above (0.023)	$(10^5, 10^5)$	0.0132	0.0033	0.0832
Below (0.01)	$(10^5, 10^5)$	0.0081	0.0050	0.0353

The two directions differ. The above-range insulation change (+0.010 m) is more than three times larger than the below-range change (−0.003 m). The roof feature's small standard deviation across the training set makes both displacements large in normalized terms, which is why even the modest below-range change already constitutes a meaningful displacement. The larger above-range displacement is reflected directly in the model's confidence interval, more than twice as wide for the above-range case (0.0832 versus 0.0353), and in its point prediction, which crosses from Green into Amber only for the above-range variant.

In the below-range case, both the predicted and actual violation fall in the Green tier, with a moderate gap between them. In the above-range case, the model's point prediction diverges from reality. The actual violation, 0.0033, is close to the lowest violation observed anywhere for Building 3 in this thesis, while the model predicted Amber-tier performance. The true value remains within the reported 95% confidence interval in both cases, so the model's uncertainty quantification remained valid even where its point estimate was conservative. The error is in the conservative direction: the model predicted worse performance than was achieved. For a comfort-critical application, this is the less consequential of the two possible directions of error.

Room Count

Room count is not itself an input to the trained model; only its indirect effects on floor area and τ are observed. Two validation buildings, extending one room below and one room above the training range of four to six rooms, were constructed by matching insulation exactly to the nearest real training building, isolating room count as the only substantive difference. The 3-room case, referred to as VH1, uses Building 3's insulation (0.110 m wall, 0.013 m roof), Building 3 being the nearest neighbor in room count. The 7-room case, referred to as VH2, uses Building 1's insulation (0.150 m wall, 0.020 m roof), Building 1 being the nearest neighbor in the other direction.

Table 4-22: Physical features of the room-count extrapolation buildings, March.

Building	Rooms	d_{wall} (m)	d_{roof} (m)	A_{tot} (m^2)	τ (hrs)
VH1 (below range)	3	0.110	0.013	252	86.26
VH2 (above range)	7	0.150	0.020	588	114.63

In both cases, τ is close to the matched training building despite the room-count difference: 86.26 hrs for VH1 versus Building 3's 86.17 hrs (0.09 hrs apart, one room removed), and 114.63 hrs for VH2 versus Building 1's 113.48 hrs (1.15 hrs apart, one room added). This indicates τ is governed primarily by envelope properties (wall and roof insulation) rather than by scale (room count or floor area), consistent across both an addition and a removal of a room. It follows that room count has little independent effect on the thermal response once insulation is fixed, so omitting it as an explicit model input costs little. The room-count information the model does receive enters through A_{tot} alone.

Table 4-23: Room-count extrapolation validation results, March.

Building	GP choice	Pred. viol.	Actual viol.	CI width
VH1 (3 rooms)	$(10^5, 10^5)$	0.0045	0.0066	0.0111
VH2 (7 rooms)	$(10^5, 10^5)$	0.0074	0.0037	0.0292

Both room-count extrapolations recommend the model's same high-regularization default and land in the Green tier for both predicted and actual violation, with no infeasibilities in either closed-loop simulation. Both validation buildings sit closer to their respective anchor building in normalized feature space than either roof extrapolation variant (0.41 and 0.95, compared to 1.16 and 3.71 for roof). This is consistent with room count, once insulation is held fixed, constituting a comparatively mild displacement in the feature space the model observes.

Across the four extrapolation cases tested, three produced predictions in the Green tier. The fourth, the most severe roof extrapolation produced a conservative point prediction whose true value nonetheless remained inside the stated confidence interval. This pattern is consistent with the model's confidence degrading appropriately with distance from the training data. In none of the four cases did the model recommend a combination that produced worse-than-predicted comfort outcomes.

4-2-3 Cross-Seasonal Consistency of the Regularization Bias

The March analysis in this chapter identified a strong bias toward $(10^5, 10^5)$ across nearly every interpolation and extrapolation case tested. To establish whether this bias is a general property of the method or specific to March, the June, September, and December models were each queried at the same 13 physical feature vectors used in the March experiments throughout this chapter. This check uses each season's own fitted Gaussian process to predict a combination and a violation at these points. No new closed-loop simulations were run for June, September, or December at these feature vectors, so only March's predictions can be compared against real, verified outcomes. The June, September, and December results reported here describe what each season's model would recommend and predict.

Table 4-24: Dominant recommended combination across the same 13 interpolation and extrapolation feature vectors, by season.

Season	Dominant combination	Frequency
March	$(10^5, 10^5)$	13/13 (100.0%)
June	$(10, 10^2)$	5/13 (38.5%)
September	$(10^5, 10^5)$	9/13 (69.2%)
December	$(10^4, 10^5)$	10/13 (76.9%)

Table 4-24 reveals a pattern considerably more structured than a single arbitrary default. In every season, the combination the model falls back to most often for perturbed or extrapolated buildings is not an arbitrary grid point. It is precisely one training building's own recommended combination for that season. Which building's combination serves as the fallback alternates by season rather than remaining fixed: March and December both default to Building 1's combination, while June and September both default to Building 2's. Building 3's combination, despite shifting considerably across seasons in its own right (§4-1-2–§4-1-5), never serves as the dominant fallback in any season tested.

This refines the March-specific finding from §4-2-1. Rather than an arbitrary bias toward a single point in the grid, the behavior generalizes as follows. When queried at a building whose features do not closely match any single training point, the model defaults to one training building's own recommended combination for that season, rather than to a fixed, season-independent point. For March, the fallback combination was verified against the measured sweep data to perform within approximately 8% of every training building's best violation (Table 4-16). For the other three seasons, the same low-regret property of the fallback is plausible but has not been verified against measured data, and this distinction is kept explicit here.

June departs from this pattern more than the other two seasons, in two respects. First, its fallback is less dominant. Building 2's combination accounts for only 38.5% of recommendations, compared to 69.2–100% in the other three seasons, with a second combination, $(10^4, 10^5)$, accounting for a further 30.8% without matching any training building's exact recommendation. Second, every one of the 13 variants receives a different recommendation from June's model than from March's, including the nine variants that were interpolation cases well inside the training range in March. This is consistent with June's violation surface being intrinsically less smooth than March's, a property already noted from the fitted kernel hyperparameters in §4-1-3 (lengthscale 1.7010 and noise standard deviation 0.2038 in June, against 2.2958 and 0.0307 in March), and now shown to have a direct, visible consequence for how the model behaves on buildings outside its exact training points.

Mean confidence interval width across these 13 points is broadly similar across March, June, and September, with December's fit notably tighter, consistent with December achieving the lowest leave-one-out RMSE of any season in this thesis (§4-1-5). This indicates that while the specific recommended combination and its underlying rationale differ by season, the model's uncertainty quantification, in the sense of producing appropriately sized confidence intervals, does not degrade in the other three seasons relative to March.

Taken as a whole, this cross-seasonal check gives a more precise characterization of the model's behavior on unfamiliar buildings than the March-only finding alone. The model does not

Table 4-25: Mean 95% confidence interval width across the same 13 feature vectors, by season. The March value over the nine interpolation cases alone is 0.0287.

Season	Mean CI width ($n = 13$)
March	0.0321
June	0.0330
September	0.0335
December	0.0225

have a single, arbitrary optimal combination. In every season examined, it falls back on one training building's own recommended combination, with the specific building that plays this role varying by season. June is the one exception where this fallback is less dominant and less exactly matched to a single training building, a limitation directly traceable to June's comparatively rougher fitted kernel rather than to the selection logic itself.

4-3 Discussion

This section addresses the three sub-questions introduced in §1-3-1: whether a Gaussian process can learn a useful mapping from building physical features to DeePC regularization parameters (SQ1), whether that mapping remains reliable across seasons (SQ2), and whether it generalizes to buildings outside the training set (SQ3). The results support a qualified yes to all three, with specific limitations identified below.

4-3-1 Recovery of Training Optima and Limits of the Selection Rule

Across all four seasons, the trained models recovered each training building's own recommended combination exactly in the large majority of cases: 3/3 in March, 2/3 in June, 2/3 in September, and 3/3 in December (§4-1-2–§4-1-5). Where the model did not recover the exact combination, the reason was traceable in both cases. June's Building 3 case (§4-1-3) is a near-tie resolved by the cost tiebreaker rather than a prediction failure: both candidates fell within 5% of each other in measured violation, and the model's own predictions placed them within its tolerance band together. September's Building 1 case (§4-1-4) is the opposite situation: the model's predictions placed the two candidates further apart than the measured data supports, so the tiebreaker did not activate when, by the measured numbers, it should have. These are two distinct limitations of the fixed 5% threshold interacting with model uncertainty, not evidence that the underlying violation predictions are unreliable.

4-3-2 Physical Interpretability of the Feature Set

A methodological choice made early in this thesis, using physically interpretable features (d_{wall}, d_{roof}, A_{tot}, τ) rather than a learned representation with no direct physical meaning, supported the validation design directly. The interpolation experiments established that roof insulation is a dominant, building-independent driver of each building's thermal time constant, while wall insulation's influence is comparatively small and building-specific (§4-2-1–§4-2-1).

This result was obtained through the construction of interpolation variants and confirmed three separate times, once per training building, with the roof sensitivity clustering tightly (5.92–6.38 hrs per 0.0035 m across all three buildings) while wall sensitivity varied by more than a factor of four (0.19–0.82 hrs for the same change). This physical grounding allowed the extrapolation design in §4-2-2 to target the feature the data itself had shown to matter most, and allowed the room-count extrapolation buildings to be constructed by holding insulation fixed at a real neighboring building's values, isolating room count as the only substantive variable under test.

4-3-3 Validation Performance

The interpolation validation produced a consistent result across all variants. Across all nine controlled variants (Wall, Roof, and Wall+Roof for each of the three training buildings), both predicted and actual violation fell within the Green comfort tier in every case (§4-2-1). The extrapolation results were more varied, as expected. Three of four cases (both room-count extrapolations and the below-range roof case) showed close agreement between prediction and reality, while the fourth, the most severe roof extrapolation tested, produced a substantial overprediction, forecasting Amber-tier performance for a building that achieved close to the best violation recorded anywhere for Building 3 in this thesis. The true value remained inside the reported 95% confidence interval in every case tested, including this one. The single point-prediction failure was in the conservative direction: the model predicted worse comfort than was achieved. At the tier level, no extrapolation case predicted a safer comfort tier than the building actually achieved, so an operator relying on the tier alone would never have been under-cautious. This conclusion rests on four extrapolation cases and cannot be guaranteed in general.

4-3-4 Structure of the Fallback Recommendation

The recurring recommendation of $(10^5, 10^5)$ throughout March's validation experiments initially appeared to be an unexplained artifact of the fitted model. Two checks resolved this. First, the measured March sweep data showed this combination is never more than approximately 8% worse than any training building's own best violation, even where it does not win outright, so it carries small regret across the training set. Second, the cross-seasonal check in §4-2-3 showed that this is not a March-specific result. In every season tested, the dominant fallback recommendation for a perturbed or extrapolated building corresponds exactly to one training building's own recommended combination for that season, with the specific building alternating between Building 1 (March, December) and Building 2 (June, September). This is a more specific finding than a bias toward one combination. It indicates a consistent pattern of defaulting to one training building's known tuning, verified to be low-regret for March and plausible but unverified for the other seasons.

4-3-5 Limits of Generalization

Two more specific limitations were established directly rather than inferred. First, outside of Building 1 in March, each training building's recommended combination behaves as an

essentially isolated point in the model's predicted landscape. Moving even one grid step away from Building 2's or Building 3's own coordinates was, with a single exception, sufficient to revert the recommendation to the season's default fallback, and that one exception held asymmetrically in only one direction. The model therefore does not treat a training building's combination as the center of a local pattern, which constrains how finely this approach distinguishes between similar buildings. Second, June stands apart from the other three seasons in the cross-seasonal check. Every one of the 13 tested feature vectors received a different recommendation from June's model than from March's, and June's dominant fallback was both less dominant (38.5%, versus 69–100% elsewhere) and less exactly aligned with a single training building's recommendation. This traces directly to June's fitted kernel being rougher than the other three seasons (shorter lengthscale, roughly seven times larger noise standard deviation than March), a property of that season's training data rather than an effect of the point-removal decision made for that season.

4-3-6 Scope Limitations

Several deliberate scope limitations have been set in the thesis. Wall insulation was excluded from extrapolation testing entirely. This is justified in retrospect by the interpolation findings showing wall insulation's comparatively weak and inconsistent effect, but it means the model's extrapolation behavior on this feature specifically remains unverified. The interpolation and extrapolation validation reported in detail in this chapter was conducted exclusively using March's model and real closed-loop simulation. The cross-seasonal check in §4-2-3 establishes how June, September, and December's models would respond to the same feature vectors, but no new closed-loop simulations were run to confirm those seasons' predictions against reality at these specific points.

Chapter 5

Practical Deployment and Conclusions

5-1 From Simulation to Practice

This chapter describes how the learned tuning rules would be used on a real building, which parts of the workflow transfer directly, and which parts require additional steps that the simulation study did not need. This chapter makes concrete the gap between the simulations of Chapter 4 and practical deployment, and shows that the method removes the most expensive part of it.

5-1-1 Deployment Workflow

The trained models take six inputs. Wall insulation thickness, roof insulation thickness, total conditioned floor area, and the dominant thermal time constant describe the building. The candidate λ_g and λ_y being evaluated select the point on the grid, since the model produces a separate prediction for each of the 25 candidate combinations (§3-2-2, §3-4-2). The first three are available from construction documentation for most office buildings and require no experiment. Given these inputs, a parameter recommendation for a new building follows five steps :

1. The operator measures the three construction features from documentation.
2. The operator estimates the thermal time constant τ as described in §5-1-2.
3. The operator selects the seasonal model matching the intended operating period.
4. The model is queried at the building's feature vector across the candidate grid, and the selection rule of §3-5 returns a combination (λ_g, λ_y).
5. The operator collects excitation data on the building and deploys Data-Enabled Predictive Control (DeePC) with the recommended weights.

The query itself requires 50 posterior mean evaluations and completes in under one second (§3-5). What the recommendation replaces is the per-building closed-loop sweep: in this thesis, one sweep for one building in one season required 25 simulations of 90 days each. On a real building, such a sweep would require years of operation under deliberately varied and partly poor tuning, which is not feasible. This step could not be carried out at all. The method replaces it entirely.

5-1-2 Estimating the Thermal Time Constant

In simulation, τ was computed from the dominant eigenvalue of the building's state matrix (§3-2-2). A real building has no state matrix. The time constant must instead be estimated from measured data. A practical route is a free-response test: heating is switched off at the start of an unoccupied period, such as a weekend, and the indoor temperature decay is recorded. The dominant time constant is then obtained by fitting an exponential to the decay [5]. Such tests in building thermal characterization require no equipment beyond the temperature sensors.

The accuracy required of this estimate is bounded by the results of Chapter 4. The room-count extrapolation showed that τ is governed primarily by the envelope properties (§4-2-2). A moderate error in the estimated τ therefore moves the query point within a region where the recommendation is flat. This robustness is a direct practical benefit of the fallback structure identified in §4-2-3, even though the same structure limits how finely the model distinguishes similar buildings.

5-1-3 Data Collection on a Real Building

Deploying DeePC on any building requires an excitation experiment to build the Hankel matrices, independently of how the regularization weights are chosen [7]. The tuning rules of this thesis do not remove this experiment. They remove the need to repeat closed-loop evaluation on top of it. In simulation, the excitation used a Pseudo-Random Binary Sequence (PRBS) signal applied in open loop for 1600 hours (§3-3-3). On an occupied building, open-loop excitation of this kind conflicts with comfort. Practical excitation must instead be applied within comfort bounds, for example by perturbing setpoints inside the comfort band, or scheduled in unoccupied periods. Prior experimental DeePC studies on real building systems have collected usable data under such constraints [17, 12]. The quality of the resulting Hankel matrices will be lower than in simulation, which is precisely the condition under which regularization matters most (§2-1-2). A sensible deployment therefore pairs the recommended combination with a short monitored trial period before unattended operation.

5-1-4 Handling Prediction Uncertainty at Deployment

The Gaussian process provides a confidence interval with every prediction, and Chapter 4 showed this interval remained valid in every validation case tested, including the one case where the point prediction missed substantially (§4-2-2). This suggests a simple deployment rule. If the confidence interval at the recommended combination is narrow, the recommendation can be applied directly. If the interval is wide, which occurred at extrapolation distance,

the building lies far from the training data and the recommendation should be treated as a starting point rather than a final choice. In the cases tested, the single large point-prediction error was in the conservative direction: the model predicted worse comfort than was achieved (§4-2-2). This was observed in four extrapolation cases and cannot be guaranteed in general, so the interval width, not the error direction, should be the operational signal.

The cross-seasonal analysis provides a second safeguard. In every season, the model's dominant fallback was one training building's own recommended combination, verified for March to perform within approximately 8% of every training building's best measured violation (§4-2-3). For a building far outside the training range, applying the season's fallback combination directly is therefore a defensible default while building-specific data accumulates.

5-1-5 Seasonal Operation

The results showed that the best combination for a single building can move across the entire grid between seasons. Building 2's recommended combination moved from $(10, 10^2)$ in June to $(10^5, 10^5)$ in September and back (§4-1-4, §4-1-5). A deployed system must therefore switch combinations with the operating season, using the model trained on the corresponding seasonal data. This fits naturally with the data-driven character of DeePC: when operating conditions change, the controller is updated by collecting new data rather than by re-identifying a model [4]. A practical schedule is to refresh the excitation data and re-query the seasonal model four times per year, at the season boundaries used in this thesis.

5-1-6 Remaining Gaps

Three gaps remain between this study and a field deployment, and each is inherited from the scope defined in §1-4. First, the training data come from one climate and one tariff structure. The weather signals were historical data for Central Europe and the tariff was a fixed two-level time-of-use profile (§3-1-4). A building in a different climate or under a different tariff requires training sweeps under those conditions before the learned mapping can be trusted. Second, the simulated buildings share one device configuration, one heat pump and one battery, and the mapping has not been tested on hubs with different equipment. Third, real measurements contain sensor noise and occupancy variation that the simulation did not include, and their effect enters exactly through the Hankel matrices that the regularization weights are meant to protect. The monitored trial period proposed in §5-1-3 addresses this last gap operationally, but a validation study on measured building data is the necessary next step before the tuning rules can be recommended without qualification

5-2 Conclusions

This thesis addressed the hyperparameter tuning problem of DeePC for building energy hubs. The regularization weights λ_g and λ_y must be chosen for each building, their optimal values cannot be computed analytically, and evaluating a single candidate requires a closed-loop simulation spanning multiple days. The central research question was how a Gaussian process regression model, trained on closed-loop performance data from a set of simulated building

energy hubs, can learn transferable tuning rules that map measurable building properties to suitable DeePC hyperparameters. This chapter summarizes the answers to the three sub-questions, states the contributions, and outlines future work.

5-2-1 Contributions

The thesis makes three contributions. First, it provides a systematic characterization of DeePC regularization behavior across three multi-zone building energy hubs and four seasons, 300 closed-loop simulations in total, showing how the effective region of (λ_g, λ_y) shifts with building construction and operating period. Second, it provides a pair of Gaussian process models per season, one for comfort violation and one for energy cost, that map building physical features and a candidate combination to predicted closed-loop performance, together with a selection rule that uses cost only to break near-ties in violation. Third, it provides an evaluation on thirteen buildings outside the training set, covering interpolation, feature extrapolation, and room-count extrapolation, including a verified explanation of how the model behaves when queried far from its training data. To the author's knowledge, this is the first implemented and validated instance of cross-building DeePC tuning rules learned from physical building characteristics.

5-2-2 Answers to the Research Questions

The first sub-question asked how DeePC closed-loop performance varies with the regularization parameters across buildings and seasons. The parameter sweeps show that the variation is large in both dimensions. Within a single season, the recommended combinations of the three training buildings span the grid from $(10, 10^2)$ to $(10^5, 10^5)$, and a combination that is best for one building can produce a violation more than double another building's best value. Across seasons, a single building's preference can reverse entirely, as Building 2's did between June and September. One structural regularity was identified: combinations with λ_g and λ_y of similar magnitude form a low-violation region in nearly every building and season, with December's Building 3 as the single exception. These results confirm that no single combination serves all buildings or all seasons, which is the premise of the learning approach.

The second sub-question asked how accurately a Gaussian process trained on a small set of buildings learns the mapping from building properties and regularization parameters to closed-loop performance. The four seasonal violation models achieved leave-one-out RMSE values between 0.0058 and 0.0078 $°\text{C} \cdot \text{room/hr}$ on 72 to 75 training points, small relative to the green comfort tier boundary of $0.010\,°\text{C} \cdot \text{room/hr}$. Queried at the training buildings' own feature vectors, the models recovered the measured recommended combination exactly in ten of twelve building-season cases. The two exceptions were traced to the interaction of the fixed 5% tolerance band with prediction error rather than to a wrong ordering of candidates. The mapping is therefore learnable from three buildings per season, with accuracy sufficient for the selection rule in most cases.

The third sub-question asked how well the recommendations generalize to buildings not seen during training. In nine controlled interpolation variants, the recommended combination produced green-tier comfort in every closed-loop evaluation, and every actual violation fell within the predicted 95% confidence interval. In four extrapolation cases, three predictions

matched reality closely and one, at the largest feature-space distance tested, was substantially conservative while remaining inside its stated confidence interval. The generalization mechanism was identified rather than assumed: for unfamiliar buildings the model falls back to one training building's own seasonal combination, a combination verified for March to lie within approximately 8% of every training building's best measured violation. Generalization is therefore reliable in the tested range.

Taken together, these answers support the central claim of the thesis. A Gaussian process trained on sweep data from three simulated buildings per season produces DeePC regularization recommendations for unseen buildings that achieve comfort performance competitive with exhaustive search, at the cost of one model query instead of 25 closed-loop simulations.

5-2-3 Limitations

The training set contains three buildings per season, and outside these points the model recommends coarsely, defaulting to a low-regret fallback rather than interpolating a building-specific optimum. The fixed 5% tolerance band of the selection rule caused both recommendation mismatches observed in this thesis, once by activating on a genuine near-tie and once by failing to activate on a distorted one. June's model is measurably rougher than the other seasons and its recommendations for unseen buildings were not verified in closed loop. The study is confined to simulation, one climate, one tariff, and one hub configuration, and the validated feature range extends one room and one insulation step beyond the training set.

5-2-4 Future Work

Four directions follow directly from the results. The first is closed-loop verification of the June model's diverging recommendations, the most consequential open question identified by the validation study. The second is enlarging the training set. The isolated-optimum behavior identified in this thesis indicates that three buildings per season is enough to learn a safe fallback but not enough to learn building-specific structure between the training points, and additional training buildings attack this directly. The third is replacing the fixed 5% tolerance band with a band scaled by the model's own predictive uncertainty at the queried point, which addresses both selection-rule failures observed in this thesis with one mechanism. The fourth is the step to measured data: estimating τ from a free-response test, collecting excitation data within comfort bounds, and validating the recommendations on a real building.

Artificial Intelligence Use Statement

In the preparation of this thesis, large language models and artificial intelligence tools were used in a supporting capacity:

- **Claude (Anthropic, Sonnet 5)** was used to assist with debugging MATLAB code, verifying citation accuracy against source material, refining academic tone and sentence structure, and auditing internal references throughout the manuscript.

All technical content, research methodology, data analysis, interpretations, and conclusions presented in this thesis are the original work of the author. The use of this tool was strictly limited to auxiliary support tasks, citation verification, and code refinement, and did not contribute to the intellectual substance or the primary scientific contributions of the research.

Bibliography

[1] L. Pérez-Lombard, J. Ortiz, and C. Pout. A review on buildings energy consumption information. *Energy and Buildings*, 40(3):394–398, 2008.

[2] J. Drgoňa, J. Arroyo, et al. All you need to know about model predictive control for buildings. *Annual Reviews in Control*, 50:190–232, 2020.

[3] G. Darivianakis, A. Georghiou, R. S. Smith, and J. Lygeros. The power of diversity: Data-driven robust predictive control for energy-efficient buildings and districts. *IEEE Transactions on Control Systems Technology*, 27(1):132–145, 2019.

[4] V. Kandylas. Hyperparameter tuning in DeePC for energy hub applications: A literature review. Technical report, Delft University of Technology, 2026.

[5] D. Sturzenegger, D. Gyalistras, V. Semeraro, M. Morari, and R. S. Smith. BRCM Matlab toolbox: Model generation for model predictive building control. In *Proceedings of the American Control Conference*, pages 1063–1069, 2014.

[6] V. Behrunani, M. Zagorowska, M. Hudoba de Badyn, F. Ricca, P. Heer, and J. Lygeros. Degradation-aware data-enabled predictive control of energy hubs. In *Journal of Physics: Conference Series*, volume 2600, page 072006, 2023. CISBAT 2023.

[7] J. Coulson, J. Lygeros, and F. Dörfler. Data-enabled predictive control: In the shallows of the DeePC. In *Proceedings of the European Control Conference*, pages 307–312, 2019.

[8] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. M. De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, 2005.

[9] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler. Robust data-enabled predictive control: Tractable formulations and performance guarantees. *IEEE Transactions on Automatic Control*, 68(5):3163–3170, 2023.

[10] J. Schwarz. Data-driven control of buildings and energy hubs. Technical report, ETH Zürich, Automatic Control Laboratory, 2020. Semester Thesis.

[11] V. Chinde, Y. Lin, and M. J. Ellis. Data-enabled predictive control for building HVAC systems. *Journal of Dynamic Systems, Measurement, and Control*, 144(8):081001, 2022.

[12] M. Zehner, A. Cavaterra, and S. Lambeck. Data-enabled predictive temperature and humidity control in a historical museum building. In *Proceedings of the European Control Conference*, pages 546–551, 2025.

[13] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[14] M. Zagorowska, C. König, H. Yu, E. C. Balta, A. Rupenyan, and J. Lygeros. Efficient safe learning for controller tuning with experimental validation. *Engineering Applications of Artificial Intelligence*, 143:109894, 2025.

[15] F. Berkenkamp, A. Krause, and A. P. Schoellig. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, 112:3713–3747, 2021.

[16] P. Mattsson and T. B. Schön. On the regularization in DeePC. In *IFAC-PapersOnLine*, volume 56, pages 625–631, 2023.

[17] D. Bilgic, A. Harding, and T. Faulwasser. Data-driven predictive control of bilinear HVAC dynamics: An experimental case study. *IEEE Control Systems Letters*, 8:1–6, 2024.

[18] J. Shi, Y. Lian, C. Salzmann, and C. N. Jones. Adaptive data-driven prediction in a building control hierarchy: A case study of demand response in Switzerland. *Energy & Buildings*, 333:115498, 2025.

[19] M. Cummins, A. Padoan, K. Moffat, F. Dörfler, and J. Lygeros. DeePC-Hunt: Data-enabled predictive control hyperparameter tuning via differentiable optimization. In *Proceedings of the 7th Annual Conference on Learning for Dynamics and Control*, volume 283 of *Proceedings of Machine Learning Research*, pages 673–685, 2025.

[20] J. Wang, S. Zhang, J. Liu, X. Ma, and H. Liu. Fine-tuning for data-enabled predictive control of noisy systems by reinforcement learning. *Evolving Systems*, 17:43, 2026.

[21] M. Zagorowska, E. C. Balta, V. Behrunani, A. Rupenyan, and J. Lygeros. Efficient sample selection for safe learning. In *IFAC-PapersOnLine*, volume 56, pages 10107–10112, 2023.

[22] S. Hirt, A. Höhl, J. Schaeffer, J. Pohlodek, R. D. Braatz, and R. Findeisen. Learning model predictive control parameters via Bayesian optimization for battery fast charging. In *IFAC-PapersOnLine*, volume 58, pages 742–747, 2024.

[23] L. F. L. Lemos, A. R. Starke, and A. K. da Silva. Constrained Gaussian processes as a surrogate model for simulation-based optimization of solar process heat systems. *Applied Energy*, 395:126028, 2025.

[24] Z. Ma, G. Jiang, Y. Hu, and J. Chen. A review of physics-informed machine learning for building energy modeling. *Applied Energy*, 381:125169, 2025.

[25] M. Zagorowska, L. Ortmann, G. Belgioioso, and L. Imsland. Adaptive tuning of online feedback optimization for process control applications. *arXiv preprint arXiv:2604.12863*, 2026.

[26] G. Darivianakis, A. Georghiou, R. S. Smith, and J. Lygeros. EHCM toolbox. Technical report, ETH Zürich, Automatic Control Laboratory, 2015.

[27] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020.

Glossary

List of Acronyms

BRCM	Building Resistance-Capacitance Modelling
DeePC	Data-Enabled Predictive Control
EHCM	Energy Hub Component Modelling
GP	Gaussian Process
HVAC	Heating, Ventilation, and Air-Conditioning
LOO	Leave-One-Out
LTI	Linear Time-Invariant
MPC	Model Predictive Control
OSQP	Operator Splitting Quadratic Program
PRBS	Pseudo-Random Binary Sequence
RMSE	Root Mean Square Error
COP	Coefficient of Performance