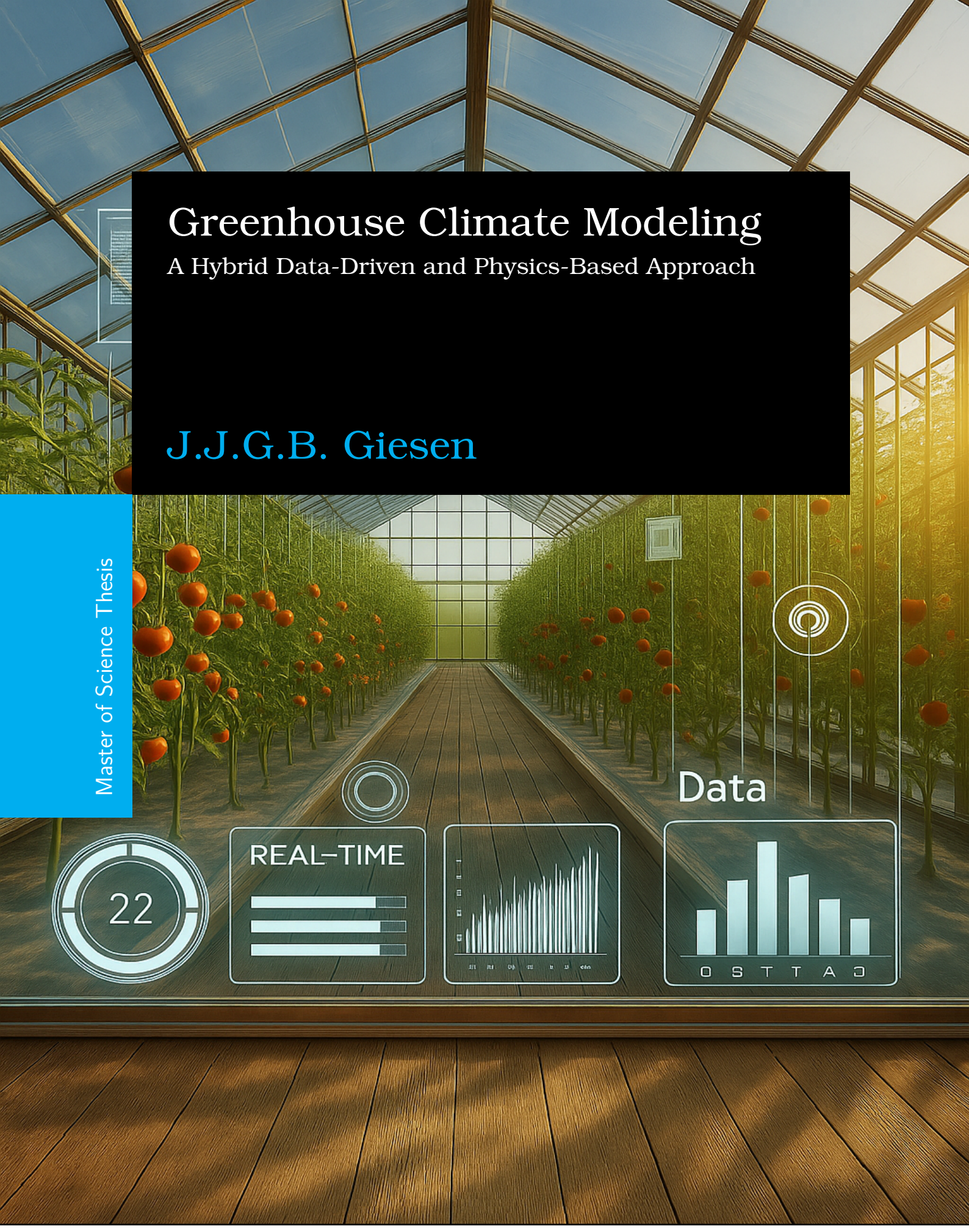
Greenhouse Climate Modeling

A Hybrid Data-Driven and Physics-Based Approach

J.J.G.B. Giesen

Master of Science Thesis



TU Delft Delft University of Technology

Delft Center for Systems and Control

Greenhouse Climate Modeling
A Hybrid Data-Driven and Physics-Based Approach

Master of Science Thesis

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

J.J.G.B. Giesen

July 1, 2025

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

The work in this thesis was supported by Hoogendoorn Growth Management. Their cooperation is hereby gratefully acknowledged.

Abstract

The urgent need for sustainable intensification of food production, driven by a growing global population and increasing climate variability, has positioned greenhouse cultivation and advanced climate control as critical areas of innovation in agriculture. Greenhouses enable precise environmental management, significantly boosting crop yields. However, realizing these benefits requires climate and crop models that accurately represent greenhouse dynamics while remaining interpretable and suitable for real-time control.

This thesis addresses greenhouse climate prediction by developing a hybrid modeling approach combining the strengths of physics-based system identification with modern machine learning techniques. In collaboration with Hoogendoorn Growth Management, a hybrid model was developed utilizing a physics-informed Sparse Identification of Nonlinear Dynamics (SINDy) method to capture primary mechanisms influencing temperature and humidity, augmented by an Long Short-Term Memory (LSTM) neural network to account for residual, unmodeled effects. A key contribution of this research is demonstrating the feasibility of transfer learning, successfully adapting a model trained initially on simulation data to real-world greenhouse scenarios, yielding reliable predictions for both air temperature and humidity under operational conditions.

Empirical results validate that transfer learning effectively bridges simulation and practical greenhouse environments, underscoring the practicality of data-driven climate models for industry applications. Integrating the model into an Model Predictive Control (MPC) framework illustrates its operational viability; the controller accurately tracks temperature setpoints but experiences challenges maintaining humidity within desired limits. These findings emphasize the necessity for accurate predictive models and underscore the importance of carefully formulating MPC strategies. Further research into modeling humidity dynamics, expanding the model's state space, and refining targeted retraining methodologies is essential to ensure robust, year-round practical deployment.

Overall, this thesis advances greenhouse climate modeling and control by showcasing how hybrid modeling combined with transfer learning can effectively close the gap between simulation-based development and operational implementation, thereby laying the groundwork for adaptive and efficient greenhouse management practices.

Table of Contents

Preface

After a long and winding academic journey, it is finally time to deliver my master piece. What began as a curiosity quickly grew into a deep fascination with the world of greenhouse agriculture, a field that is far more high-tech and dynamic than I could have imagined at the outset. The opportunity to collaborate with Hoogendoorn made this journey even more rewarding, giving me the chance to experience real-world innovation alongside passionate professionals.

Of course, none of this would have been possible without the support of many people, to whom I wish to express my gratitude.

First, I would like to thank my first supervisor, Prof. dr. ir. Tamas Keviczky, for his invaluable feedback and guidance throughout this thesis. I am equally grateful to my second supervisor, PhD candidate Ioannis Panagopoulos, for the countless hours spent discussing greenhouses, modeling, and navigating the thesis process as a whole. Both of you have helped shape my research and I want to thank you for your commitment and encouragement.

My thanks also go to Jeroen Crompvoets and the entire Data Science and Research team at Hoogendoorn. Working alongside you for the past year made the process a lot more enjoyable, and I appreciate your friendly conversations and expertise during my office days.

I am deeply grateful to my family and friends for their unwavering support, not only during this past year of writing but throughout my entire academic journey. A special thank you to my girlfriend, who has lived with me through the highs and lows of thesis life but has never stopped supporting me. This has been a challenging and lengthy chapter in my life, but also one of growth and joy. I am proud to close it now with this thesis.

Delft, University of Technology — J.J.G.B. Giesen
July 1, 2025

Chapter 1

Introduction

1-1 Motivation

Feeding a global population projected to reach approximately 9.7 billion people by 2050 will require a substantial increase in food production, estimated at around 70% more compared to 2010 levels [1]. At current productivity levels, this would lead to a significant expansion of arable land, potentially resulting in widespread deforestation [2]. Simultaneously, climate change is intensifying the frequency of extreme weather events, introducing increased climatic variability and stress on agricultural systems. Severe events such as heatwaves and droughts consistently have negative impacts on global crop yields [3]. These challenges underline the urgent need for sustainable intensification of agriculture, producing more food with fewer resources while increasing resilience to external disruptions.

Controlled environment agriculture, particularly greenhouse cultivation, provides a promising pathway toward addressing these challenges. Greenhouses allow precise climate control, effectively decoupling crop growth from external weather conditions and creating stable, highly productive environments. Compared to conventional open-field farming, greenhouses can significantly increase yields per unit area and substantially reduce the use of pesticides and irrigation water [4]. Thus, greenhouses present an attractive solution for enhancing agricultural productivity sustainably.

Realizing the full potential of greenhouse cultivation hinges on precise and adaptive climate control. The greenhouse environment is inherently complex, governed by dynamic interactions between external weather conditions, internal crop processes, and various actuators, including heating, ventilation, shading, irrigation, and CO_2 dosing. Automation is therefore crucial to effectively manage this complexity. In this context, model-based control approaches, particularly MPC, have become increasingly attractive. MPC utilizes a predictive model to anticipate future greenhouse climate states, enabling proactive adjustments of actuators and thus offering improved stability and efficiency compared to reactive control methods.

The accuracy and predictive capability of MPC depend critically on the underlying climate model. Traditional mechanistic models based on physical principles offer transparency and

interpretability but often struggle to fully represent complex real-world dynamics without extensive calibration. Conversely, purely data-driven methods, such as neural networks, can flexibly adapt to data but typically lack interpretability and face difficulties in generalizing beyond their training datasets.

To overcome these limitations, hybrid modeling approaches have gained interest, as they combine the strengths of both mechanistic and data-driven modeling paradigms. One prominent example is discrepancy modeling, where a physics-based model is augmented by a data-driven component trained specifically to capture residual errors. This method has proven effective in balancing interpretability with predictive accuracy, addressing previously unmodeled dynamics [5].

This thesis was initiated in collaboration with Hoogendoorn Growth Management, a leading developer of automation solutions for greenhouse climate control. Hoogendoorn is actively exploring new advanced control strategies and seeks to leverage extensive historical climate and crop data from commercial greenhouse operations to improve predictive modeling. Their primary question was whether historical operational data could be effectively used to enhance climate model accuracy. To address this, the thesis adopts a hybrid modeling approach that combines physics-based modeling with data-driven discrepancy learning. This approach was selected for its potential to make effective use of historical data while ensuring model interpretability and accuracy.

The main objective of this research is therefore to develop and evaluate a hybrid greenhouse climate model that effectively leverages available data, maintains interpretability, and provides accurate predictions suitable for use in real-time predictive control. This objective directly informs the research questions addressed in this thesis, outlined in the following section.

1-2 Research Questions

The central objective of this thesis is captured in the following main research question:

> *How can a hybrid modeling framework, combining physics-informed SINDy for system identification, LSTM-based discrepancy modeling, and transfer learning, be systematically designed and validated to provide accurate, interpretable, and generalizable greenhouse climate predictions for use in MPC?*

This is further explored through the following sub-questions:

1. How can domain knowledge be effectively incorporated into the candidate library of SINDy, and how does this inclusion impact the interpretability and physical consistency of the resulting greenhouse climate model?
2. To what extent does the hybrid SINDy-LSTM discrepancy model improve predictive accuracy compared to a purely physics-based SINDy approach?
3. How well does the pre-trained hybrid model generalize to unseen real-world greenhouse data, and how does transfer learning (fine-tuning) affect the prediction accuracy?

4. How robust is the fine-tuned model to seasonal variation, and can targeted re-training on recent or season-specific data further improve predictive performance throughout the year?
5. Is the final hybrid model suitable for integration into an MPC framework for greenhouse climate management, and how does it perform in terms of setpoint tracking and constraint satisfaction in operational scenarios?

1-3 Thesis Contribution

This thesis makes several contributions at the intersection of greenhouse climate modeling, machine learning, and control systems, addressing both scientific and practical challenges in sustainable agriculture. The key contributions are as follows:

1. Development of a Hybrid Physics-Informed Modeling Framework This thesis introduces and systematically develops a hybrid modeling framework that couples first-principles-based modeling with advanced machine learning techniques. By integrating the SINDy algorithm with a data-driven LSTM discrepancy learner, the framework achieves both physical interpretability and high predictive accuracy. Unlike conventional black-box approaches, the hybrid model explicitly encodes physical knowledge while remaining flexible enough to capture unmodeled or nonlinear phenomena present in real greenhouse operations.

2. Physics-Informed Library Construction and Model Sparsity A significant methodological contribution is the construction of a physics-informed candidate function library for SINDy, tailored to the unique thermodynamic and hydrodynamic processes in greenhouse environments. This ensures that the resulting models remain physically meaningful and interpretable, facilitating understanding and trust for both researchers and practitioners. The work demonstrates how domain knowledge can be systematically incorporated into data-driven modeling pipelines to enhance model sparsity, generalizability, and explanatory power.

3. Systematic Evaluation of Discrepancy Learning for Climate Prediction This thesis provides an empirical assessment of the discrepancy modeling paradigm, evaluating the hybrid SINDy-LSTM approach on both simulated and real-world greenhouse datasets. By explicitly modeling previously unrepresented dynamics, such as crop dynamics and other unmeasured processes, the hybrid approach demonstrates an enhanced ability to capture critical climate variables, including air temperature and absolute humidity. Furthermore, the work explores the robustness of this approach under seasonal and operational variability, providing insight into practical deployment challenges.

4. Application of Transfer Learning for Domain Adaptation Addressing the challenge of domain shift between simulation and operational greenhouses, this research applies transfer learning to adapt the pre-trained hybrid model to real-world conditions. The study demonstrates how fine-tuning on a limited amount of operational data enables the hybrid model to generalize and maintain predictive performance, thereby reducing reliance on extensive retraining or manual recalibration.

5. Integration and Validation within Model Predictive Control The thesis extends beyond model development to practical application, integrating the hybrid climate model into an MPC framework. It provides a systematic validation of the end-to-end approach, assessing its suitability for real-time climate management in terms of setpoint tracking, constraint satisfaction, and control smoothness. This contribution bridges the gap between advanced modeling and operational deployment.

Taken together, these contributions advances the state of the art in hybrid dynamical modeling and predictive control for greenhouse environments, offering practical tools and insights for achieving sustainable, data-driven agricultural intensification.

1-4 Thesis Outline

This thesis is organized to guide the reader from foundational system knowledge, through methodological innovation, to practical application and final conclusions.

Chapter 2 introduces the operational and physical principles underlying modern greenhouse cultivation. It provides an overview of the greenhouse climate system, describing the energy and mass flows, as well as the main control mechanisms used in practice. The Tap greenhouse climate model is introduced, and the specific model structure and variables used throughout this thesis are defined.

Chapter 3 presents the theoretical and methodological background for the hybrid modeling and control framework developed in this work. It reviews the SINDy algorithm for interpretable physical modeling, the use of LSTM neural networks for modeling temporal patterns in climate data, techniques for transfer learning to enable domain adaptation, and the core principles of MPC as applied to greenhouse environments.

Chapter 4 describes the data sources and processing steps that are used in this thesis. This includes the generation of simulation data using the Tap greenhouse model, a discussion of the available real-world greenhouse datasets, and the procedures used for preprocessing, cleaning, and structuring the data for subsequent analysis.

Chapter 5 details the development and evaluation of a physics-informed SINDy model for greenhouse climate prediction. It explains the construction of candidate function libraries for absolute humidity and air temperature, describes the training and parameter selection process, and presents an evaluation of the model's predictive performance, with a particular focus on generalization across different seasons.

Chapter 6 introduces the hybrid modeling framework that augments the SINDy model with an LSTM-based discrepancy learner. It outlines the two-stage training approach, the design of the LSTM network, the selection of hyperparameters, and the transfer learning strategy that enables adaptation to real-world greenhouse data. Validation results are provided for both simulated and real datasets, including analyses of multi-step forecasting accuracy, model sensitivity, and robustness to seasonal changes.

Chapter 7 demonstrates the application of the developed hybrid model within a MPC framework for greenhouse climate management. This chapter formulates the MPC problem, including the definition of control objectives, constraints, and cost functions. It then evaluates the closed-loop performance of the hybrid model in various simulation scenarios, assessing the

system's ability to track climate setpoints, satisfy operational constraints, and deliver robust performance across different conditions.

Finally, Chapter 8 summarizes the main findings of the thesis and provides direct answers to the research questions. It discusses the broader implications of the results, reflects on the limitations of the current work, and offers recommendations for future research directions in hybrid modeling and advanced greenhouse control.

Chapter 2

The Greenhouse System

2-1 Introduction

Greenhouse systems are vital to modern agriculture, providing a controlled environment that enhances growth conditions for various crops by regulating key factors such as temperature, humidity, and CO_2 levels [6]. This ability to manipulate the internal climate enables year-round cultivation and improved yields, making greenhouses essential for efficient and sustainable agricultural practices [7].

Despite the diversity of greenhouse types worldwide, which may differ in shape, covering materials, and operational subsystems, the fundamental principles governing greenhouse climate remain consistent [8]. Regardless of these differences, the core mass and energy exchanges, as well as information flows, are generic and critical to all greenhouse systems. These fluxes involve the transfer of heat, water vapor, and CO_2 between the greenhouse and its surroundings, as well as internally, driven by processes such as ventilation, heating, cooling, and transpiration [9]. Accurate modeling of these exchanges is essential for predicting climate dynamics and implementing effective control measures.

In addition to these physical exchanges, information flows are integral to greenhouse climate control. These flows involve real-time data acquisition through sensors monitoring climate variables, and the subsequent adjustments made by control systems to maintain optimal growing conditions [7]. Effective information management enables precise regulation, ensuring that the internal environment supports optimal plant growth.

This chapter provides an overview of the fundamental principles governing greenhouse systems, describing key components and their functions, and a detailed examination of mass, energy, and information flows between the greenhouse climate and crops. Particular emphasis is placed on the general climate system and on the variables and conceptual structure used in the Tap model [10], which is applied in this thesis for generating simulation data and as a proxy for a real greenhouse in the control loop.

2-2 Greenhouse Climate Principles

Greenhouse systems rely on an intricate balance of mass, energy, and information fluxes, as visualized in Figure 2-1. Mass and energy fluxes involve the movement of materials such as water vapor and CO_2, and energy in the form of heat, both within the greenhouse and between the greenhouse and its surrounding environment [11]. These include fluxes between equipment and the greenhouse environment, between the greenhouse air and the outdoor environment, between the internal greenhouse environment and the crop, and between the crop and the outdoor environment [12]. These transfers are crucial in regulating the internal climate, impacting factors such as humidity, temperature, and CO_2 levels, all of which are vital for plant growth.

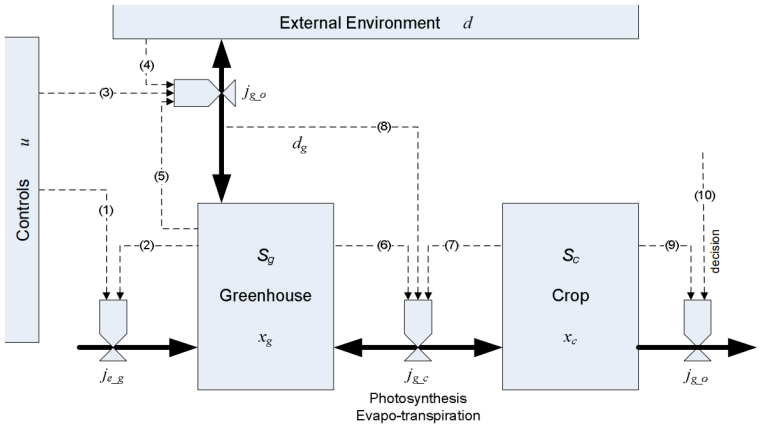


Figure 2-1: Simplified Greenhouse System with the mass and energy fluxes (solid lines) and the information flows (dashed lines) [12]

Mass and Energy Fluxes The system shows four main types of mass and energy fluxes:

- j_{e_g}: Fluxes between the equipment and greenhouse.
- j_{g_o}: Fluxes between the greenhouse air and the outdoor environment.
- j_{g_c}: Fluxes between the greenhouse internal environment and the crop.
- j_{c_o}: Fluxes between the crop and the outdoor environment.

Information Fluxes The system also contains multiple information flows:

- (1) Control inputs such as opening and closing of the heat- and CO_2-supply valve.
- (2), (5), (6) States of the greenhouse such as air temperature, CO_2 concentration, and humidity.
- (3) Control input for the window opening and closing.
- (4) External disturbances such as wind speed, radiation flux, and external temperature.

- (7), (9) States of the crop such as growth stage and number of fruits and leaves.
- (8) Direct influence of solar radiation on j_{g_c} fluxes such as CO_2 and water uptake and release.
- (10) Discrete decision actions such as picking leaves and pruning.

2-3 The Tap Greenhouse Climate Model

The Tap model is a widely cited, physics-based framework for simulating greenhouse climate dynamics, providing a quantitative description of the primary energy and mass balances within the greenhouse environment [10, 13]. The Tap model represents five core state variables: greenhouse air temperature (T_g), soil temperature (T_s), heating pipe temperature (T_p), CO_2 concentration (C_i), and absolute humidity of the greenhouse air (V_i). The time evolution of these variables is governed by a system of differential equations, reflecting the physical processes of heat and mass transfer as well as biological fluxes.

In this thesis, the Tap model plays a three-fold role. First, it is used to generate simulation data for model development and benchmarking. Second, it serves as the basis for analysis to inform the construction of the physics-informed candidate library for SINDy. Third, it acts as the ground-truth system within the MPC framework, providing a realistic reference for closed-loop control evaluation. This integrated approach ensures that all models and control strategies are developed, benchmarked, and validated against a physically meaningful and well-established greenhouse climate description.

The dynamic behavior of each state variable in the Tap model is described by the following set of differential equations, which together provide a comprehensive representation of greenhouse climate processes.

Greenhouse Air Temperature (T_g): The dynamics of greenhouse air temperature are described by:

$$\dot{T}_g = \frac{1}{C_g}\left[(k_v + k_r)(T_o - T_g) + \alpha(T_p - T_g) + k_s(T_s - T_g) + \eta G - \lambda E + \frac{\lambda}{1+\epsilon} M_c\right] \tag{2-1}$$

where C_g denotes the heat capacity of the greenhouse air. The model accounts for several heat and mass transfer mechanisms: ventilation and cover heat transfer coefficients (k_v and k_r) represent heat exchange with the outside air at temperature T_o; α is the heat transfer coefficient between the heating pipes and air, with T_p as the heating pipe temperature; k_s is the soil-to-air heat transfer coefficient with T_s as the soil temperature. Additional terms include the fraction of solar radiation transmitted through the greenhouse cover (ηG), the latent heat associated with crop transpiration (λE), and condensation at the cover (M_c), which is modulated by the latent heat factor and cover heat resistance (ϵ).

Heating Pipe Temperature (T_p): The dynamics of heating pipe temperature are described by:

$$\dot{T}_p = \frac{1}{V_p}\left[\phi_h(T_h - T_p) + \frac{A_p}{\rho_w C_p}(\beta G - \alpha(T_p - T_g))\right] \tag{2-2}$$

Here, V_p is the pipe volume, ϕ_h is the opening of the heating valve, and T_h is the supply water temperature. The first term quantifies the energy supplied to the pipes, while the second term describes heat exchange between the pipes and both the greenhouse air and solar radiation, where A_p is the pipe surface area, ρ_w is the density of water, C_p is the specific heat of water, and β is the absorption coefficient for solar energy.

Soil Temperature (T_s): The dynamics of soil temperature are described by:

$$\dot{T}_s = \frac{1}{C_s}\left[k_s(T_g - T_s) + k_d(T_d - T_s)\right] \tag{2-3}$$

In this equation, C_s is the soil heat capacity, k_s is the soil-to-air heat transfer coefficient, and k_d is the transfer coefficient with the deep soil layer at temperature T_d. This model captures the exchange of heat between the greenhouse air, upper soil, and deeper soil strata.

Greenhouse CO_2 Concentration (C_i): The dynamics of greenhouse air CO_2 concentration are described by:

$$\dot{C}_i = \left(\frac{V_g}{A_g}\right)^{-1}\left[\Phi_v(C_o - C_i) + \phi_{inj} + R - \mu P\right] \tag{2-4}$$

where V_g/A_g is the average greenhouse height, Φ_v is the ventilation flux, C_o is the outside CO_2 concentration, ϕ_{inj} is the injection flux of CO_2, R is crop respiration, P is the rate of photosynthesis, and μ is a stoichiometric constant accounting for CO_2 consumption.

Greenhouse Absolute Humidity (V_i): The dynamics of greenhouse absolute humidity are described by:

$$\dot{V}_i = \left(\frac{V_g}{A_g}\right)^{-1}\left[E - \Phi_v(V_i - V_o) - M_c\right] \tag{2-5}$$

Here, E denotes crop transpiration, Φ_v is the ventilation flux, V_o is the outside absolute humidity, and M_c is the condensation mass flow at the cover. This equation captures the interplay between moisture production, ventilation-driven exchange, and condensation losses.

The most important auxiliary relations for heat and mass transfer coefficients and biological processes are given below:

- **Ventilation heat transfer coefficient:** $k_v = \rho_a c_p \Phi_v$, where ρ_a is air density, c_p is specific heat of air, and Φ_v is the ventilation flux.
- **Ventilation flux:**
$$\Phi_v = \left(\frac{\sigma\phi_{\text{lee}}}{1 + \chi\phi_{\text{lee}}} + \zeta + \xi\phi_{\text{wind}}\right) w + \psi$$
where ϕ_{lee} and ϕ_{wind} are vent openings (as a percentage of maximum opening), w is wind speed, and σ, χ, ζ, ξ, ψ are empirical coefficients.
- **Pipe heat transfer:** $\alpha = \nu/\tau + \sqrt{|T_g - T_p|}$, with ν, τ empirical parameters.
- **Latent heat:** $\lambda = l_1 - l_2 T_g$, with l_1, l_2 coefficients.

- **Crop transpiration:**
$$E = \frac{s\eta G + \rho_a c_p D_g g_b}{s + \gamma(1 + \frac{g_b}{g})}$$
where s is the slope of the saturated vapor pressure curve, D_g is the vapor pressure deficit, g_b is leaf boundary conductance, γ is the psychrometric constant, and g is the leaf conductance.

- **Slope of saturated vapor pressure:** $s = s_1 T_g^2 + s_2 T_g + s_3$.

- **Vapor pressure deficit:** $D_g = p_g^* - p_g$, with p_g^* the saturated vapor pressure and p_g the air vapor pressure.

- **Cover condensation mass:**
$$M_c = \begin{cases} m_1 |T_g - T_c|^{m_2} (W_g - W_c^*) & \text{if } W_g > W_c^* \\ 0 & \text{otherwise} \end{cases}$$
where T_c is cover temperature, W_g and W_c^* are humidity ratios.

All parameter definitions, empirical relationships, and auxiliary states are provided in detail in Tap et al. [10]. This comprehensive structure enables the Tap model to represent the major thermodynamic and physiological drivers of greenhouse climate.

In compact form, the state, control, and external input vectors for the Tap model are:

$$x_g = \begin{bmatrix} T_g \\ T_p \\ T_s \\ C_i \\ V_i \end{bmatrix}, \quad u = \begin{bmatrix} T_h \\ \phi_{\text{lee}} \\ \phi_{\text{wind}} \\ \phi_c \end{bmatrix}, \quad v = \begin{bmatrix} T_o \\ T_d \\ C_o \\ V_o \\ w \\ G \end{bmatrix}$$

where x_g denotes the greenhouse state, u the actuators, and v the exogenous disturbances.

2-4 Climate Model Structure and Variables

While the full Tap model allows for detailed mechanistic studies of greenhouse climate, a reduced set of variables is typically sufficient for control and operational purposes. In this thesis, a minimal subset is selected to balance physical representativeness, measurement availability, and relevance for climate control. For simplicity, and because the primary aim is to explore the modeling framework itself, the focus is on variables that are routinely measured, directly controllable, and essential for describing the main climate dynamics.

The variables used in this thesis are:

- **Greenhouse air temperature (T_{air}, state):** Temperature of the air inside the greenhouse.

- **Absolute humidity inside ($AH_{\mathbf{in}}$, state):** Mass concentration of water vapor in the air inside the greenhouse.
- **Energy screen position ($E_{\mathbf{screen}}$, control):** Fractional position of the energy screen.
- **Windward vent position ($Vent_{\mathbf{wind}}$, control):** Fractional opening of the vent on the windward side.
- **Leeward vent position ($Vent_{\mathbf{lee}}$, control):** Fractional opening of the vent on the leeward side.
- **Heating pipe temperature ($T_{\mathbf{pipe}}$, control):** Temperature of the water supplied to the greenhouse heating pipes.
- **Outside air temperature ($T_{\mathbf{out}}$, exogenous):** Temperature of the outside air.
- **Outside absolute humidity ($AH_{\mathbf{out}}$, exogenous):** Mass concentration of water vapor in the outside air.
- **Global solar radiation ($I_{\mathbf{glob}}$, exogenous):** Total solar radiation incident on the greenhouse.
- **Outside wind speed ($U_{\mathbf{wind}}$, exogenous):** Speed of the wind measured outside the greenhouse.

Variables such as CO_2 concentration, soil temperature, and canopy temperature, while valuable for specialized or research-focused applications, are not explicitly included in this thesis, in order to maintain focus and simplicity. The modeling framework remains extensible should additional states be needed in future work.

In the original Tap model, the heating pipe temperature T_p is a dynamic state determined by the heating system, with the heating water temperature T_h and valve position ϕ_h as the main control inputs. In this thesis, T_{pipe} is treated directly as a control input, which maintains physical interpretability and simplifies integration into climate control schemes.

Chapter 3

Methodology and Background Information

3-1 Methodology Overview

This thesis employs a hybrid modeling and control framework for greenhouse climate management, integrating physics-based modeling, data-driven machine learning, and model based control. The central objective is to develop a control-oriented model that achieves a balance between physical interpretability, generalizability across different conditions, and predictive accuracy, and to implement this model within a real-time MPC architecture. The methodology consists of four main stages:

1. Physics-Informed Modeling with SINDy

The modeling process begins with the application of the SINDy algorithm to identify interpretable Ordinary Differential Equations (ODEs) governing the evolution of key greenhouse climate variables, specifically air temperature and absolute humidity. SINDy utilizes a candidate function library that incorporates terms based on physical principles relevant to greenhouse dynamics, such as solar radiation, ventilation, and heating. The resulting models are sparse, emphasizing interpretability and physical consistency.

2. Learning Residual Dynamics with LSTM Networks

The SINDy model, while physics-informed, may not capture all system dynamics due to unmodeled effects, nonlinearities, or temporal dependencies. To address this, a LSTM neural network is trained on the residuals between observed and SINDy-predicted derivatives. This LSTM-based discrepancy model learns to represent the remaining dynamics not captured by the initial physics-based model, thereby improving overall predictive accuracy without compromising interpretability.

3. Transfer Learning for Real-World Adaptation

The initial hybrid model (SINDy + LSTM) is trained using data generated from a simulated greenhouse environment. To adapt the model for use with real-world greenhouses, which may differ in structure, crop type, or sensor configuration, transfer learning is applied. The pre-trained LSTM model is fine-tuned using a limited set of measured operational data from the target greenhouse. This process enables the model to capture site-specific dynamics while retaining its general predictive capability.

4. Implementation in Model Predictive Control

The fully trained hybrid model is integrated into a MPCframework using the GEKKO optimization library. The MPC computes optimal control actions for variables such as heating setpoints, vent positions, and screen shading, subject to constraints that reflect system limits and operational requirements, including temperature and humidity bounds. The objective of the MPC is to minimize a cost function that balances setpoint tracking accuracy and actuator usage, enabling effective and efficient climate management in the greenhouse. This approach supports real-time optimization and practical deployment in a control environment.

This methodology enables the systematic combination of physics-based and data-driven modeling approaches for robust, interpretable, and adaptive greenhouse climate control.

3-2 Background Information on SINDy

The SINDy framework is a data-driven method for discovering parsimonious, interpretable models of nonlinear dynamical systems directly from time-series data [14]. In many engineering applications, governing equations are unknown or only partially known, while data are abundant. SINDy leverages the empirical observation that the dynamics of many physical systems are sparse in a suitable function space, meaning only a few terms from a large set of possible candidate functions are actually active in the system dynamics.

Mathematical Formulation

Consider a continuous-time deterministic dynamical system,

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}), \tag{3-1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector and $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is an unknown nonlinear function.

SINDy assumes that $\mathbf{f}(\mathbf{x})$ can be represented as a sparse linear combination of candidate basis functions. Define a library of candidate functions e.g.,

$$\Theta(\mathbf{x}) = [\mathbf{1}, x_1, \ldots, x_n, x_1^2, x_1x_2, \ldots, \sin(x_1), \ldots], \tag{3-2}$$

where the choice of terms is tailored to the problem domain. Evaluated over the data, this yields $\Theta(\mathbf{X}) \in \mathbb{R}^{m \times p}$, where m is the number of samples and p is the number of candidate functions.

The dynamical system is then written as

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi, \tag{3-3}$$

where $\dot{\mathbf{X}}$ are estimated time derivatives, and $\Xi \in \mathbb{R}^{p \times n}$ is a sparse coefficient matrix (each column corresponds to one state equation).

SINDy formulates a sparse regression problem for each state x_j:

$$\xi_j = \arg\min_{\xi_j} \|\Theta(\mathbf{X})\xi_j - \dot{\mathbf{x}}_j\|_2^2 + \lambda\|\xi_j\|_1, \tag{3-4}$$

where λ is a regularization parameter, and $\|\cdot\|_1$ denotes the ℓ_1 norm to promote sparsity. While LASSO is a popular choice, Sequential Thresholded Least Squares (STLSQ) is also widely used for computational efficiency [14, 15].

STLSQ Optimizer

A critical component of the SINDy framework is the optimizer used to identify the sparse set of active terms in the candidate library. STLSQ optimizer is widely used due to its simplicity, efficiency, and robustness to noise.

The STLSQ algorithm operates as follows:

- **Initial regression:** A standard least-squares regression is first performed to fit all candidate functions to the target derivatives, solving

$$\min_{\mathbf{w}} \|y - X\mathbf{w}\|_2^2 + \alpha\|\mathbf{w}\|_2^2, \tag{3-5}$$

  where X is the library of candidate functions, y is the target derivative, $\mathbf{w}$ is the coefficient vector, and α is the L2 regularization parameter.

- **Thresholding:** After fitting, coefficients with an absolute value below a set threshold are zeroed, enforcing sparsity in the model.

- **Iterative refinement:** The regression and thresholding steps are repeated, using only the active (nonzero) terms, until the set of active terms stabilizes or a convergence criterion is met.

Key parameters include:

- **Threshold:** Controls model sparsity by zeroing small coefficients; higher thresholds produce sparser models.

- **Alpha:** Regularizes coefficient magnitudes, balancing fit quality and model simplicity.

Noise Robustness and Practical Considerations

The STLSQ optimizer, by leveraging regularization and iterative thresholding, offers robustness to moderate measurement noise. Additional noise-mitigation strategies in SINDy include pre-processing or denoising time derivatives before sparse regression. The expressiveness of the candidate library is also crucial; it must be rich enough to represent the true system, but not so large as to encourage overfitting or spurious dynamics. Incorporating domain knowledge into the library improves both interpretability and model identifiability.

SINDy with Control (SINDy-c)

SINDy has been extended to identify systems subject to exogenous inputs and feedback control, a method known as SINDy-c [15]. In this framework, the dynamical system is modeled as

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{3-6}$$

where $\mathbf{u}(t)$ denotes the input or control vector. The candidate function library is augmented to include functions of both states and inputs, denoted $\Theta(\mathbf{x}, \mathbf{u})$. Typical libraries can include all monomials in $\mathbf{x}$ and $\mathbf{u}$ up to a specified degree, as well as cross terms and nonlinearities reflecting physical intuition or prior knowledge.

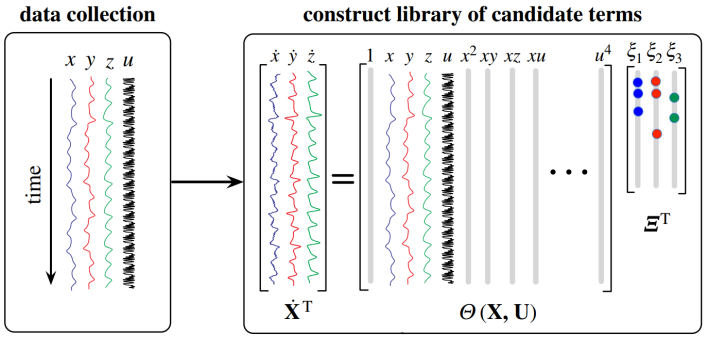


Figure 3-1: Schematic workflow for SINDy with control (SINDy-c) [15]. (**Left**) Data collection: measurements of states $\mathbf{x}$ and control inputs $\mathbf{u}$ over time. (**Right**) Construction of a candidate library $\Theta(\mathbf{x}, \mathbf{u})$ containing functions of both state and input. Sparse regression is used to identify a minimal set of active terms that govern the controlled dynamics.

SINDy-c enables the identification of data-driven models suitable for control design and analysis in systems where actuation or external forcing plays a significant role.

Model Selection: Individual vs. Joint Optimization of Coupled ODEs

When applying SINDy to systems with multiple coupled state variables, such as greenhouse temperature and humidity, an important modeling decision is how to formulate the optimization problem for identifying the underlying ODEs. Two main strategies are commonly used: individual optimization and joint optimization [16, 17]. The mathematical formulation and practical consequences of each approach are outlined below.

Individual Optimization In individual optimization, each ODE is identified independently by defining a separate loss function for each state variable. For example:

$$\mathcal{L}_{T_{\text{air}}} = \sum_{i=1}^{N} \left[\left| \frac{dT_{\text{air}}}{dt} - f_1(T_{\text{air}}, AH, \ldots) \right|^2 + \lambda_1 \left| T_{\text{air}} - \hat{T}_{\text{air}} \right|^2 \right] \tag{3-7}$$

$$\mathcal{L}_{AH} = \sum_{i=1}^{N} \left[\left| \frac{dAH}{dt} - f_2(T_{\text{air}}, AH, \ldots) \right|^2 + \lambda_2 \left| AH - \hat{AH} \right|^2 \right] \tag{3-8}$$

Each loss is minimized independently, resulting in two separate optimization problems. This can be viewed as solving for the parameters $\xi_{T_{\text{air}}}$ and ξ_{AH} in parallel:

$$\xi^*_{T_{\text{air}}} = \arg\min_{\xi_{T_{\text{air}}}} \mathcal{L}_{T_{\text{air}}} \qquad \xi^*_{AH} = \arg\min_{\xi_{AH}} \mathcal{L}_{AH} \tag{3-9}$$

While this approach offers flexibility, it can miss important cross-dependencies. If there is coupling in the true dynamics (e.g., temperature directly affects humidity and vice versa), optimizing independently may ignore or under-represent these effects. As a result, the identified models might contain spurious or missing coupling terms and may fail to capture feedback between variables.

Joint Optimization In joint optimization, a single, global loss function is defined over all state variables, and all ODE parameters are estimated simultaneously:

$$\begin{aligned} \mathcal{L}_{\text{joint}} = \sum_{i=1}^{N} \Bigg[& \left| \frac{dT_{\text{air}}}{dt} - f_1(T_{\text{air}}, AH, \ldots) \right|^2 + \lambda_1 \left| T_{\text{air}} - \hat{T}_{\text{air}} \right|^2 \\ & + \left| \frac{dAH}{dt} - f_2(T_{\text{air}}, AH, \ldots) \right|^2 + \lambda_2 \left| AH - \hat{AH} \right|^2 \Bigg] \end{aligned} \tag{3-10}$$

All coefficients across all ODEs are then jointly optimized:

$$\xi^* = \arg\min_{\xi} \mathcal{L}_{\text{joint}} \tag{3-11}$$

where ξ contains all the parameters for both ODEs. This shared loss couples the parameter estimation processes: gradients for one state's equation may depend on errors in the other, especially through shared or cross-terms.

Mathematical Implications:

- **Parameter coupling:** Joint optimization naturally penalizes models that are accurate for one state but inaccurate for others, resulting in solutions where all variables are fit together and true physical couplings are more likely to be discovered.
- **Optimization:** The loss surface in joint optimization is higher-dimensional and the optimization is performed in a larger parameter space. While this may increase computational cost, it typically results in improved physical consistency.
- **Information sharing:** Robustness in one state, such as temperature which is often better measured and modeled, can regularize the fit for noisier or more weakly modeled states such as humidity, enhancing overall model reliability.

- **Cross-validation:** Validation metrics for each state are jointly optimized, helping to avoid overfitting to one state at the expense of others.

In summary, individual optimization fits each ODE in isolation, which may result in decoupled or unphysical solutions in systems with coupled dynamics. Joint optimization, by fitting all ODEs together with a shared set of parameters, enforces that the model as a whole best explains the data across all variables.

Application in This Thesis: In this thesis, SINDy is used to identify interpretable, physics-informed models for greenhouse air temperature and absolute humidity. A joint optimization strategy is employed to ensure the accurate recovery of both individual and coupled dynamics. Candidate libraries are constructed using physical domain knowledge, and model selection is based on both interpretability and performance on training and test data. This approach provides a reliable foundation for subsequent hybrid modeling and control development.

Software Used. *PySINDy* [18] is an open-source Python package for sparse system identification, used in this thesis to automatically discover parsimonious dynamical models from time-series data. The package offers a modular framework where users can define libraries of candidate nonlinear functions—either built-in (e.g. polynomials, trigonometric terms) or custom—by composing Python `lambda` functions and bundling them into `CustomLibrary` or `GeneralizedLibrary` objects. This enables flexible specification of which input variables appear in each candidate term, allowing domain knowledge to shape the search space and yielding interpretable, physics-informed models.

To identify governing equations from noisy measurements, PySINDy first applies robust numerical differentiation methods such as `SmoothedFiniteDifference` or total variation regularization to estimate time derivatives from experimental data. It then formulates a sparse regression problem, fitting linear combinations of candidate terms to these derivatives using optimizers such as sequential thresholded least squares (STLSQ), SR3, or constrained variants, which promote model parsimony by selecting only a small set of nonzero terms, optionally under explicit coefficient or convexity constraints. Once identified, the governing equations can be automatically exported in several formats, including symbolic strings, callable Python functions, and JIT-compiled code, and the package provides tools for model validation such as prediction accuracy metrics and significance tests. Together, these features enable transparent, automated discovery of compact dynamical models that can be readily analyzed and embedded in larger simulation or control frameworks.

3-3 Background Information on LSTM Networks

Articfical Neural Networks (ANNs) are powerful computational models inspired by the human brain, composed of layers of interconnected neurons that process information and learn complex relationships from data. The learning process in a standard feedforward neural network consists of a sequence of steps: (1) initializing the model weights, (2) feeding forward the input data through the network and applying activation functions at each neuron, (3) computing the loss function based on the difference between the predicted and target outputs, (4) backpropagating the error through the network, and (5) updating the weights using an optimization algorithm such as gradient descent [19, 20, 21].

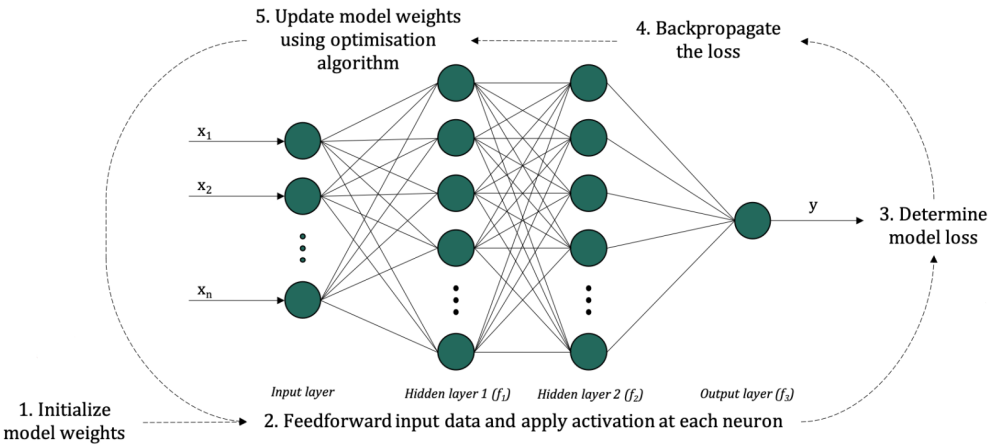


Figure 3-2: General workflow and architecture of a feedforward neural network, illustrating the steps of forward propagation, loss calculation, backpropagation, and weight updates. [22]

In training neural networks, a range of hyperparameters must be specified. These include:

- **Batch size:** The number of samples processed before the model's parameters are updated. Typical values range from 16 to 256, with 32 being a common default.
- **Epochs:** The number of times the entire training dataset is passed forward and backward through the network.
- **Optimizer:** The algorithm used to update the model weights, such as Adam or standard stochastic gradient descent.
- **Loss function:** A metric such as the Mean Squared Error (MSE) that the network attempts to minimize during training.
- **Early stopping:** A regularization technique that halts training if the validation loss does not improve for a set number of epochs, helping to prevent overfitting.
- **Learning rate scheduling:** Methods for adapting the learning rate during training, for example by reducing it when the validation loss plateaus (e.g., using ReduceLROnPlateau).

These hyperparameters strongly influence the convergence speed, model generalization, and the ability to learn complex data patterns.

This structure enables deep networks to model highly nonlinear functions and fit complex data, but in a standard feedforward network, each prediction is based solely on the current input, without memory of previous data points.

Recurrent Neural Networks (RNNs): Adding Memory

Unlike feedforward networks, RNNs are designed to process sequential data by incorporating cycles within their architecture, allowing information to persist across time steps. In an RNN,

the hidden state h_t at each time step t is computed as a function of both the current input x_t and the previous hidden state h_{t-1}:

$$h_t = \phi(W_{hh}h_{t-1} + W_{xh}x_t + b_h), \tag{3-12}$$

where ϕ is a nonlinear activation function, and W_{hh}, W_{xh}, b_h are trainable parameters. This design enables RNNs to capture temporal dependencies and is fundamental for tasks such as time series forecasting, speech recognition, and natural language processing [23].

However, traditional RNNs are limited in their ability to learn long-term dependencies due to the problem of vanishing and exploding gradients during backpropagation through time. In practice, they struggle to retain information over many time steps, especially when the relevant signals are separated by large gaps in the sequence [24].

LSTM Overcoming RNN Limitations

LSTM networks were introduced by Hochreiter and Schmidhuber [24] to address these limitations and to enable learning of long-range dependencies in sequential data. The LSTM architecture augments the standard RNN by introducing a memory cell (c_t) and a system of gates that regulate the flow of information. These mechanisms protect important information from being overwritten or forgotten too soon, thereby preserving gradients and enabling effective training over long sequences [23].

LSTM Cell Structure and Information Flow

An LSTM unit consists of three multiplicative gates: forget, input, and output gates. Each controlling a different aspect of information flow:

- **Forget gate** (f_t): Decides what fraction of the previous cell state to keep.
- **Input gate** (i_t): Regulates the incorporation of new information into the cell state.
- **Output gate** (o_t): Controls how much of the internal cell state should be exposed as output.

At each time step t, the LSTM cell receives the input x_t, previous hidden state h_{t-1}, and previous cell state c_{t-1}, and performs the following computations:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3-13}$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3-14}$$
$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{3-15}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{3-16}$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3-17}$$
$$h_t = o_t \odot \tanh(c_t) \tag{3-18}$$

Here, $\sigma(\cdot)$ is the sigmoid activation function, $\tanh(\cdot)$ is the hyperbolic tangent, and $\odot$ denotes element-wise multiplication. The cell state c_t serves as an internal memory, while h_t is the hidden state passed to the next time step or output layer. This gating mechanism is illustrated in Figure 3-3, which shows how information is selectively forgotten, updated, and exposed by the LSTM unit.

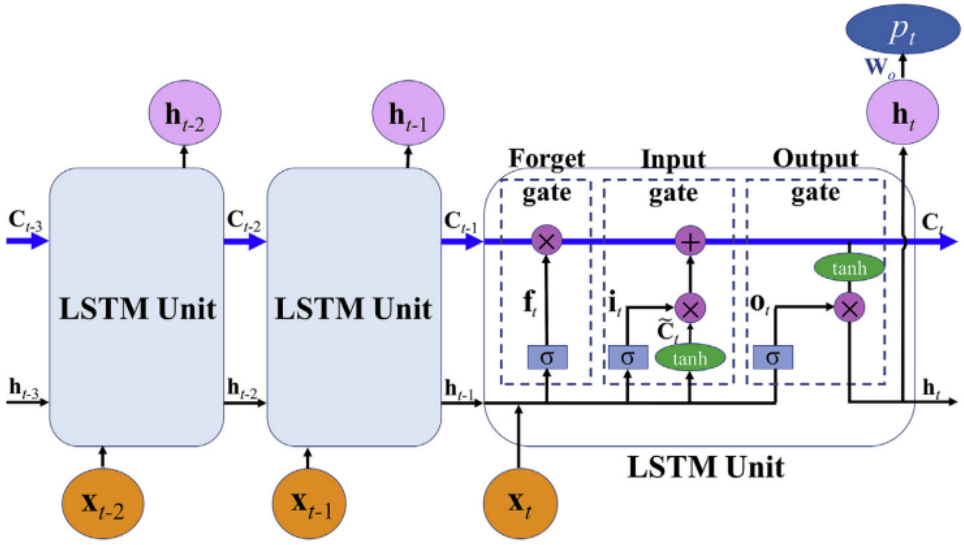


Figure 3-3: LSTM unit architecture. Each unit receives the current input x_t, previous hidden state h_{t-1}, and cell state c_{t-1}, and computes the new cell and hidden states using three gates. [23]

Training LSTM Networks

Training an LSTM network follows the general process of neural network optimization, as with feedforward networks: weights are initialized, data is propagated forward through the network to produce predictions, loss is computed, gradients are backpropagated through time, and weights are updated via an optimizer such as stochastic gradient descent or its variants (e.g., Adam, RMSProp) [23]. Due to the presence of multiple gates and the need to process sequences, LSTMs are typically more computationally intensive and contain more parameters than traditional RNNs.

Recent research has highlighted the importance of proper weight initialization [25, 26] and advanced optimization techniques to stabilize and accelerate LSTM training, especially for long and complex sequences [23].

Application in This Thesis In this thesis, LSTM networks are used as data-driven discrepancy models to complement the physics-informed SINDy framework for greenhouse climate dynamics. After identifying interpretable ODE models for air temperature and absolute humidity, an LSTM network is trained to capture complex residual patterns, such as temporal dependencies, delayed responses, and periodic phenomena, that are not fully represented by the SINDy model. This hybrid approach leverages the LSTM's ability to learn from sequential data. The resulting combined model achieves higher predictive accuracy and robustness, effectively integrating physical domain knowledge with the flexibility of neural network learning.

Furthermore, to enable effective application in real-world greenhouses where labeled data are often scarce, this thesis makes use of transfer learning techniques for the LSTM network. By pretraining the model on large-scale simulated data and then fine-tuning it with limited real-world measurements, transfer learning ensures that the LSTM model adapts to site-specific

conditions while retaining generalizable knowledge. The next section provides an overview of transfer learning and its specific role within the proposed hybrid modeling framework.

Software Used. *TensorFlow* [27] is a widely used open-source framework for numerical computation and machine learning, here serving as the backend for constructing and training deep learning models. Leveraging its high-level Keras API, neural network architectures are specified declaratively by stacking layers such as LSTM units for sequence modeling and dense layers for nonlinear readout. TensorFlow's functional API enables models with multiple parallel input streams—such as system states, controls, and disturbances—to be flexibly combined and processed within a unified network. Model training is supported by a suite of built-in callbacks, including `EarlyStopping` to halt training when validation loss ceases to improve, `ReduceLROnPlateau` for dynamic adjustment of the learning rate, and `ModelCheckpoint` for saving optimal weights during long training runs. The framework also offers comprehensive support for transfer learning: pre-trained models can be loaded without recompilation, further fine-tuned at modified learning rates, or re-used as modules in larger architectures.

3-4 Background Information on Transfer Learning

Transfer learning is a paradigm in machine learning where knowledge acquired from training a model on one task or domain (the *source task*) is repurposed to improve learning or generalization on a different but related task or domain (the *target task*) [28, 29]. This framework is particularly advantageous when data in the target domain are scarce or noisy, but substantial data or computational resources are available for the source domain. Transfer learning enables the reuse of feature representations, parameter initializations, or learned structures from the source, with the assumption that there exist commonalities or shared patterns between the source and target tasks.

A common approach within transfer learning is *fine-tuning*. In fine-tuning, a neural network is first pretrained on the source task, often using a large and diverse dataset [30, 31]. The pretrained model is then further trained (fine-tuned) on the target task using its (usually smaller) dataset. During this stage, some or all of the model's parameters are updated to better accommodate the specific characteristics of the target domain, while retaining the knowledge encoded during pretraining [28]. Fine-tuning often results in faster convergence, improved accuracy, and increased robustness.

Application in This Thesis In this thesis, transfer learning is employed to bridge the domain gap between simulated and real-world greenhouse climate data. Initially, the LSTM discrepancy model is pretrained on abundant simulated data, enabling it to learn generic temporal dependencies and variability patterns present in the climate dynamics. Subsequently, the pretrained LSTM model undergoes fine-tuning on a limited set of real-world greenhouse measurements. This two-stage process allows the model to adapt its parameters to unique environmental, structural, or operational features specific to the actual greenhouse system, while retaining generalizable patterns learned from simulation. The use of transfer learning thus maximizes the value of limited real-world data and enhances the model's ability to generalize to previously unseen or rare operational scenarios.

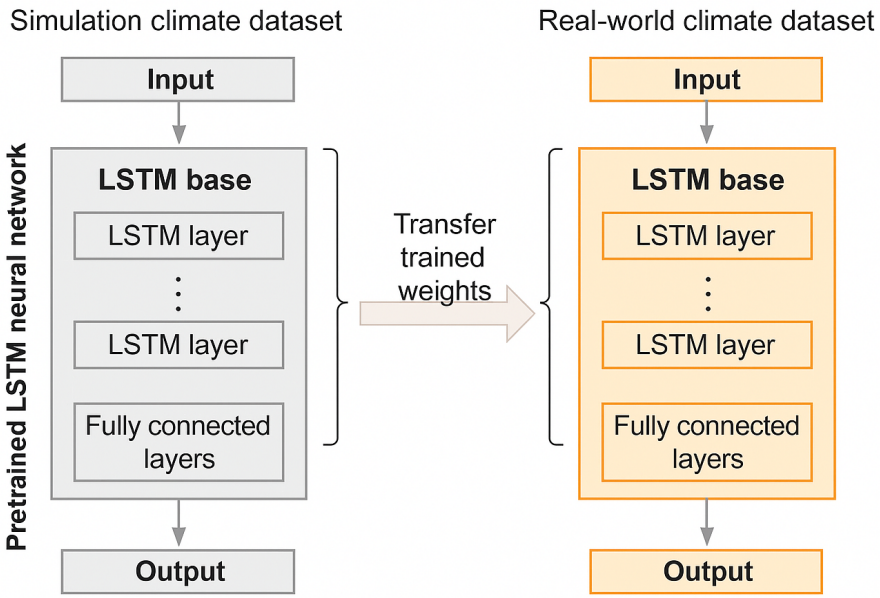


Figure 3-4: Schematic illustration of the transfer learning process for greenhouse climate modeling. An LSTM network is first pretrained on simulated (source) climate data and then fine-tuned using real-world (target) greenhouse measurements, facilitating adaptation to the specific characteristics of the deployment environment.

3-5 Background Information on MPC

MPC is an advanced, optimization-based control methodology widely used in modern process industries, robotics, and building automation. Unlike classical feedback controllers, which compute control actions based solely on the current state, MPC utilizes a dynamic model of the system to predict future evolution and to optimize a sequence of control inputs over a receding time horizon [32, 33]. This predictive framework enables MPC to handle Multi-Input Multi-Output (MIMO) systems with hard constraints on states and actuators, and to proactively anticipate disturbances or changing setpoints.

Mathematical Formulation

At each control interval k, MPC solves a finite-horizon optimal control problem of the form:

$$\min_{\{u_{j|k}\}_{j=0}^{N_p-1}} \quad \sum_{j=0}^{N_p-1} \left(\|y_{j|k} - r_{j|k}\|_Q^2 + \|u_{j|k} - u_{j|k}^{\text{ref}}\|_R^2 \right) \tag{3-19}$$

$$\text{s.t.} \quad x_{j+1|k} = f(x_{j|k},\, u_{j|k},\, d_{j|k}), \qquad x_{0|k} = x_k \tag{3-20}$$

$$x_{j|k} \in X, \quad u_{j|k} \in U, \quad x_{N_p|k} \in X_N \tag{3-21}$$

where:

- x_k is the current state of the system at time k.

- $u_{j|k}$ denotes the input at predicted time $k+j$, computed at time k.
- $u_{j|k}^{\text{ref}}$ is the reference control input at prediction step $k+j$.
- $y_{j|k}$ is the predicted output.
- $r_{j|k}$ is the reference trajectory.
- $f(\cdot)$ is the (possibly nonlinear) system model, mapping state, input, and disturbance to the next state.
- Q and R are positive semi-definite weighting matrices for tracking and actuation effort.
- X, U, and X_N define admissible state, input, and terminal state sets.

After solving the optimization, only the first input $u_{0|k}^*$ of the optimal control sequence is applied. At the next step, the state is remeasured, the horizon is shifted forward (receding horizon principle), and the optimization is repeated. This rolling optimization introduces closed-loop feedback and allows the controller to adapt to model mismatches and disturbances in real time.

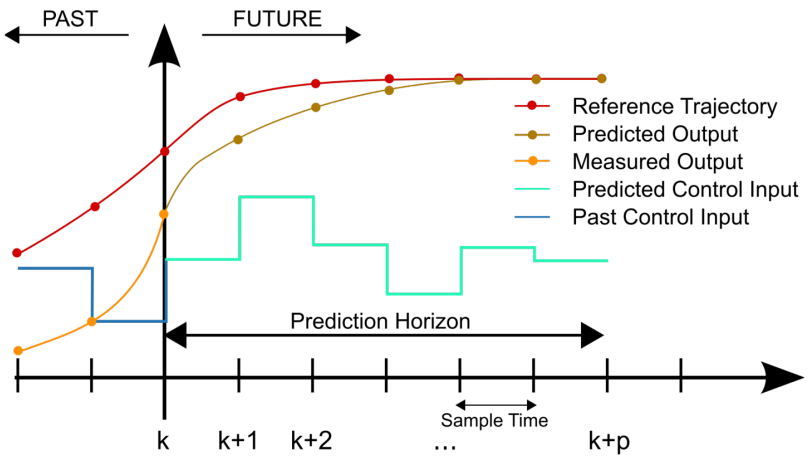


Figure 3-5: Receding horizon MPC: At each time step, the controller predicts system outputs and control inputs over a finite horizon, optimizing the sequence to track a reference while respecting constraints. Only the first input is applied before re-optimizing at the next step. Adapted from [34].

Advantages and Practical Considerations

MPC provides several advantages over conventional control approaches:

- **Constraint handling:** MPC incorporates operational and safety constraints on both states and actuators directly into the control synthesis, ensuring safe and feasible operation.
- **Multivariable control:** MPC can simultaneously coordinate multiple, possibly interacting actuators to achieve multiple control objectives.

- **Proactive adjustment:** By explicitly forecasting system dynamics and anticipated disturbances, MPC makes proactive adjustments, improving performance under time-varying conditions.
- **Flexibility:** MPC can be readily extended to nonlinear models, time-varying references, and to integrate external forecasts.

These properties make MPC particularly attractive for applications such as building climate control, chemical processes, and advanced robotics, where safe, robust, and efficient operation is critical.

Application in This Thesis In this thesis, MPC serves as the supervisory control strategy for greenhouse climate management. The developed hybrid model, which combines physics-based and data-driven elements, is used as the predictive model within the MPC framework. The controller determines optimal control actions, such as heating, ventilation, and shading, by solving a constrained optimization problem over a finite prediction horizon. The MPC framework systematically accounts for operational constraints, setpoint tracking, and actuator limitations, enabling adaptive and efficient climate regulation under varying conditions.

Software Used. *GEKKO* [35] is a Python library that translates systems of algebraic and differential equations into a nonlinear programme, solved with IPOPT or APOPT. Model entities are declared with high-level objects: manipulated variables (`MV`) for actuators, controlled variables (`CV`) for measured states, fixed variables (`FV`) for parameters, and `Param` for any time-series data (e.g. weather forecasts or LSTM corrections). Dynamics are added with symbolic `Equation` statements.

For MPC, GEKKO provides two key settings. First, rate-of-change penalties are assigned through the `DCOST` attribute: the solver adds Δu^2 to the objective, discouraging large moves between consecutive control actions. Second, setting `IMODE = 6` engages the built-in receding-horizon routine, which automatically shifts the time grid, updates measurements, re-optimises, and applies the first control move every cycle. Internally, in MPC mode, GEKKO discretises differential equations using orthogonal collocation on finite elements, effectively translating continuous dynamics into algebraic constraints equivalent to implicit Runge–Kutta numerical integration. The resulting collocated equations and objective are then assembled into a sparse nonlinear programme solved by IPOPT (selected by `SOLVER = 3`). GEKKO automatically manages mesh shifting and solver warm-starting with each MPC iteration, enabling a fully declarative MPC interface where numerical integration, horizon updates, and nonlinear optimisation are transparently handled by the software.

Chapter 4

Data Generation and Processing

This chapter details the origin, characteristics, and pre-processing of the datasets used for modeling and control in this thesis. Both high-fidelity simulation data and real-world greenhouse measurements are included to enable robust model development and evaluation.

Simulation data are generated using the Tap greenhouse model, as described in Chapter 2, coupled with a receding-horizon MPC. Two distinct 90-day simulation datasets, based on different years of real weather data, serve as training and test sets for model development and validation. In addition, three real-world datasets from Dutch tomato greenhouses are used for model transfer and evaluation under realistic conditions.

To ensure data quality and improve model performance, all datasets undergo systematic outlier detection and smoothing procedures prior to model training. The following sections describe the data generation process, real-world datasets, and the pre-processing pipeline in detail.

4-1 Simulation Data Generation

High-fidelity simulation data are generated by integrating the Tap greenhouse simulator [10] with a receding-horizon MPC. Two 90-day simulation datasets are created: one using April–June 2024 weather data for model training, and another using April–June 2023 weather data for testing and validation. At each MPC timestep ($\Delta t = 300$ s), the controller optimizes control actions over a 3-hour prediction horizon ($N_p = 36$), relying on forecasts of outside temperature, solar radiation, and absolute humidity, with wind speed fixed at 2 m/s. The first calculated control actions, vent positions, screen position, and heating-pipe temperature, are then applied to the Tap simulator, which advances with a 30 s internal timestep. All simulated states and control inputs are logged at a 5-minute resolution.

To ensure idealized data for model development, the Tap simulator is also used as the internal prediction model within the MPC. This removes any model-plant mismatch, ensuring that the simulation data have zero setpoint tracking error and complete state observability.

Setpoints for air temperature are generated using the Radiation–Temperature Ratio (RTR) strategy, which is inspired by the Plant Empowerment framework [36]. This method links average daily temperature targets to the cumulative daily sum of solar radiation, reflecting a key crop physiological relationship:

$$T_{\text{avg}} = T_{\text{base}} + \text{RTR} \cdot \frac{\sum_{k=1}^{288} R_k \Delta t}{1000},$$

where $T_{\text{base}} = 18\,^{\circ}\text{C}$ and $\text{RTR} = 2$. Daily average setpoints are mapped to full 24-hour profiles via piecewise-linear day–night transitions, with additional Gaussian smoothing ($\sigma = 30$ samples) to ensure gradual setpoint changes. The resulting profiles (illustrated in Figure 4-1) span 14–28 °C, with the lower range applied during periods of low radiation.

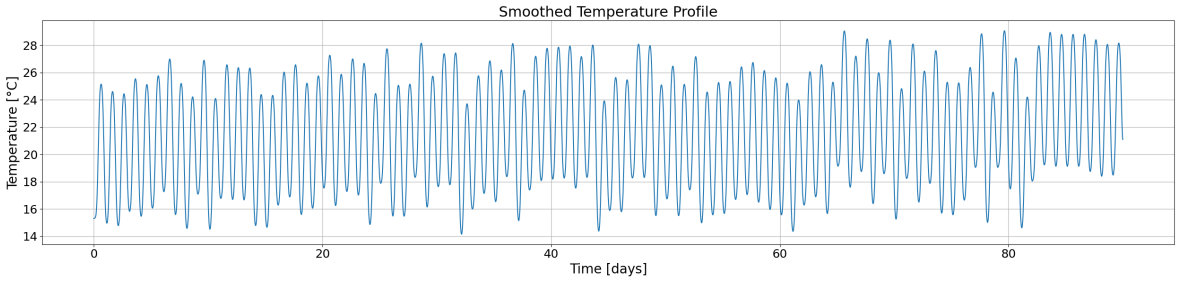


Figure 4-1: Smoothed air temperature setpoint profile generated by the RTR strategy over 90 days using the training dataset.

Relative Humidity (RH) is maintained between 60% and 90% through automatic ventilation and screen control. The heating-pipe temperature is constrained to always exceed air temperature, supporting realistic heat transfer dynamics.

Key assumptions and value ranges for the simulation are as follows:

- **Crop processes:** Mature tomato crop, with net leaf biomass of approximately 400–450 g/m^2.
- **External weather:** Realistic Dutch weather data: outside temperature T_{out} ranges from 2–32 °C, global radiation from 0–1000 W m^{-2}, outside absolute humidity from 5–25 g m^{-3}, and wind speed fixed at 2 m/s.
- **Simulated variable ranges:**
  - Air temperature: $T_{\text{air}} \in [14, 28]$ °C
  - Pipe temperature: $T_{\text{pipe}} \in [20, 80]$ °C
  - Vent/screen position: 0–100%
  - Internal absolute humidity: $\text{AH}_{\text{in}} \in [5, 20]$ g m^{-3}

4-2 Real-World Datasets

In addition to simulated data, three real-world datasets are used for external validation and transfer learning:

- **Hoogendoorn Customer A:** May 2024–May 2025
- **Hoogendoorn Customer B:** May 2024–May 2025
- **TomatoWorld:** May 2024–May 2025

All datasets originate from commercial Dutch tomato greenhouses and are sampled at a 5-minute resolution. Each dataset covers a full annual cycle, capturing a wide range of environmental and operational conditions relevant for greenhouse climate modeling.

4-3 Data Processing

Prior to model training, all datasets undergo systematic cleaning and smoothing to improve robustness and numerical stability, especially for data-driven models such as LSTM networks.

- **Outlier removal:** For each variable (except wind speed), outliers are identified using a z-score threshold of 3.0. Outlier values are set to NaN and subsequently imputed by linear interpolation, minimizing the risk of propagating spurious values through the modeling pipeline.
- **Smoothing:** After outlier removal, a Savitzky–Golay filter (window length = 5, polynomial order = 3) is applied to each time series. This technique reduces high-frequency measurement noise while preserving underlying trends and signal shape.

Both the cleaned and the smoothed series for each variable are retained and made available for subsequent model training and evaluation. This two-step data processing workflow ensures that the resulting datasets are suitable for robust model identification, accurate forecasting, and reliable control synthesis.

Chapter 5

Physics-Informed SINDy Model

This chapter details the development and validation of a physics-informed SINDy model for greenhouse climate dynamics, with a focus on the coupled evolution of air temperature and absolute humidity. The modeling approach leverages both physical insight and data-driven methods to identify interpretable ODEs suitable for control design.

The modeling procedure follows four main steps:

1. Construction of candidate function libraries for both absolute humidity and air temperature, based on physical insight and domain knowledge.

2. Numerical differentiation of observed state trajectories to obtain target derivatives for model fitting.

3. Fitting the SINDy model to simulation data using a sparsity-promoting optimizer, with hyperparameter tuning to balance model accuracy and parsimony.

4. Evaluation of model accuracy and complexity on both training and test data, leading to the selection of a final model for control-oriented applications.

The structure of this chapter is as follows. Section 5-1 and Section 5-2 describe the construction of the candidate function libraries for absolute humidity and air temperature, respectively, each based on an analysis of the dominant physical mechanisms in the greenhouse. Section 5-3 presents the model training procedure, including numerical differentiation, hyperparameter tuning, and joint optimization of the coupled ODEs. Section 5-4 evaluates the final model's performance on both training and independent test datasets, reporting on predictive accuracy, robustness, and generalization across different seasons.

Residual modeling errors due to unmodeled or complex dynamics are subsequently addressed in the next chapter using a data-driven discrepancy modeling approach, which augments the physics-informed model with a neural network correction.

5-1 Physics-Informed Candidate Library for Absolute Humidity

The first step in constructing the SINDy model is to analyze the underlying physical mechanisms that govern absolute humidity in the greenhouse. The high-fidelity Tap model describes the dynamics of absolute humidity with an ODE comprising three core processes: ventilation, transpiration, and condensation. By decomposing the ODE and examining the contribution of each process using simulation data, physically interpretable candidate functions are selected for inclusion in the SINDy library.

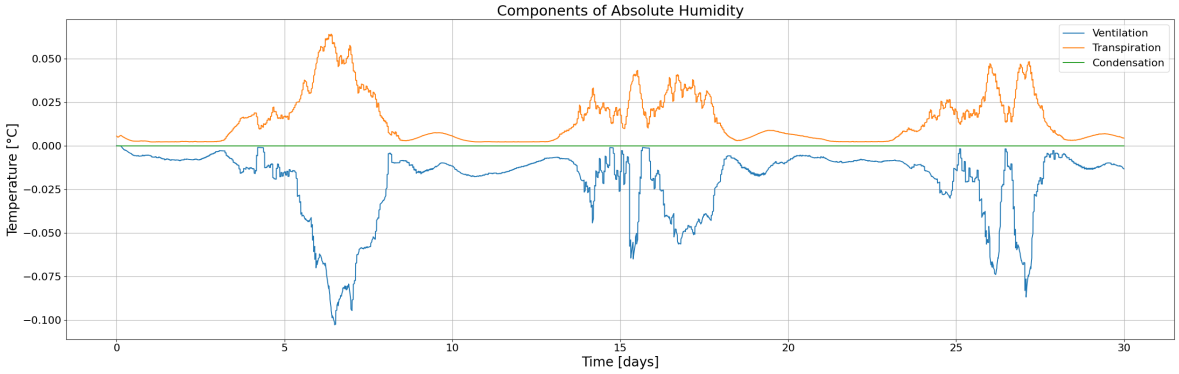


Figure 5-1: Simulated contributions of ventilation, transpiration, and condensation to the absolute humidity ODE.

Figure 5-1 shows that the contribution of condensation is negligible, this is consistent across different weather conditions. Condensation is therefore omitted from the candidate library and any remaining effects are addressed in the discrepancy model described in the following chapter. However, ventilation and transpiration are significant and further analyzed.

Ventilation: Ventilation drives the exchange of moisture between indoor and outdoor air, represented in the Tap model as:

$$\Phi(AH_{\text{out}} - AH_{\text{in}}), \tag{5-1}$$

where the ventilation rate is a function of window positions and wind speed (d_{Wind}):

$$\Phi = \left(\frac{a_1 \text{Vent}_{\text{lee}}}{1 + a_2} + a_3 + a_4 \text{Vent}_{\text{wind}}\right) d_{\text{Wind}}. \tag{5-2}$$

Neglecting the constants, the ventilation rate can be expressed as a weighted sum of the following terms:

- $\text{Vent}_{\text{lee}}\, d_{\text{Wind}}\, (AH_{\text{out}} - AH_{\text{in}})$
- $\text{Vent}_{\text{wind}}\, d_{\text{Wind}}\, (AH_{\text{out}} - AH_{\text{in}})$
- $d_{\text{Wind}}\, (AH_{\text{out}} - AH_{\text{in}})$

An additional candidate is the sum of all vents, $(\text{Vent}_{\text{lee}} + \text{Vent}_{\text{wind}})\, d_{\text{Wind}}\, (AH_{\text{out}} - AH_{\text{in}})$. All terms are included as candidate functions and are appropriately scaled to prevent the optimizer from favoring numerically dominant terms.

Transpiration: Transpiration is the process by which plants lose water vapor through their stomata, transferring moisture from the leaf interior into the greenhouse air. In the Tap model, the transpiration rate is modeled as:

$$\text{Trans\_rate} = \frac{DLW \cdot (SWP \cdot RCL + c_1 \cdot \text{VPD})}{EEW \cdot (SWP + c_2 + \frac{c_3}{LC})} \tag{5-3}$$

where DLW (dryLeafWeight) is the effective crop leaf area, SWP (saturatedWaterPressure) is a temperature-dependent function, RCL (radiationCropLevel) quantifies the available radiation at the crop, VPD is the vapor pressure deficit, EEW (evaporationEnergyWater) is the latent heat of evaporation, and LC (leafConductance) characterizes vapor transfer from leaf to air. The constants c_1, c_2, and c_3 are determined by physical properties and crop parameters.

Simulation analysis and decomposition (see Figure 5-2) highlight the primary drivers of transpiration. Radiation at crop level and air temperature are especially influential, while VPD and leaf conductance also play significant roles. The denominator is typically dominated by saturated water pressure under greenhouse conditions, and the evaporation energy can be considered approximately constant. As such, these simplifications guide the selection of physically meaningful candidate functions.

The key physical variables, each with a clear mechanistic interpretation and potential for inclusion in the candidate library, are:

$$SWP = 1.8407 \cdot 10^{-4}\, T_{\text{air}}^2 + 9.7838 \cdot 10^{-4}\, T_{\text{air}} + 0.051492 \tag{5-4}$$

$$RCL = 0.68\, I_{\text{glob}}\, (1 - 0.3\, E_{\text{screen}}) \tag{5-5}$$

$$\text{SVP} = 610.78 \exp\left(\frac{17.27\, T_{\text{air}}}{T_{\text{air}} + 237.3}\right) \tag{5-6}$$

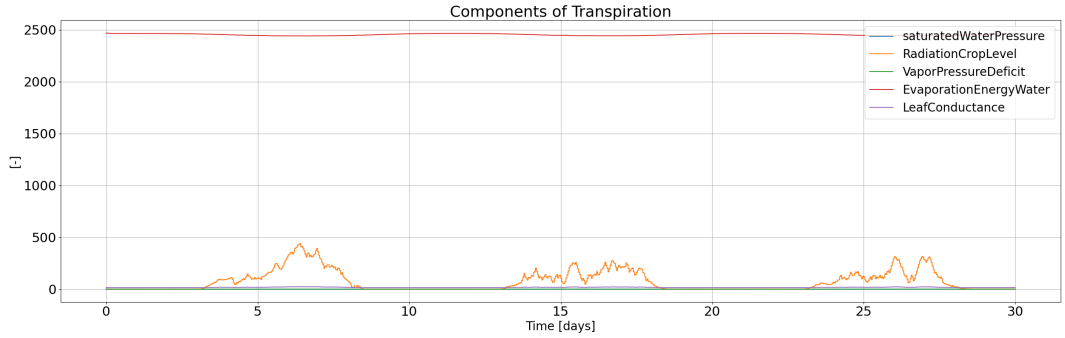
$$\text{VP} = \frac{AH_{\text{in}}\, p_{\text{atm}}}{0.622 + AH_{\text{in}}} \tag{5-7}$$

$$\text{VPD} = \text{SVP} - \text{VP} \tag{5-8}$$

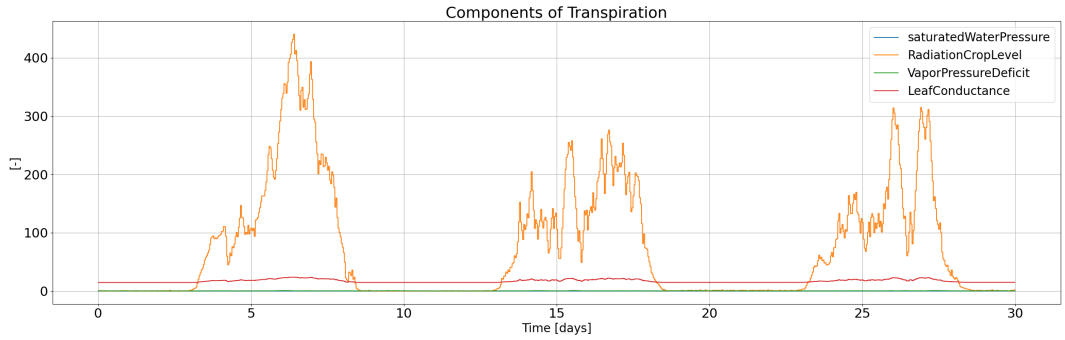
$$LC = 20.3\, (1 - 0.44\, \exp(-2.5 \cdot 10^{-3}\, I_{\text{glob}})) \tag{5-9}$$

where T_{air} is in degrees Celsius, I_{glob} is incoming global radiation, Escreen is the screen position, AH_{in} is the internal absolute humidity, and p_{atm} is atmospheric pressure.

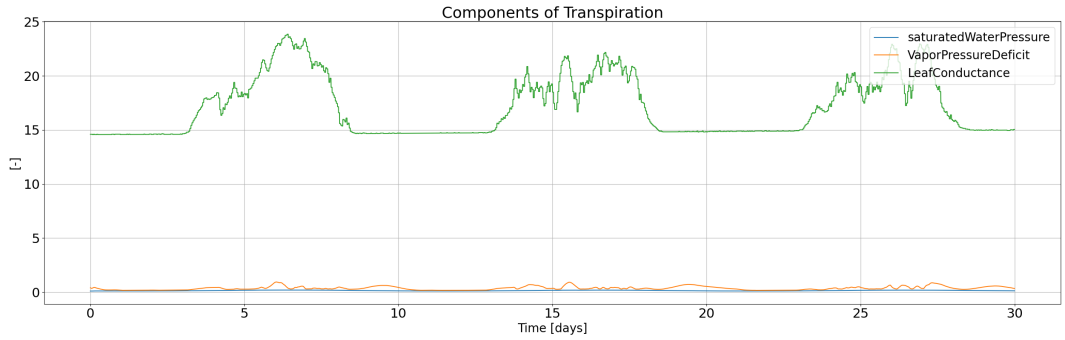
Figure 5-2 shows the relative influence of these components: removing radiation, for example, greatly reduces predicted transpiration, while omitting leaf conductance or the energy term has more subtle effects.



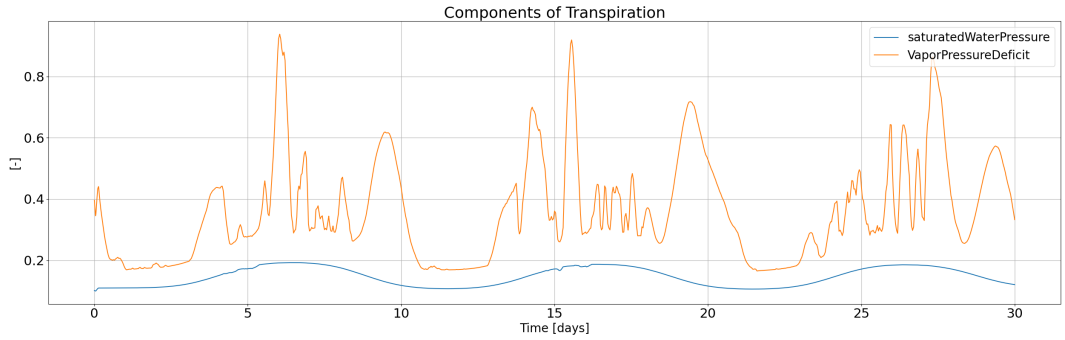
(a) All transpiration terms.



(b) Without evaporation energy.



(c) Without radiation term.



(d) Without leaf conductance.

Figure 5-2: Stepwise decomposition of the transpiration ODE terms: (a) all terms, (b) without evaporation energy, (c) without radiation, (d) without leaf conductance.

Based on this physical analysis and the ODE structure, the candidate function library for absolute humidity includes terms representing:

- Air temperature and its nonlinear effects: T_{air}, T_{air}^2
- Radiation effects: I_{glob}, $I_{\text{glob}}\,(1 - 0.3\,E_{\text{screen}})$, $I_{\text{glob}}\,E_{\text{screen}}$
- Humidity and temperature interaction: $AH_{\text{in}}\,(T_{\text{air}} + 273.15)$
- Nonlinearities: $\exp(c\,T_{\text{air}})$, $1/\exp(c\,I_{\text{glob}})$
- Vapor pressure deficit: VPD, as constructed above
- Interaction terms: $T_{\text{air}}\,I_{\text{glob}}$, $T_{\text{air}}^2\,I_{\text{glob}}$, $(T_{\text{air}} + T_{\text{air}}^2)\,I_{\text{glob}}\,(1 - 0.3\,E_{\text{screen}})$, $T_{\text{air}}\,I_{\text{glob}}\,E_{\text{screen}}$

These functions directly represent the main drivers of greenhouse transpiration, as supported by both model structure and simulation results. This ensures that the SINDy model can capture the essential effects of environmental control and plant response on absolute humidity dynamics.

5-2 Physics-Informed Candidate Library for Air Temperature

The development of a physics-informed SINDy model for greenhouse air temperature begins with a detailed analysis of the physical mechanisms represented in the Tap model. The ODE for T_{air} includes contributions from ventilation, heating, convection, soil, radiation, lighting, evaporation, and condensation. The contribution of each process over a representative three-day simulation is shown in Figure 5-3.

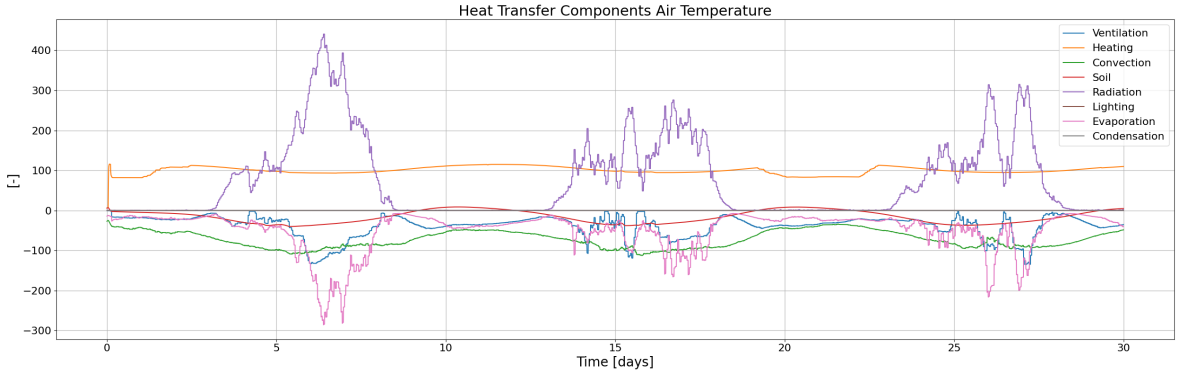


Figure 5-3: Simulated contributions of all heat transfer mechanisms to the air temperature ODE. Radiation and evaporation are the largest contributors.

Figure 5-3 highlights that *radiation* and *evaporation* are the dominant drivers of air temperature dynamics. To clarify the influence of other processes, these two terms are omitted in Figure 5-4.

The effect of condensation is again negligible and is therefore omitted from the candidate library. The soil contribution shows a small, lagged effect that closely follows air temperature and is difficult to measure in practice. Therefore, it is also omitted from the SINDy

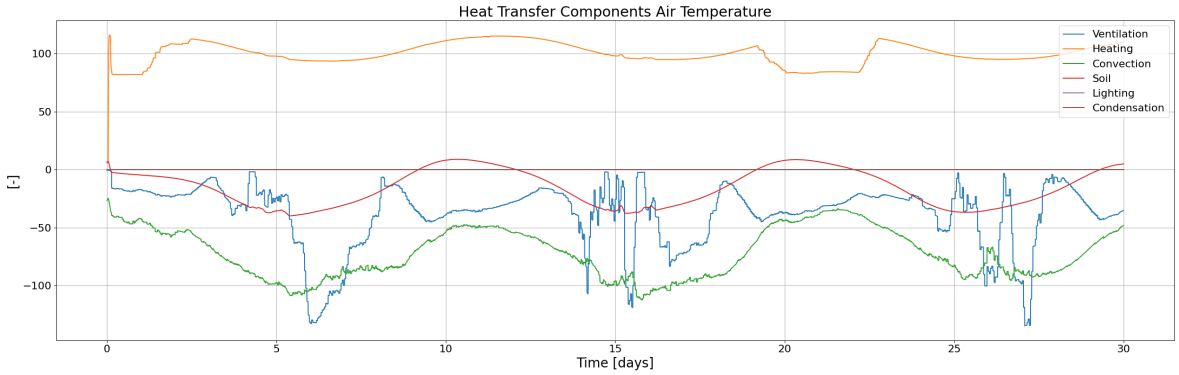


Figure 5-4: Heat transfer mechanisms contributing to air temperature, with radiation and evaporation omitted for clarity. Ventilation, convection, and heating have significant, dynamic contributions; soil and condensation play minor roles.

model. Consequently, the candidate library for T_{air} primarily includes ventilation, heating, convection, radiation, and evaporation terms.

Ventilation: Ventilation governs heat exchange between indoor and outdoor air. In the Tap model this is represented as

$$\dot{Q}_{\text{vent}} = \Phi \, \rho_{\text{air}} \, c_p \, (T_{\text{out}} - T_{\text{in}}), \tag{5-10}$$

with the ventilation rate Φ given by (5-2). Disregarding constants, this motivates the following candidate functions:

- $\text{Vent}_{\text{lee}} \, d_{\text{Wind}} \, (T_{\text{out}} - T_{\text{in}})$
- $\text{Vent}_{\text{wind}} \, d_{\text{Wind}} \, (T_{\text{out}} - T_{\text{in}})$
- $d_{\text{Wind}} \, (T_{\text{out}} - T_{\text{in}})$

An additional candidate is the sum of all vents, $(\text{Vent}_{\text{lee}} + \text{Vent}_{\text{wind}}) \, d_{\text{Wind}} \, (T_{\text{out}} - T_{\text{in}})$.

Heating: The sensible heat supplied by the pipe network is described by the following expression, which models heat transfer between the pipe surface and the greenhouse air:

$$\dot{Q}_{\text{PipeHeating}} = \frac{\alpha A_p (T_{pipe} - T_{air})}{A_{\text{greenhouse}}} \tag{5-11}$$

where

- α is the heat transfer coefficient between the pipe and the air $[\text{W m}^{-2}\text{K}^{-1}]$,
- A_p is the total surface area of the heating pipes $[\text{m}^2]$,
- $A_{\text{greenhouse}}$ is the total greenhouse floor area $[\text{m}^2]$.

Ignoring constants, this yields the candidate function:

$$T_{pipe} - T_{air}.$$

Convection: Convective exchange between the interior air and the roof surface is described by

$$\dot{Q}_{\text{conv}} = h_{\text{roof}} \, (T_{\text{out}} - T_{\text{air}}), \tag{5-12}$$

with

- h_{roof} the overall heat-transfer coefficient of the roof [W m^{-2} K^{-1}].

A suitable candidate term is therefore $T_{\text{out}} - T_{\text{air}}$.

Radiation: Direct solar gain inside the greenhouse (attenuated by roof and, when present, the screen) is

$$\dot{Q}_{\text{rad}} = I_{\text{glob}} \, \tau_{\text{roof}} \Big[1 - E_{\text{screen}} + E_{\text{screen}} \, \tau_{\text{scr}} \Big]. \tag{5-13}$$

Relevant candidate functions are

- I_{glob}
- $I_{\text{glob}} \, E_{\text{screen}}$
- $I_{\text{glob}} \, (1 - E_{\text{screen}})$
- $I_{\text{glob}} \, [1 - 0.3 \, E_{\text{screen}}]$

Evaporation: Latent cooling due to plant transpiration and misting is given by

$$\dot{Q}_{\text{evap}} = -L_{\text{v}} \, (\dot{m}_{\text{trans}} + \dot{m}_{\text{mist}}), \tag{5-14}$$

where

- L_{v} is the latent heat of vaporisation of water ($\approx 2.45 \cdot 10^6$ J kg^{-1} at 20 °C).
- $\dot{m}_{\text{trans}}$ is the water-vapour mass-flux from crop transpiration [kg s^{-1}].
- $\dot{m}_{\text{mist}}$ is the mass-flux from misting systems [kg s^{-1}].

The same candidate functions devised for transpiration in the absolute-humidity library (Section 5-1) apply here; misting is neglected in this study.

Based on this analysis, the SINDy candidate library for air temperature is constructed from physically motivated functions representing each of the dominant heat transfer mechanisms:

- $\text{Vent}_{\text{lee}} \, d_{\text{Wind}} \, (T_{\text{out}} - T_{\text{in}})$
- $\text{Vent}_{\text{wind}} \, d_{\text{Wind}} \, (T_{\text{out}} - T_{\text{in}})$
- $T_{\text{pipe}} - T_{\text{air}}$
- $T_{\text{out}} - T_{\text{air}}$
- $I_{\text{glob}} \cdot (1 - 0.3 \cdot E_{\text{screen}})$
- Evaporation-related terms (see Section 5-1)

All terms are scaled according to their magnitudes.

5-3 Model Training

The SINDy framework is used to identify interpretable models from the candidate libraries defined for air temperature and absolute humidity. Model training consists of three main steps: (1) constructing a design matrix from the selected candidate functions, (2) differentiating the observed state trajectories to obtain the target derivatives, and (3) solving a regularized sparse regression problem to identify the governing equations.

Data Preparation and Differentiation: To estimate the time derivatives required for model fitting, a smoothed finite difference approach is applied to the time series data:

- **Smoothing and Differentiation:** A Savitzky–Golay filter (window length 15, polynomial order 2) is used to smooth the measured time series and estimate $\frac{dx}{dt}$.
- **Time Vector Alignment:** The time vector is constructed to match the sampling interval of the measurements, ensuring that each derivative estimate corresponds to the correct input features.
- **Feature Construction:** Each input variable is mapped to its corresponding candidate functions, resulting in a design matix.

Training both ODEs jointly requires careful tuning of the optimizer's hyperparameters to achieve the best balance between model complexity and predictive accuracy. In particular, the STLSQ optimizer relies on two critical parameters: the sparsity threshold, which determines the minimum size for coefficients to be retained, and the regularization parameter α, which penalizes large coefficients and controls overfitting.

To systematically explore the trade-off between sparsity and accuracy, a grid search was performed across a logarithmic range of both parameters (thresholds from 10^{-10} to 10^{0} and α values from 10^{-10} to 10^{10}). For each parameter combination, the total model complexity and the combined R^2 score on held-out validation data were recorded.

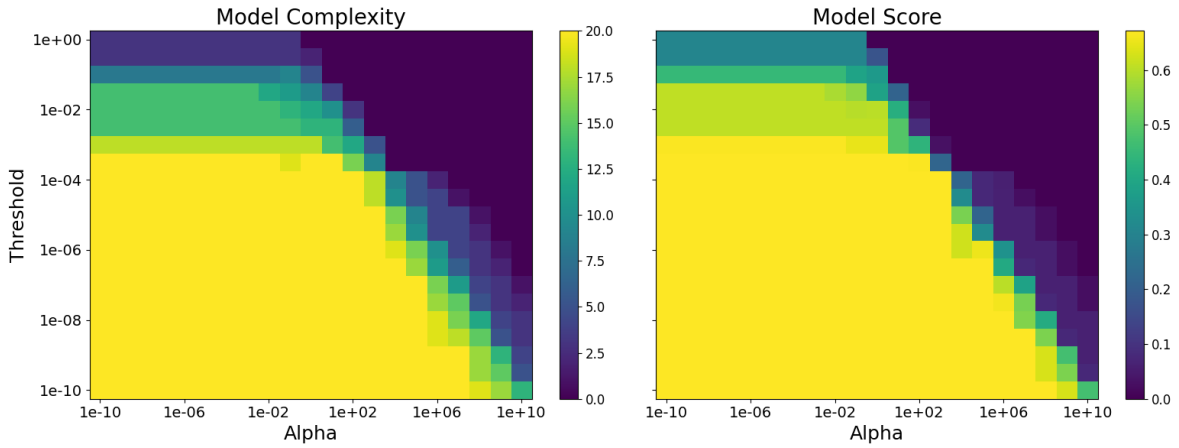


Figure 5-5: Double heatmap of (left) model complexity (number of nonzero terms) and (right) model score (R^2) as functions of threshold and α in the grid search.

Figure 5-5 presents a double heatmap, with model complexity (number of nonzero terms) shown on the left and model score (R^2) on the right, as functions of threshold and α. In the lower-left region (low threshold, low α), the model includes nearly all candidate terms and achieves the highest fit, but risks overfitting and reduced interpretability. Conversely, the upper-right region (high threshold, high α) yields highly sparse, sometimes trivial models with poor accuracy. Between these extremes, a clear diagonal transition zone emerges where sparsity and predictive power are balanced: only the most informative terms are retained, while redundant or noisy terms are set to zero.

The trade-off between sparsity and accuracy is further illustrated in Figure 5-6, which shows the R^2 score as a function of model complexity for all models found during the grid search. Each point corresponds to a model trained with a unique set of parameter values, and the blue trend line highlights the region where the highest accuracy is achieved for a given complexity. Although the maximum combined model score is approximately 0.7, achieved with a model complexity of 14 nonzero terms, the final model selected has a lower complexity of 11 and a model score of 0.56.

This choice reflects a critical trade-off, while including all candidate terms can maximize R^2 on validation data, it also produces an overparameterized model that lacks physical interpretability and may encode spurious relationships. Such models are unsuitable for control frameworks, as excess terms can introduce nonphysical behavior and obscure the control law. By selecting the model with a complexity of 11, the resulting ODEs retain only the relevant physical processes, reflect the expected model structure, and ensure interpretability, robustness, and suitability for use in model-based control. This systematic tuning procedure ensures that the final SINDy models are not only accurate and generalizable, but also remain parsimonious and physically meaningful.

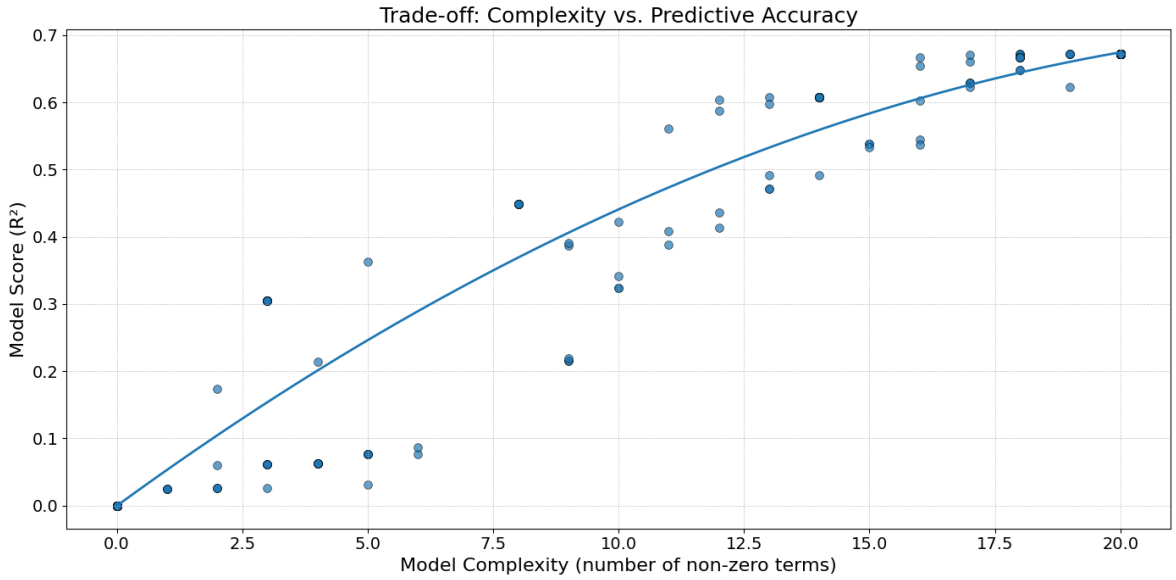


Figure 5-6: Trade-off between model complexity and predictive accuracy. Each point corresponds to a unique combination of α and threshold, and the blue line represents the trend line.

The final SINDy model selected for greenhouse climate dynamics is given by:

$$\begin{aligned}\frac{dT_{\text{air}}}{dt} &= 0.76\,\text{Vent}_{\text{lee}}\,d_{\text{Wind}}\,(T_{\text{out}} - T_{\text{air}}) + 0.32\,\text{Vent}_{\text{wind}}\,d_{\text{Wind}}\,(T_{\text{out}} - T_{\text{air}}) \\ &\quad + 0.10\,(T_{\text{pipe}} - T_{\text{air}}) + 0.65\,(T_{\text{out}} - T_{\text{air}}) + 0.93\,\text{Vent}_{\text{wind}}\,(AH_{\text{out}} - AH_{\text{in}})\,d_{\text{Wind}} \\ &\quad + 2.87\,I_{\text{glob}}\,(1 - 0.3\,(E_{\text{screen}}/100)),\end{aligned}$$

$$\begin{aligned}\frac{d\,AH_{\text{in}}}{dt} &= -0.25\,\text{Vent}_{\text{lee}}\,d_{\text{Wind}}\,(T_{\text{out}} - T_{\text{air}}) + 0.06\,\text{Vent}_{\text{wind}}\,d_{\text{Wind}}\,(T_{\text{out}} - T_{\text{air}}) \\ &\quad + 2.75\,\text{Vent}_{\text{lee}}\,(AH_{\text{out}} - AH_{\text{in}})\,d_{\text{Wind}} + 1.79\,\text{Vent}_{\text{wind}}\,(AH_{\text{out}} - AH_{\text{in}})\,d_{\text{Wind}} \\ &\quad + 3.80\,I_{\text{glob}}\,(1 - 0.3\,(E_{\text{screen}}/100)).\end{aligned}$$

The final model structure reveals several key features of greenhouse climate dynamics. The dominant components in both ODEs are associated with ventilation and solar radiation, which is consistent with established physical understanding for a ventilated greenhouse. Ventilation terms-, both temperature- and humidity-driven-, are present in both the air temperature and absolute humidity equations, illustrating the coupled nature of heat and moisture exchange through air movement. The consistent inclusion of global radiation terms in both equations highlights the role of solar energy as a primary driver of temperature and humidity dynamics.

No direct transpiration terms are present in the final model, despite their theoretical relevance. This outcome indicates that transpiration effects may be indirectly represented through interactions with radiation and ventilation, or that these effects are not sufficiently distinct in the available data to be isolated by the sparse identification process. This finding reflects both the strengths and the limitations of the SINDy approach: while it yields interpretable and physically consistent models, processes that are less directly observable or are highly correlated with other mechanisms may be underrepresented.

Overall, the identified ODEs retain only the most influential mechanisms for heat and moisture exchange, with ventilation emerging as the primary coupling pathway between temperature and humidity. This result demonstrates the utility of a physics-informed candidate library, joint optimization, and systematic model selection in obtaining parsimonious, robust, and interpretable models. The resulting model is suitable for use in model-based control and further analysis of greenhouse climate systems.

5-4 Model Evaluation through Simulation

With the final SINDy model identified, we evaluate its predictive performance and suitability for greenhouse climate control. Model evaluation is carried out using the simulation data, providing an assessment of both in-sample fit and generalizability to new conditions.

Evaluation Metrics for Control-Oriented Models

To comprehensively assess model quality, we employ four key performance metrics. These will be used to evaluate all simulations in this report:

- **Mean Squared Error (MSE):** MSE quantifies the average of the squared differences between model predictions and observed values. A lower MSE indicates a more accurate model, penalizing larger errors more heavily.
- **Root Mean Squared Error (RMSE):** The Root Mean Squared Error provides the square root of MSE, giving an interpretable, scale-dependent summary of prediction error in the original units of measurement. For control applications, a low RMSE ensures that the model can track reference trajectories closely, minimizing cumulative regulation error.
- **Maximum Absolute Error (MAE):** The Mean Absolute Error measures the average absolute difference between model predictions and observations. MAE is less sensitive to outliers than MSE and RMSE, providing an intuitive measure of typical prediction error.
- **Coefficient of Determination (R^2):** This metric quantifies the proportion of variance in the observed data explained by the model. A high R^2 indicates that the model captures the dominant dynamics of the system, which is essential for robust long-term prediction and feedback control.

By combining these metrics, we not only assess overall fit and accuracy, but also ensure that the model's typical and worst-case performance remain within acceptable bounds—an essential requirement for real-world greenhouse operation and closed-loop control.

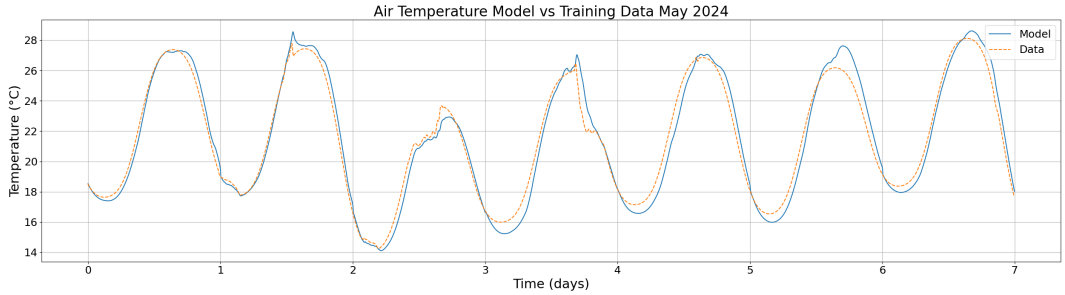
5-4-1 Results on Training and Test Data

The predictive performance of the final SINDy model was evaluated on both the training dataset (2024) and a temporally separated test set (2023), with results summarized in Figures 5-7 and 5-8. The model was re-initialized every 24 hours during simulation, allowing for a realistic assessment of how prediction errors evolve over a typical operational prediction horizon. This approach is consistent with practical MPC deployment and enables identification of any tendencies toward error accumulation or drift.

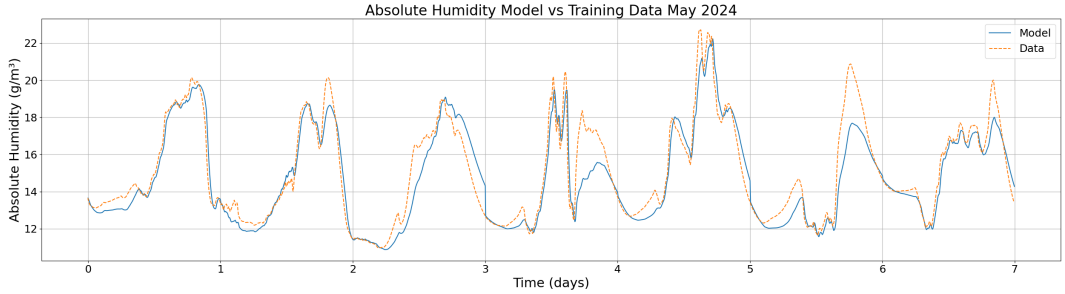
On the training data, the model achieved a high degree of fit for both greenhouse air temperature and absolute humidity. For air temperature, the R^2 score was 0.9601, with a RMSE of 0.7630 °C and a MAE of 2.90 °C (Figure 5-7a). Absolute humidity predictions were also robust, with $R^2 = 0.8679$, RMSE = 0.9583 $\mathrm{g\,m^{-3}}$, and a MAE of 3.90 $\mathrm{g\,m^{-3}}$ (Figure 5-7b). These results confirm that the SINDy model, constructed with a physics-informed library, captures the dominant dynamics of the greenhouse climate on the training set.

On the test set, the SINDy model maintained strong predictive ability, with an R^2 of 0.9421 for air temperature and an RMSE of 0.9101 °C (Figure 5-8a). The MAE remained below 3 °C, demonstrating that the model does not accumulate substantial drift over a typical prediction horizon. For absolute humidity, the model achieved $R^2 = 0.8771$ and RMSE = 0.9778 $\mathrm{g\,m^{-3}}$ (Figure 5-8b), only modestly reduced from the training performance.

Overall, these results show that the identified SINDy model provides a balance between physical interpretability and predictive accuracy. The error metrics remain within the range required for practical, real-time predictive control applications. Furthermore, no significant



(a) MSE = 0.5822, RMSE = 0.7630, MAE = 2.9000, R^2 = 0.9601



(b) MSE = 0.9183, RMSE = 0.9583, MAE = 3.8959, R^2 = 0.8679

Figure 5-7: Performance of the SINDy models on training data for greenhouse air temperature (**top**) and absolute humidity (**bottom**).

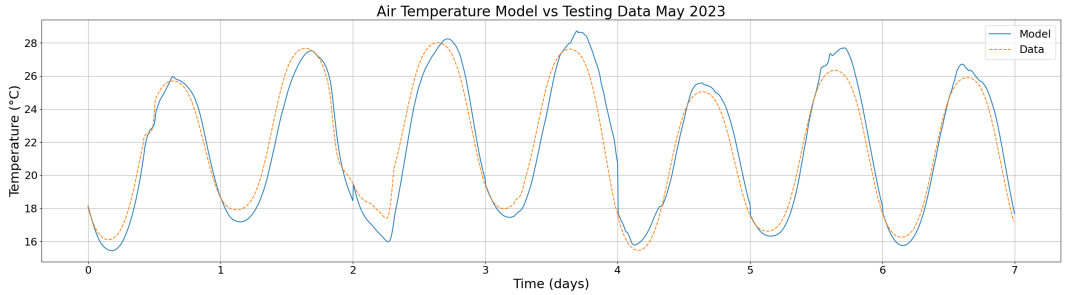
error accumulation or outlier deviations are observed for either air temperature or absolute humidity, supporting the robustness and reliability of the SINDy modeling approach when informed by physical knowledge and systematic model selection.

5-4-2 Generalization to Different Seasons

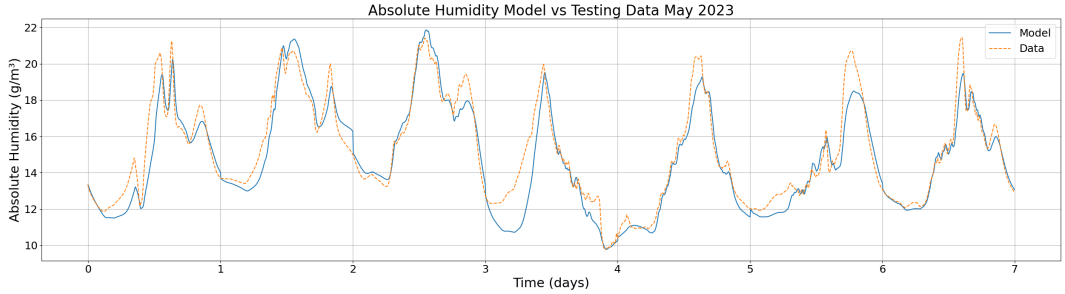
To further assess the robustness and versatility of the SINDy model, its predictive performance was evaluated on test data representing distinct seasonal conditions. Specifically, the months of February (winter), August (summer), and November (autumn). Figure 5-9 presents model predictions for both air temperature and absolute humidity across these months, while Table 5-1 summarizes the corresponding quantitative metrics.

The results indicate that the SINDy model maintains high predictive accuracy for air temperature across different seasons. R^2 values are consistently above 0.94, with RMSE values ranging from 0.66 to 0.91. Even in February, which represents a colder winter period, the model retains a strong fit with $R^2 = 0.96$ and RMSE = 0.79. Maximum absolute errors for air temperature are also within typical control boundaries for greenhouse applications.

Performance for absolute humidity displays more seasonal variability. The model achieves its highest accuracy during August, with $R^2 = 0.94$ and RMSE = 0.67, while lower R^2 values are observed during February ($R^2 = 0.86$) and November ($R^2 = 0.85$). The increase in prediction error during these months may be attributed to more complex humidity dynamics under low external absolute humidity and temperature, or to processes not fully captured



(a) MSE = 0.8283, RMSE = 0.9101, MAE = 2.8388, R^2 = 0.9421



(b) MSE = 0.9561, RMSE = 0.9778, MAE = 3.6459, R^2 = 0.8771

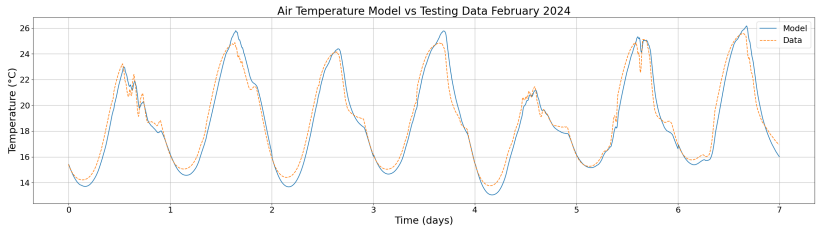
Figure 5-8: Performance of the SINDy models on **testing** data for greenhouse air temperature (**top**) and absolute humidity (**bottom**).

by the SINDy model. However, MAE values for absolute humidity remain below 4 for all months, indicating that the largest individual errors are bounded within a range that is likely acceptable for operational purposes.

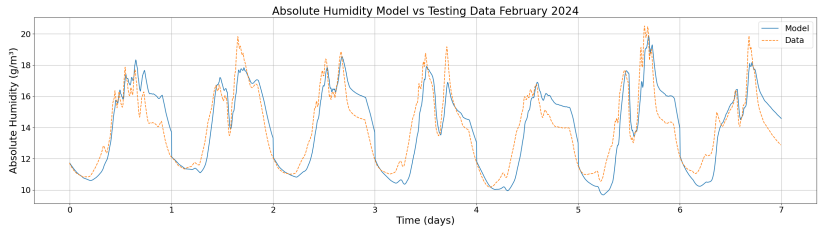
Overall, these results demonstrate that the SINDy model is capable of robust year-round prediction for greenhouse air temperature and provides generally reliable estimates for absolute humidity. Some loss of accuracy is observed for humidity under challenging seasonal regimes, highlighting opportunities for further improvement, such as the inclusion of discrepancy modeling or adaptive retraining protocols. Nonetheless, the strong cross-seasonal performance supports the practical suitability of the identified SINDy model as the foundation for data-driven, interpretable, and robust greenhouse climate control.

Table 5-1: Performance metrics for the SINDy model on test datasets from different months. Metrics are shown for both air temperature and absolute humidity.

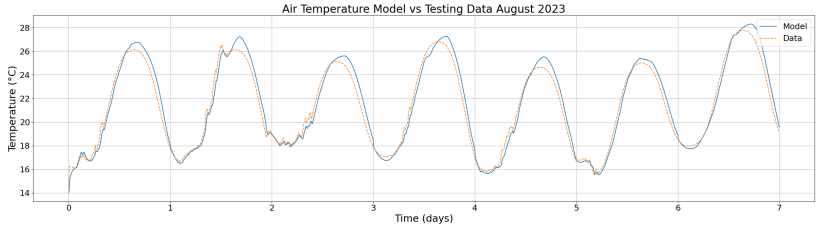
Month	Air Temperature				Absolute Humidity			
	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE	R^2
February	0.5887	0.7673	2.3427	0.9459	1.4827	1.2177	2.8891	0.7309
May	0.8283	0.9101	2.8388	0.9421	0.9561	0.9778	3.6459	0.8771
August	0.5052	0.7108	1.7289	0.9595	0.4464	0.6681	3.2876	0.9432
November	0.4322	0.6574	1.6266	0.9597	0.9004	0.9489	2.6942	0.8542



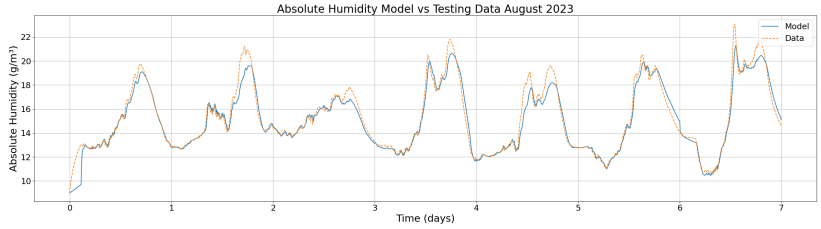
February T_{air}



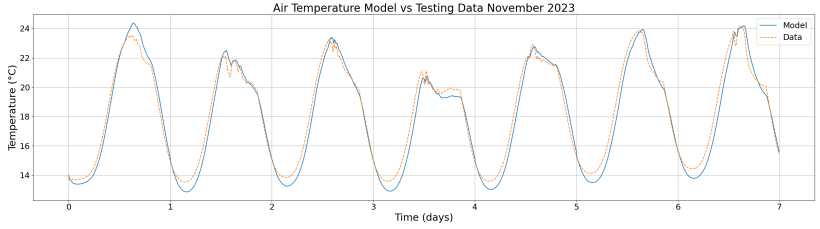
February AH



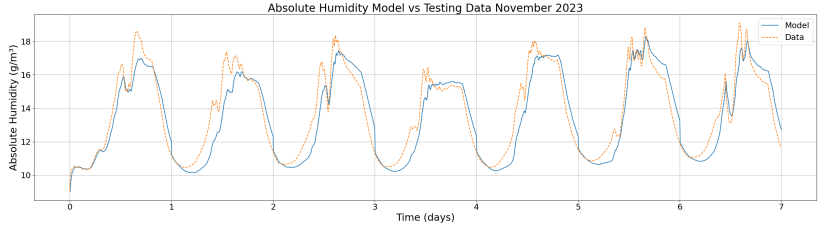
August T_{air}



August AH



November T_{air}



November AH

Figure 5-9: Comparison of model predictions (solid blue) and measurements (dashed orange) for air temperature and absolute humidity in February, August, and November.

Chapter 6

Discrepancy Model

While the physics-informed SINDy model effectively captures the dominant greenhouse climate dynamics, systematic discrepancies between model predictions and observed data remain. These residual errors often stem from complex or unmodeled physical processes, measurement noise, or greenhouse-specific influences not fully represented in the model. To address these limitations, this chapter employs a data-driven discrepancy modeling approach based on a LSTM network. The discrepancy model serves two key functions: it corrects the dynamic residual errors of the physics-based model and enables adaptation to new greenhouses through transfer learning. By leveraging both simulation and real-world data, the combined approach aims to improve prediction accuracy and generalization across diverse greenhouse conditions.

The structure of this chapter is as follows. Section 6-1 introduces the two-stage LSTM-based discrepancy modeling framework, describing both the pre-training on simulation data and the fine-tuning on real greenhouse data. Section 6-2 details the design and training procedure for the LSTM model. Section 6-3 presents validation results on simulated data, including sensitivity analysis, multi-step forecasting, and generalization to real greenhouse datasets. Section 6-5 explores targeted transfer learning for real-world adaptation, and assesses the impact of seasonal and window-based retraining strategies.

6-1 Two-Stage LSTM Training for Discrepancy Modeling

The core of the discrepancy modeling approach is a two-stage LSTM framework designed to supplement the physics-informed SINDy model. Rather than modeling the complete dynamics, the LSTM network is trained to predict the residual errors, defined as the difference between the derivatives predicted by the SINDy model and those observed in the data. The training procedure begins with pre-training on simulated data to capture general dynamic patterns, followed by fine-tuning on a limited set of real greenhouse measurements. This staged process enables the model to generalize to new environments while maintaining adaptability to specific greenhouse conditions.

Pre-training on Simulated Greenhouse Data

In the first stage, the LSTM discrepancy model is trained on the same simulation dataset from 2024 as the SINDy model, with part of the 2023 dataset used for validation. This allows the model to learn the unmodeled dynamics of the system by detecting discrepancies between the physics-based predictions and the true values generated by the simulator. The objective of this stage is to capture the general dynamic residuals that the SINDy model fails to explain.

Fine-tuning with Real Greenhouse Data

In the second stage, the pre-trained LSTM model is further fine-tuned using a small set of historical data from the target greenhouse. This transfer learning step allows the model to specialize to the specific characteristics, processes, and climate control strategies of individual greenhouses, ensuring robust and accurate predictions in practical applications.

6-2 Discrepancy Model Design and Training Procedure

Building on the two-stage training framework, this section details the design and implementation of the LSTM-based discrepancy model, including network architecture selection, training settings, and the transfer learning strategy.

The dataset used for pre-training is the 2024 simulation dataset as used for training the SINDy model. The 2023 simulation dataset is split into a validation set, which is two-thirds of the dataset, and a testing set which is the other one-third. Fine-tuning is subsequently performed on real-world datasets collected from target greenhouses as described in Section 4-2.

The following subsections describe the network architecture and hyperparameter selection, as well as the training settings and procedures adopted for both the pre-training and fine-tuning stages.

6-2-1 LSTM Network Architecture and Hyperparameter Selection

The LSTM network was chosen for the discrepancy model because of its demonstrated ability to capture temporal dependencies in sequential greenhouse data. To determine the most effective architecture, we conducted a grid search over several architectural hyperparameters, including the number and type of layers, the number of units per layer, and the use of dropout for regularization.

The following factors were systematically varied during grid search:

- **Layer composition:** Networks with both dense (1–3 layers) and LSTM (1–3 layers) layers were evaluated. Dense layers capture instantaneous relationships, while LSTM layers are designed for temporal features.
- **Number of layers:** Architectures with 1 to 3 dense layers and 1 to 3 LSTM layers were considered to capture complex dynamics.

- **Units per layer:** Each layer was assigned 16, 32, 64, 128, or 256 units to balance underfitting and overfitting.
- **Dropout rates:** Dropout rates of 0.1 and 0.2 were applied to reduce overfitting.

The best-performing LSTM network has the following structure:

- **First LSTM layer:** 64 units, 0.2 dropout, returns sequences.
- **Second LSTM layer:** 128 units, 0.1 dropout, does not return sequences.
- **Dense layer:** 32 nodes, ReLU activation.
- **Output layer:** 2 nodes (predicting Tair and AH).

6-2-2 Training Settings and Transfer Learning Strategy

Training was performed using a batch size of 32 and a maximum of 50 epochs. The Adam optimizer was used, and a MSE loss function. Early stopping was implemented with a patience of 5 epochs. Learning rate scheduling was handled using ReduceLROnPlateau when the validation loss plateaued.

After initial training on simulated data, the pre-trained LSTM model was adapted to each real greenhouse using transfer learning. The model's learned weights served as initialization; fine-tuning was performed with a reduced learning rate of 1×10^{-4} and a reduced epoch count of 30. Model checkpointing was used to retain the best validation performance.

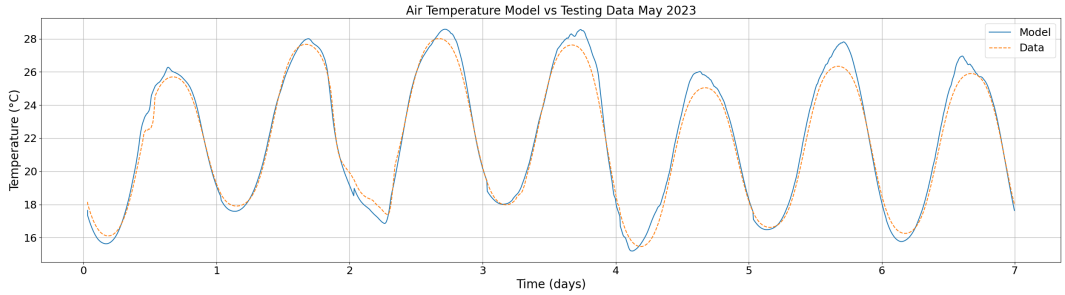
The transfer learning process consisted of loading the pre-trained simulation model, fine-tuning on real greenhouse data with the reduced learning rate, and saving the model checkpoint with the lowest validation loss.

6-3 Validation on Simulated Data

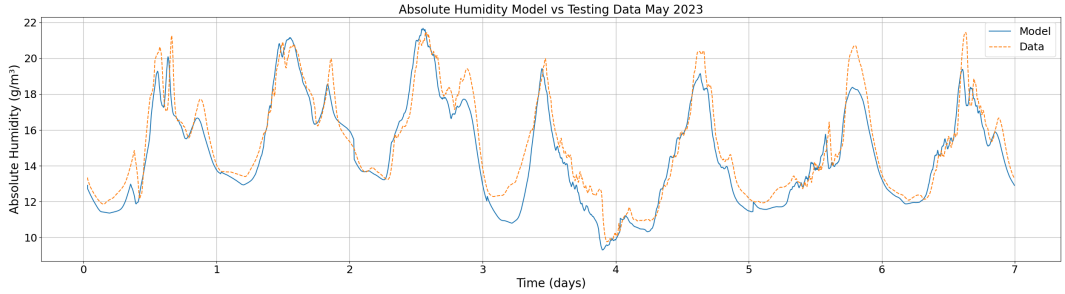
After training the discrepancy model, the combined SINDy and discrepancy model is tested on the simulation testing dataset. A sensitivity analysis follows to assess the impact of different parameters on the model's predictions, helping to identify key factors influencing its behavior. The model's response to varying simulation horizons is also analyzed, providing insights into its performance over short-term and long-term periods. Finally, the model is tested on real-world greenhouse data to evaluate its generalization capabilities. This step helps assess how well the model can predict greenhouse conditions without being trained specifically on each greenhouse's data.

6-3-1 Model Performance on Simulated Test Sets

The discrepancy model's accuracy is evaluated by running simulations on the testing dataset and analysing the performance metrics.



(a) Tair: MSE=0.2582, RMSE=0.5081, MAE=1.8380, R^2=0.9819



(b) AH: MSE=0.8147, RMSE=0.9026, MAE=3.8605, R^2=0.8954

Figure 6-1: Tair and AH predictions on the testing dataset from May 2023.

The metrics, MSE, RMSE, and MAE, show significant reductions for the simulation of Tair. Specifically, the MSE decreased from 0.8283 to 0.3062, the RMSE from 0.9101 to 0.5533, and the MAE from 2.8388 to 1.7516, with the R^2 increasing from 0.9421 to 0.9786. These results indicate that the discrepancy model effectively enhances the predictive accuracy for Tair, capturing system dynamics that the SINDy model failed to model sufficiently.

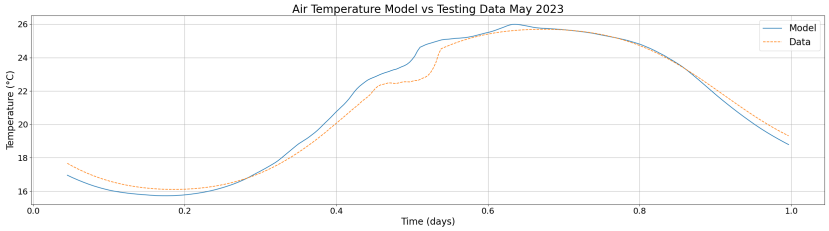
For AH, although the MAE slightly increased from 3.6459 to 3.8605, the MSE and RMSE improved (MSE from 0.9561 to 0.8147, RMSE from 0.9778 to 0.9026), and the R^2 score improved from 0.8771 to 0.8954, indicating a better fit to the data. This suggests that while there was a slight increase in the MAE, the model still achieved better overall predictive performance, especially in terms of the explained variance (R^2).

6-3-2 Sensitivity Analysis

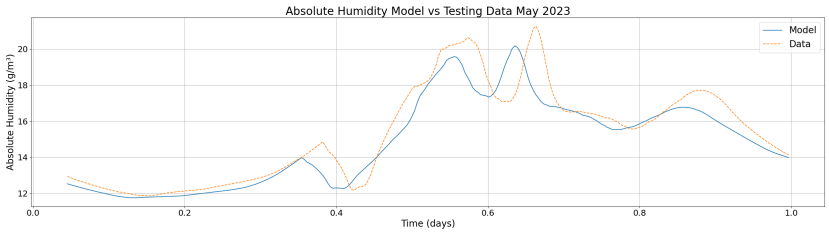
A sensitivity analysis was performed to evaluate the model's response to changes in key environmental and system parameters. The analysis consisted of a series of scenarios designed to test model behavior under conditions relevant for greenhouse climate control. The following paragraphs describe the observed outcomes for each scenario.

Decrease in Radiation (Cloud Cover)

In this scenario, the radiation is reduced between 12:00 and 13:00, simulating cloud cover. A reduction in air temperature is observed, along with a corresponding decrease in absolute humidity. These results are consistent with the expected effects of reduced solar heating



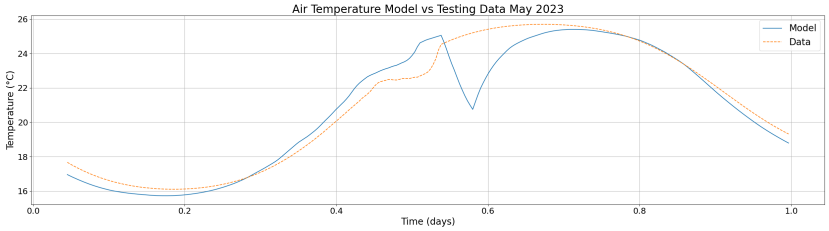
(a) One day simulation Tair



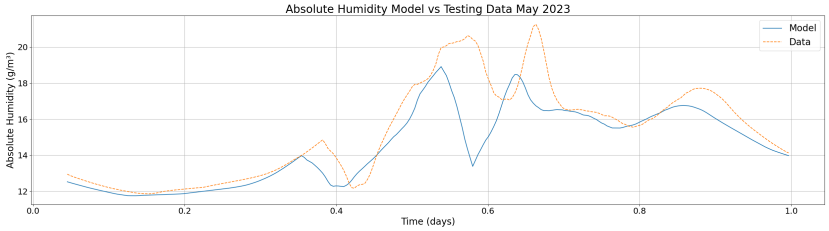
(b) One day simulation AH

Figure 6-2: One day simulation showing the base case for the sensitivity analysis.

and lower transpiration. After the radiation returns to normal, the model output for both air temperature and absolute humidity recovers and returns to the original trajectory. The results are illustrated in Figure 6-3.



(a) Sensitivity to Radiation: Tair

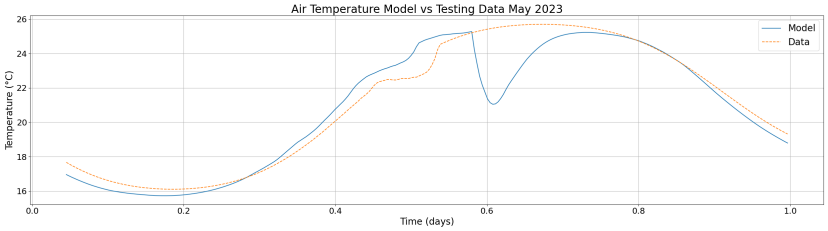


(b) Sensitivity to Radiation: AH

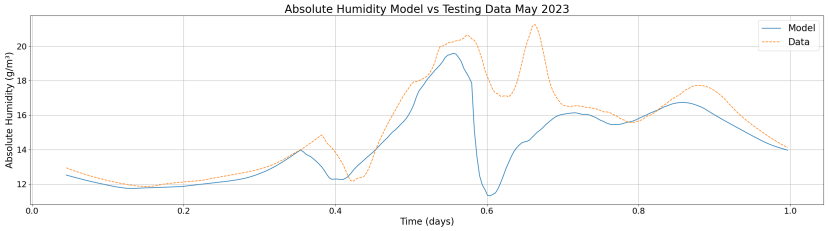
Figure 6-3: Sensitivity to Radiation: Effect on Air Temperature and Absolute Humidity.

Increase in Windspeed (Wind Blows)

In this the wind speed is increased from 2 m/s to 8 m/s between 14:00 and 15:00. Enhanced ventilation leads to reductions in both air temperature and absolute humidity. The model output displays these cooling and drying effects, reflecting the increased air exchange with the outside environment. The changes are visible in Figure 6-4.



(a) Sensitivity to Windspeed: Tair

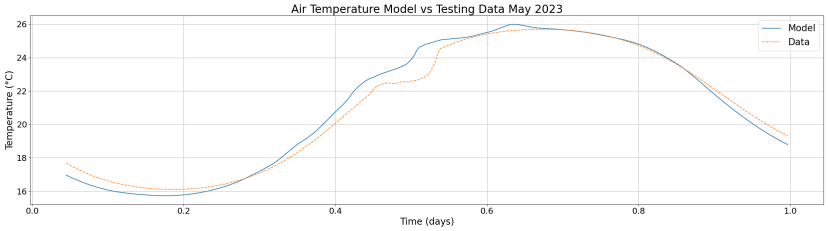


(b) Sensitivity to Windspeed: AH

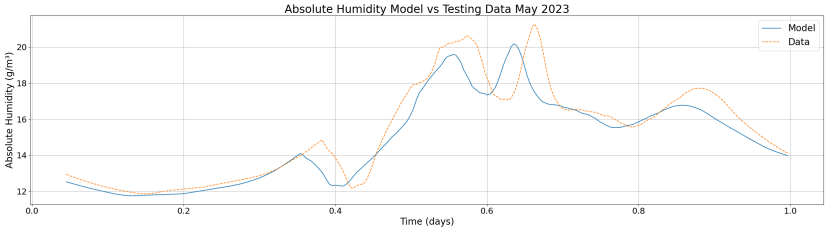
Figure 6-4: Sensitivity to Windspeed: Effect on Air Temperature and Absolute Humidity.

Increased Outside Absolute Humidity (Fog)

In this scenario outside absolute humidity is increased to 9.5 g/m^3 between 05:00 and 07:00 to simulate foggy conditions. only minor changes are observed in the model output for both air temperature and absolute humidity. The performance metrics for this scenario remain close to those of the base case, indicating that the model does not show a substantial response to this disturbance under the tested conditions. These results are presented in Figure 6-5.



(a) Sensitivity to Fog: Tair

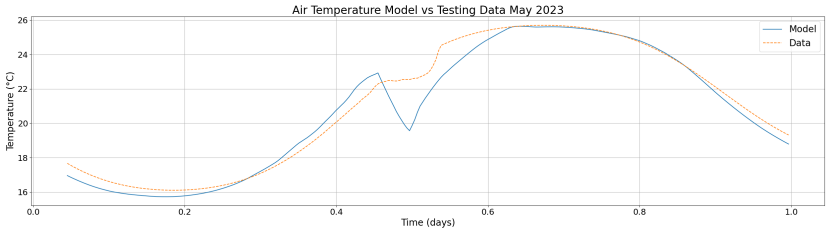


(b) Sensitivity to Fog: AH

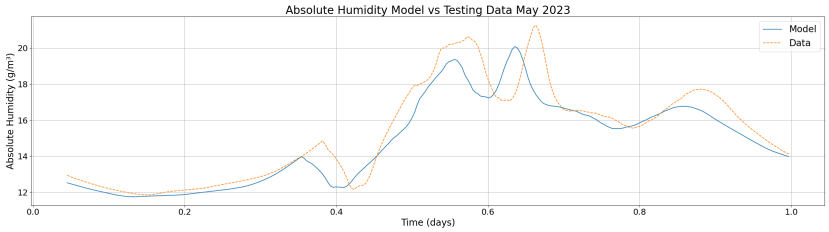
Figure 6-5: Sensitivity to Fog: Effect on Air Temperature (Tair) and Absolute Humidity (AH).

Decreased Tpipe (Heating Failure)

In this scenario, simulating a heating failure, the pipe temperature is reduced to ambient air temperature between 11:00 and 12:00. The model output indicates a temporary decrease in air temperature, while absolute humidity remains largely unaffected. The model recovers after the disturbance, and the main change in the performance metrics is an increase in mean absolute error for air temperature during the transient. The results are shown in Figure 6-6.



(a) Sensitivity to Heating: Tair



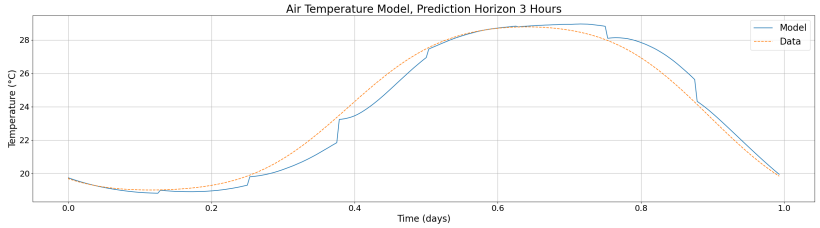
(b) Sensitivity to Heating: AH

Figure 6-6: Sensitivity to Heating: Effect on Air Temperature (Tair) and Absolute Humidity (AH).

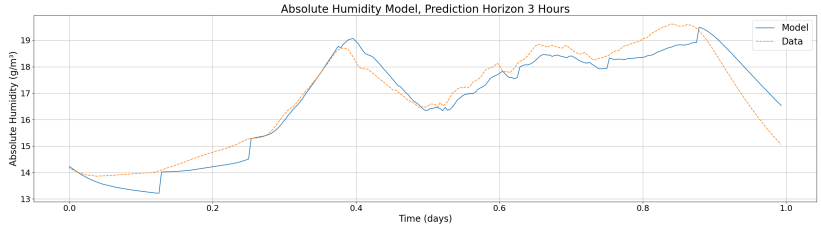
Overall, the results indicate that the model responds appropriately to a range of environmental and system disturbances, with outputs that return to the desired trajectory after each scenario. This suggests that the model is able to recover from transient perturbations and maintain consistent behavior under varying conditions.

6-3-3 Multi-Step Forecasting Across Prediction Horizons

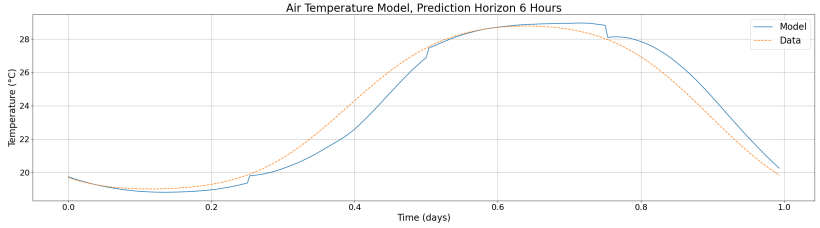
To assess the model's predictive performance over varying forecast intervals, a series of simulations were conducted in which the model was re-initialized every x hours, with x set to 3, 6, 12, or 24 hours. In each scenario, the model makes continuous predictions over the specified horizon without re-initialization, thus providing a test of its robustness for multi-step ahead forecasting, like the prediction horizon in MPC, where accurate forecasts are required over the controller's prediction window.



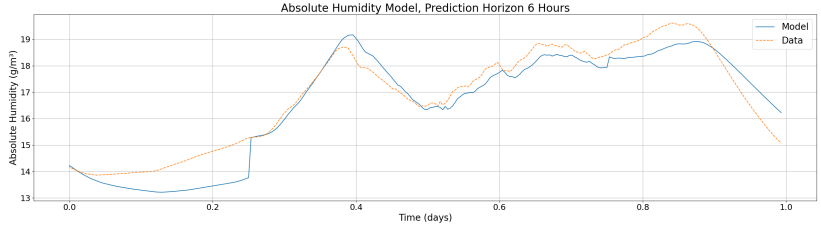
3 h horizon T_{air}



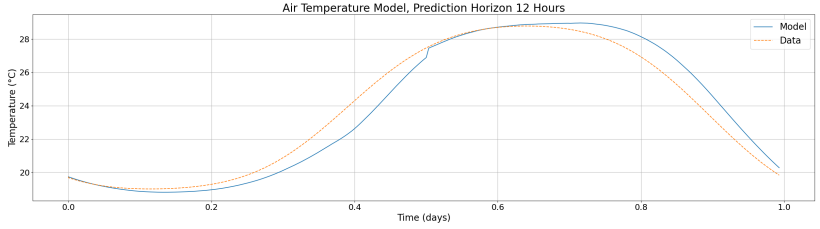
3 h horizon AH



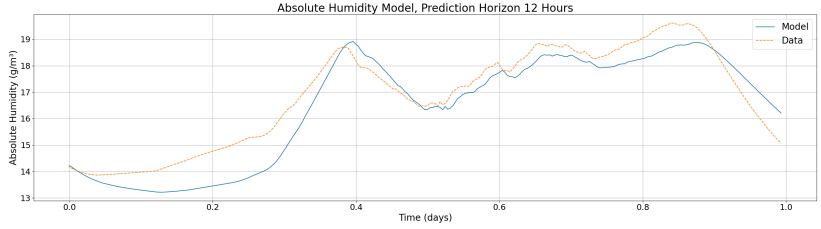
6 h horizon T_{air}



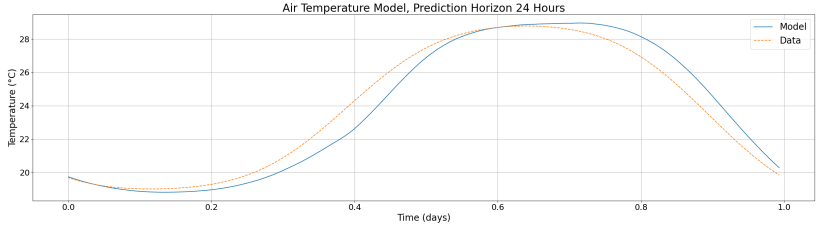
6 h horizon AH



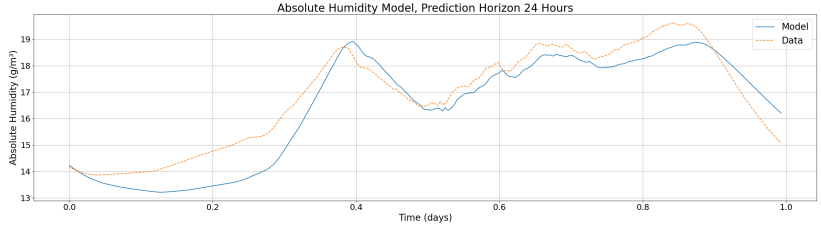
12 h horizon T_{air}



12 h horizon AH



24 h horizon T_{air}



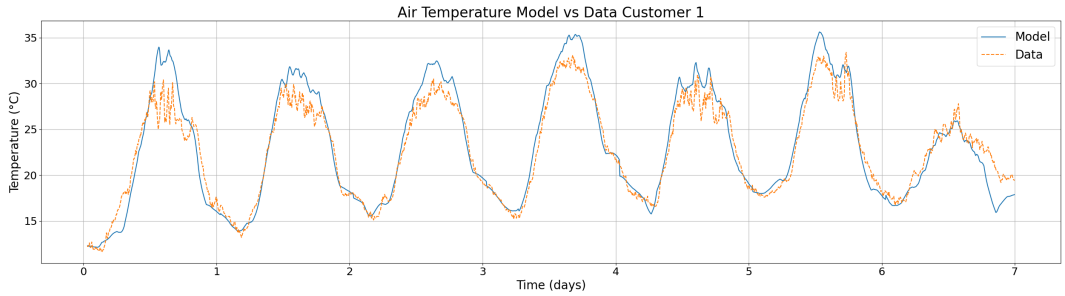
24 h horizon AH

Figure 6-7: Prediction accuracy for different re-initialisation horizons. Left column: air-temperature forecasts; right column: absolute-humidity forecasts. Each subplot reports the associated error metrics (MSE, RMSE, MAE, R^2).

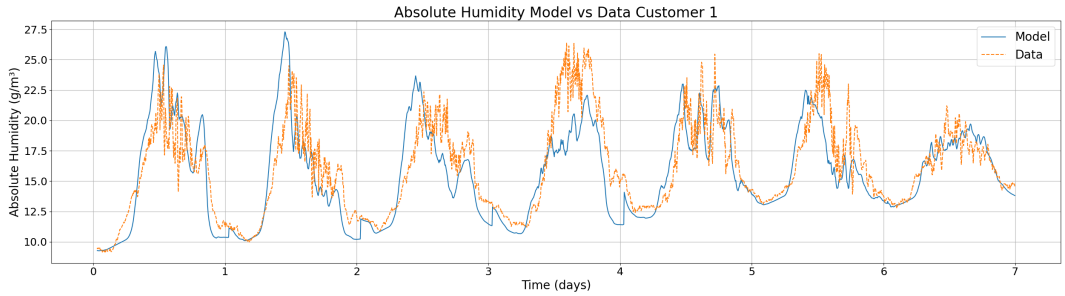
As summarized in Figure 6-7, the model demonstrates strong predictive accuracy for both Tair and AH across all tested horizons. The R^2 values remain high (> 0.94 for Tair and > 0.82 for AH) even for the longest, 24-hour prediction window, indicating that the model is able to provide reliable forecasts without frequent re-initialization. As expected, a gradual decline in accuracy is observed with increasing prediction horizon, as shown by higher RMSE and MSE values. However, the overall level of error remains low, and the maximum absolute errors remain below 2 units for all cases. These results confirm the model's robustness for use in practical MPC applications, where accurate multi-step predictions are required.

6-4 Generalization to Real Greenhouse Data

To assess the model's ability to generalize to new, unseen, and realistic data, simulations were conducted using real-world datasets from three different greenhouses. One dataset is provided by TomatoWorld, while the other two datasets are from anonymous customers of Hoogendoorn, all located in the Netherlands and focused on tomato production. All simulations were performed using data from May 2024.



(a) Tair: MSE=4.0133, RMSE=2.0033, MAE=8.2937, R^2=0.8524

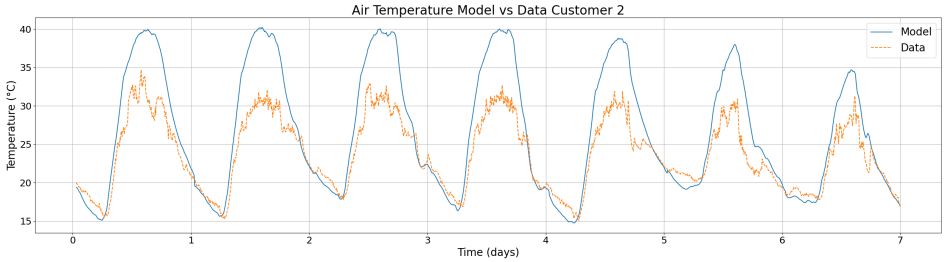


(b) AH: MSE=6.6363, RMSE=2.5761, MAE=9.1576, R^2=0.5189

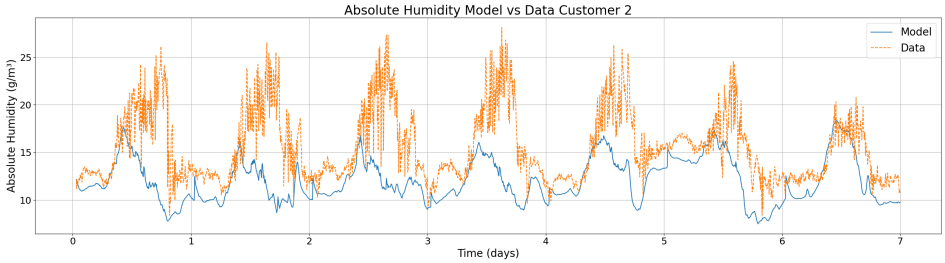
Figure 6-8: Tair and AH simulations on real-world data from Customer 1.

The results for Customer 1 (Figure 6-8) indicate that the model captures the temperature dynamics with reasonable accuracy, as reflected by an R^2 of 0.85 for Tair. The corresponding RMSE is 2.00, and the MAE is 8.29, suggesting moderate overall error but strong correlation between predicted and observed values. For AH, the performance is weaker, with an R^2 of 0.52 and RMSE of 2.58, indicating that while some variance is explained, significant residual errors remain.

For Customer 2 (Figure 6-9), the predictive accuracy is substantially lower. The R^2 for Tair



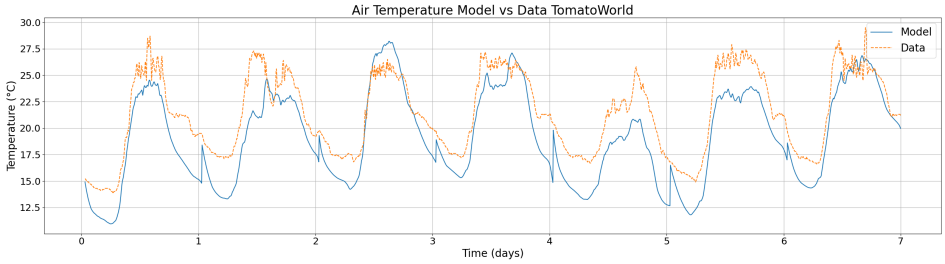
(a) Tair: MSE=21.6585, RMSE=4.6539, MAE=10.9405, R^2=0.0144



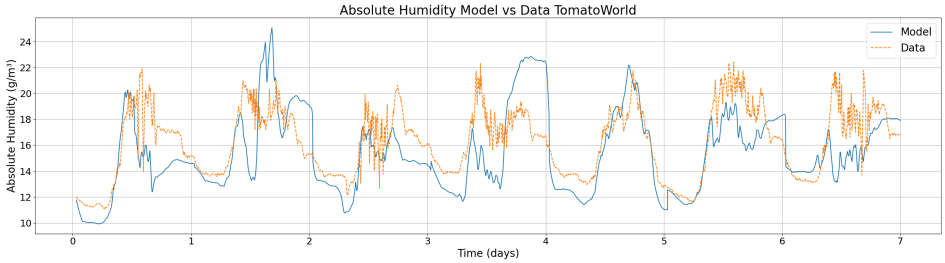
(b) AH: MSE=24.7750, RMSE=4.9774, MAE=16.6440, R^2=-0.7927

Figure 6-9: Tair and AH simulations on real-world data from Customer 2.

drops to 0.01, with RMSE rising to 4.65 and MAE to 10.94, indicating large deviations between predicted and observed temperatures. For AH, the model yields a negative R^2 (-0.79) and a high RMSE of 4.98, pointing to poor model fit and substantial prediction error. The model consistently overpredicts Tair and AH for this dataset, indicating a lack of generalization to these particular greenhouse conditions.



(a) Tair: MSE=11.8595, RMSE=3.4438, MAE=7.5486, R^2=0.1811



(b) AH: MSE=6.5393, RMSE=2.5572, MAE=10.1507, R^2=0.0618

Figure 6-10: Tair and AH simulations on real-world data from TomatoWorld.

For TomatoWorld (Figure 6-10), the R^2 for Tair is 0.18, with an RMSE of 3.44 and MAE

of 7.55. This reflects a limited ability to track the observed temperature trajectory, with predictions that tend to underpredict the measured values. For AH, the R^2 is close to zero (0.06), with RMSE of 2.56 and MAE of 10.15, indicating that the model explains very little of the variance in the observed humidity data. The figure also show that the AH dynamics are highly irregular making it difficult to model.

Overall, the results show that the model generalizes well to the data from Customer 1. For Customer 2 and TomatoWorld, both Tair and AH predictions are less accurate, with notably lower R^2 values and higher error metrics. The performance drop, particularly for absolute humidity, suggests that greenhouse-specific factors and unmodeled crop effects limit the transferability of the model when trained solely on simulation data. These findings highlight the importance of adaptation or fine-tuning to new operational environments, especially for variables strongly influenced by local and biological conditions.

Given these limitations in generalization, it is necessary to adapt the model to better account for the unique characteristics of individual greenhouses.

6-5 Adapting the Model via Transfer Learning

In this research, the predictive model was initially trained on simulated greenhouse climate data, providing a strong physics-based foundation. To enhance its accuracy and adaptability in real-world applications, the model is subsequently fine-tuned using datasets collected from operational greenhouses, from two customers of Hoogendoorn and from TomatoWorld. This transfer learning approach allows the model to leverage general knowledge acquired from simulation, while efficiently adapting its parameters to capture the specific dynamics, crop conditions, and environmental characteristics of each real greenhouse setting.

6-5-1 Model Performance on Real Data

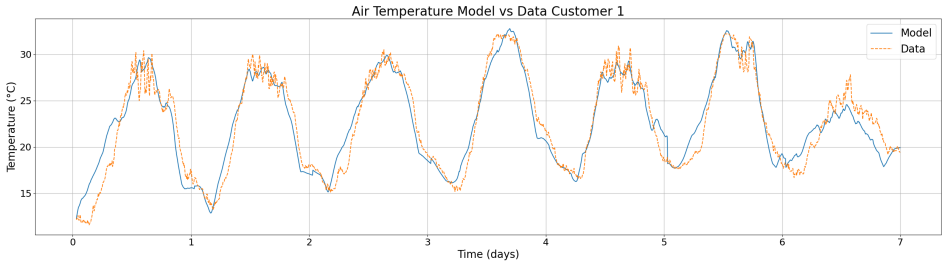
In this section, we evaluate the performance of the fine-tuned LSTM model using real-world greenhouse data. The pre-trained LSTM model, initially trained on simulation data, is fine-tuned using 100 days of data from each greenhouse. Simulations are then conducted on the same real-world datasets as in the previous section. Performance metrics for both air temperature and absolute humidity are summarized in Table 6-1.

For Customer 1, fine-tuning leads to improved model performance for both Tair and AH, as demonstrated by increased R^2 values and lower error metrics in Table 6-1. The most pronounced improvements are observed in Tair prediction, showing that the model can adapt to the conditions of Customer 1 with additional real-world data. While AH prediction also improves after fine-tuning, the error metrics remain higher than for Tair, which may be attributable to the complexity of humidity dynamics or factors such as crop management that are not fully captured in the data.

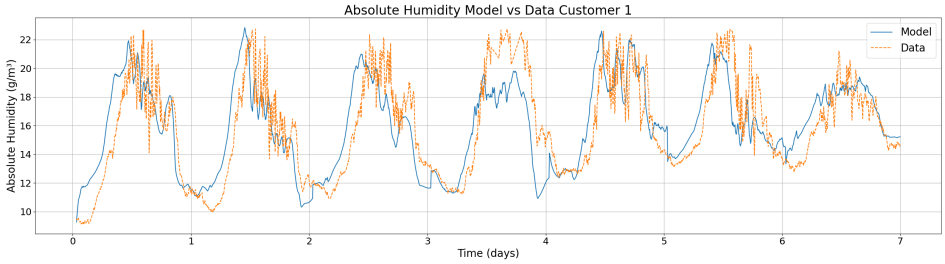
For Customer 2, fine-tuning improves Tair prediction, reflected in a notable increase in R^2 and reduction in error metrics. However, the performance for AH remains poor, with low R^2 and high error metrics both before and after fine-tuning. This indicates that while the model can adapt to temperature in Customer 2's greenhouse with transfer learning, it does not generalize as well to humidity.

Table 6-1: Performance metrics (MSE, RMSE, MAE, R^2) for Tair and AH before and after fine-tuning for each greenhouse.

		MSE	RMSE	MAE	R^2
Customer 1	Tair (pre)	4.0133	2.0033	8.2937	0.8524
	Tair (fine)	3.1080	1.7630	7.0479	0.8857
	AH (pre)	6.6363	2.5761	9.1576	0.5189
	AH (fine)	5.1393	2.2670	8.3090	0.6274
Customer 2	Tair (pre)	21.6585	4.6532	10.9405	0.0144
	Tair (fine)	3.9719	1.9930	5.8483	0.8193
	AH (pre)	24.7750	4.9774	16.6440	-0.7927
	AH (fine)	8.9489	2.9915	12.1757	0.3525
TomatoWorld	Tair (pre)	11.8595	3.4438	7.5486	0.1811
	Tair (fine)	2.3826	1.5436	7.1087	0.8355
	AH (pre)	6.5393	2.5572	10.1507	0.0618
	AH (fine)	4.1741	2.0431	7.3628	0.3966



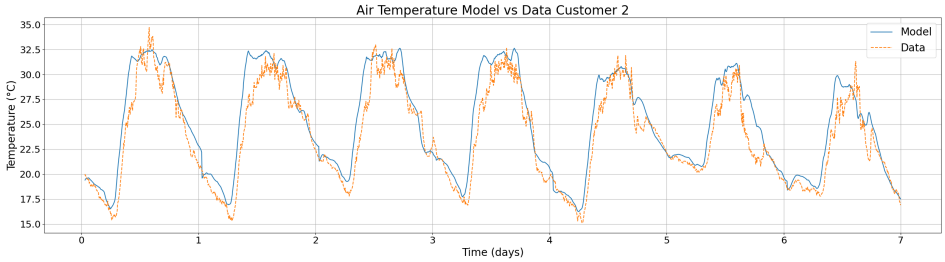
(a) Tair predictions for Customer 1 after fine-tuning.



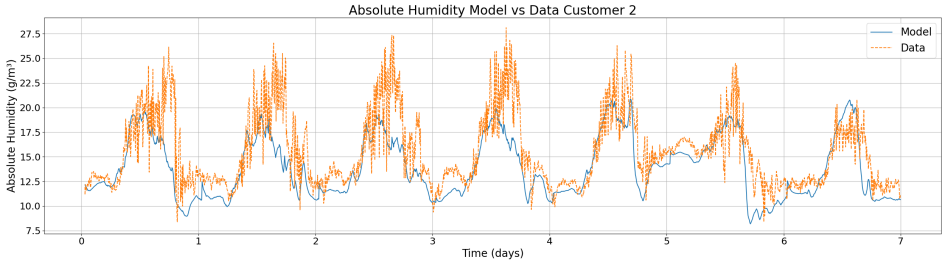
(b) AH predictions for Customer 1 after fine-tuning.

Figure 6-11: Simulations for Customer 1: Tair and AH predictions after fine-tuning.

For TomatoWorld, fine-tuning improves Tair prediction, with higher R^2 and reduced errors as reported in Table 6-1. The performance for AH also shows some improvement, but overall error metrics remain relatively high. These results indicate that transfer learning effectively adapts the model for temperature prediction in TomatoWorld, but further work is required to achieve similar improvements in humidity prediction.

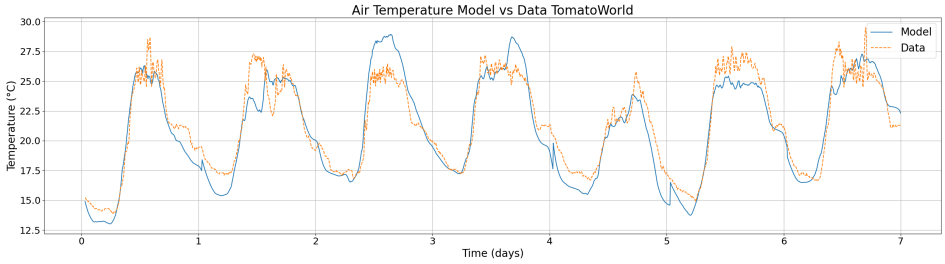


(a) Tair predictions for Customer 2 after fine-tuning.

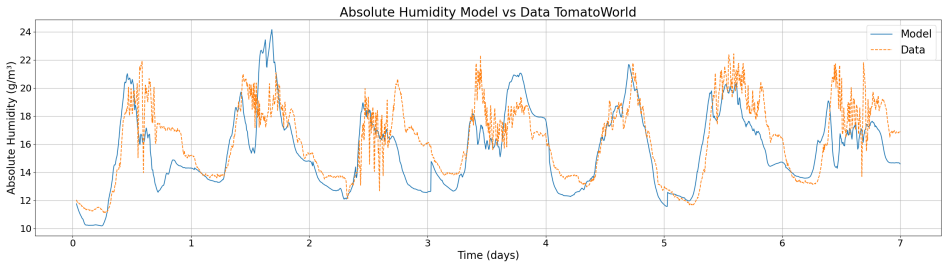


(b) AH predictions for Customer 2 after fine-tuning.

Figure 6-12: Simulations for Customer 2: Tair and AH predictions after fine-tuning.



(a) Tair predictions for TomatoWorld after fine-tuning.



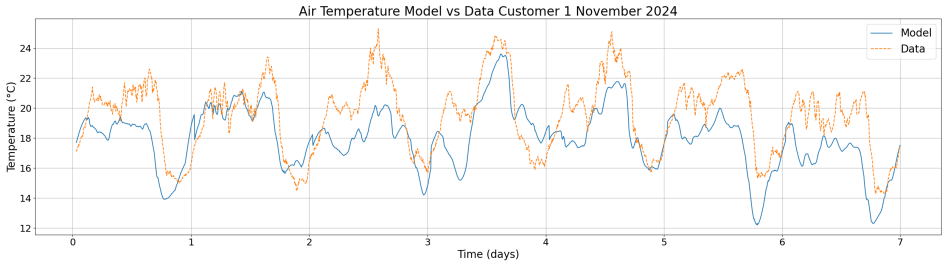
(b) AH predictions for TomatoWorld after fine-tuning.

Figure 6-13: Simulations for TomatoWorld: Tair and AH predictions after fine-tuning.

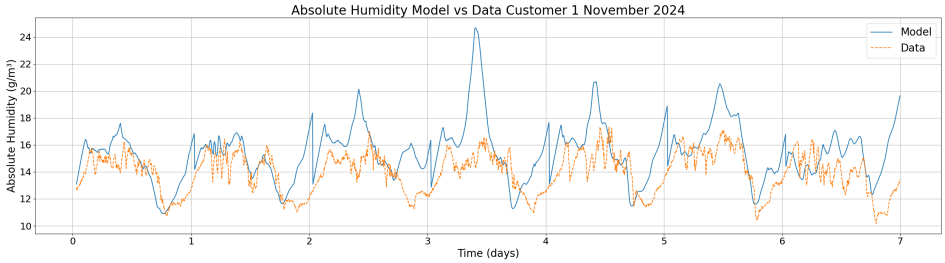
Overall, these results demonstrate that transfer learning via fine-tuning the LSTM model on real-world greenhouse data leads to clear improvements in air temperature prediction across all cases. However, the prediction of absolute humidity remains less consistent, indicating the need for further model refinement or additional data to better capture the underlying greenhouse humidity dynamics. Having established the benefits of fine-tuning on real-world datasets, the next step is to assess how well the fine-tuned model generalizes to new environmental conditions not present in the training data.

6-5-2 Seasonal Generalization: Testing Across Multiple Months

In this section, the generalization capability of the fine-tuned model is evaluated by testing its performance across different periods of the year using data from Customer 1. This dataset was selected due to its high predictive accuracy in previous simulations, making it a suitable case for assessing seasonal robustness. The model is tested on simulation data corresponding to February, August, and November, representing winter, summer, and late autumn conditions, respectively. The training data for the fine-tuned model covered only the spring and early summer period (April, May, and June), enabling an assessment of predictive accuracy under both familiar and unfamiliar environmental conditions.



(a) Tair predictions for November. MSE=7.5810, RMSE=2.7534, MAE=8.1562, R^2=-0.3098



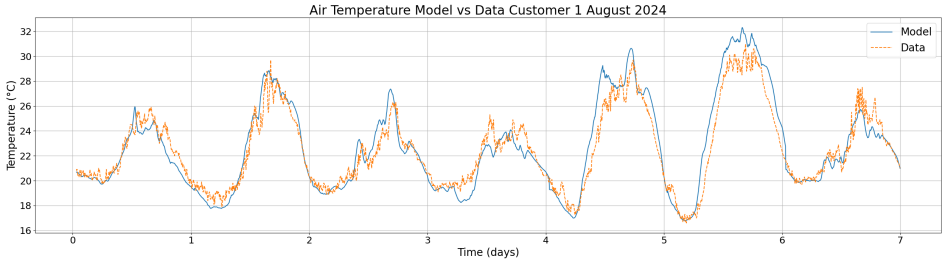
(b) AH predictions for November. MSE=4.8452, RMSE=2.2012, MAE=9.0900, R^2=-1.1548

Figure 6-14: Tair and AH predictions after fine-tuning for November.

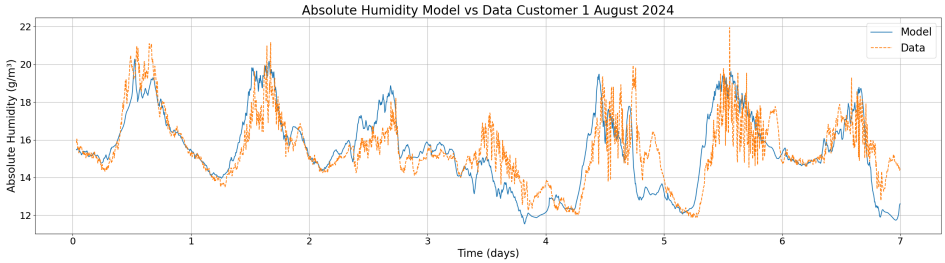
For November, the model exhibits low predictive accuracy for both air temperature and absolute humidity. The R^2 values are negative (R^2 = -0.31 for Tair and R^2 = -1.15 for AH), indicating that the model fails to capture the structure of the observed data for both variables. The RMSE values for Tair and AH are 2.75 and 2.20, respectively, and MAE values are 8.16 and 7.09. These high error metrics suggest a considerable mismatch between predictions and observations.

In August, the model's predictive accuracy improves for both Tair and AH. The R^2 value for Tair increases to 0.71, with RMSE and MAE dropping to 1.74 and 5.56, respectively. For AH, the R^2 value is 0.54, RMSE is 1.30, and MAE is 6.45. These results indicate that the model is able to explain a substantial portion of the observed variance for both variables. August is characterized by environmental conditions that are more similar to the model's training period, which may explain the improved performance relative to the other months.

In February, predictive performance declines for both Tair and AH, with R^2 values of -0.24 and -0.83, respectively. The RMSE for Tair is 2.86, and for AH it is 2.53. The MAE values are 6.64 for Tair and 7.13 for AH. These results demonstrate that the model is not able to reliably capture the climate dynamics during winter conditions, which are not represented in the

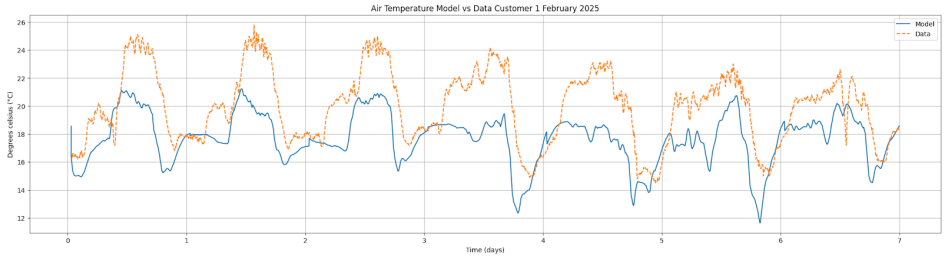


(a) Tair predictions for August. MSE=1.3729, RMSE=1.1717, MAE=4.3208, R^2=0.8578

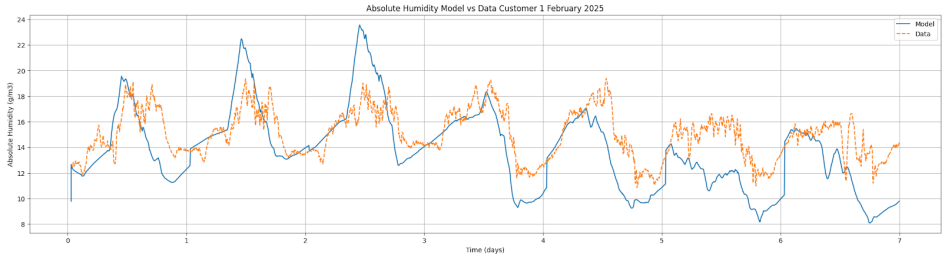


(b) AH predictions for August. MSE=1.6651, RMSE=1.2865, MAE=6.1344, R^2=0.4596

Figure 6-15: Tair and AH predictions after fine-tuning for August.



(a) Tair predictions for February. MSE=8.1785, RMSE=2.8598, MAE=6.6376, R^2=-0.2355



(b) AH predictions for February. MSE=6.3938, RMSE=2.5286, MAE=7.1321, R^2=-0.8334

Figure 6-16: Tair and AH predictions after fine-tuning for February.

training data. The combination of negative R^2 values and higher error metrics in February and November compared to August provides evidence that the model's generalization is limited when external conditions differ substantially from those seen during training.

Overall, these results demonstrate that the fine-tuned model maintains higher predictive accuracy in August, a period similar to the original training window, but struggles during the winter and late autumn months. The observed performance drop outside the training season is reflected in both lower R^2 values and increased error metrics. This suggests that

the model's predictive skill is strongly influenced by the similarity between the current and training environmental conditions.

To address this limitation, the next section investigates whether targeted transfer learning, by re-training the model with data from February and November, can improve predictive performance during these months. By supplementing the original training data with additional season-specific samples, it is possible to assess whether the model can better adapt to differing winter and late autumn conditions.

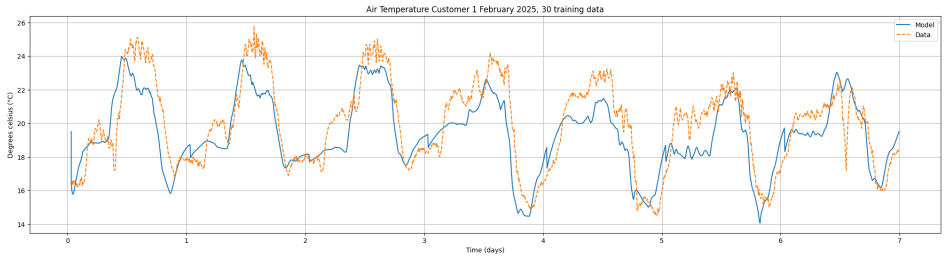
6-5-3 Seasonal Adaptation: Month-Specific and Moving Window Retraining

To systematically evaluate the effect of season-specific retraining, two approaches are considered. First, a month-specific strategy is implemented, in which the model is retrained using all available data from the target month excluding the simulation week. This setup establishes an upper benchmark for predictive performance by training on data that closely match the conditions of the test period.

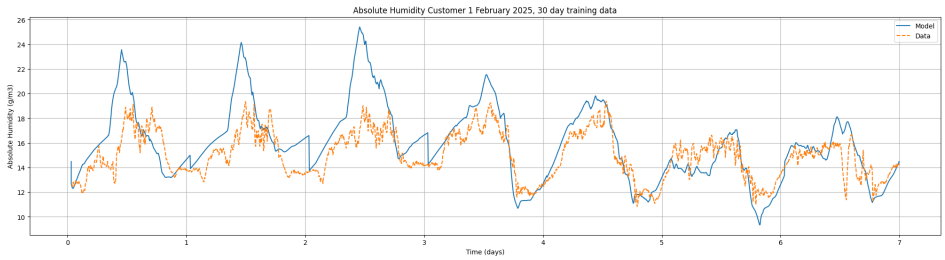
Table 6-2: Performance metrics Tair and AH under different retraining windows in February and November. "Initial" is the non re-trained model; "Month" uses all data from the month except the test week; "30d", "60d", and "90d" denote rolling-window lengths.

Test Period	Window	Tair / AH			
		MSE	RMSE	MAE	R^2
February	Initial	8.1785 / 6.3938	2.8598 / 2.5286	6.64 / 7.13	-0.2355 / -0.8334
	Month	2.2500 / 4.6522	1.5000 / 2.1569	5.35 / 8.99	0.6601 / -0.3340
	30d	3.1279 / 11.1704	1.7686 / 3.3422	5.73 / 12.45	0.5275 / -2.2031
	60d	5.1798 / 17.2778	2.2759 / 4.1567	6.60 / 12.05	0.2175 / -3.9543
	90d	4.3183 / 11.6310	2.0781 / 3.4104	6.09 / 10.06	0.3477 / -2.3351
November	Initial	7.5810 / 4.8452	2.7534 / 2.2012	8.16 / 7.09	-0.3098 / -1.1548
	Month	3.5045 / 5.5595	1.8720 / 2.3579	6.88 / 8.71	0.3945 / -1.4725
	30d	4.9208 / 6.8696	2.2183 / 2.6210	6.62 / 10.31	0.1498 / -2.0551
	60d	7.4079 / 5.8271	2.7217 / 2.4139	6.67 / 9.67	-0.2799 / -1.5915
	90d	9.3675 / 7.1054	3.0606 / 2.6656	7.87 / 9.69	-0.6185 / -2.1600

After establishing this benchmark, a moving window approach is explored, where the model is fine-tuned using only the most recent 30, 60, or 90 days of data prior to the test period. This strategy provides a more practical, real-time retraining method that could be applied in operational settings. By comparing the two approaches, the analysis assesses the trade-off between maximum achievable accuracy and the feasibility of continual adaptation in real-world applications.

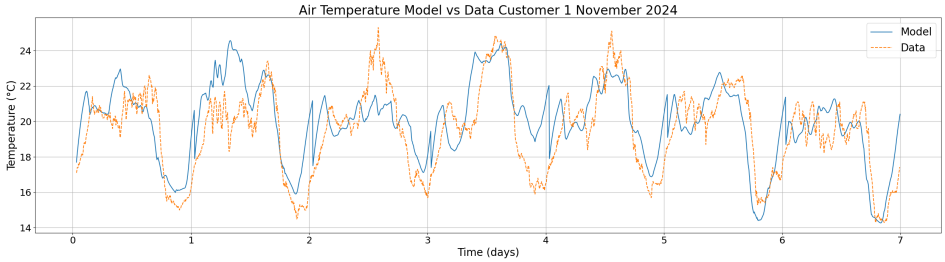


(a) Tair predictions for February after re-training on all February data except the simulated week.

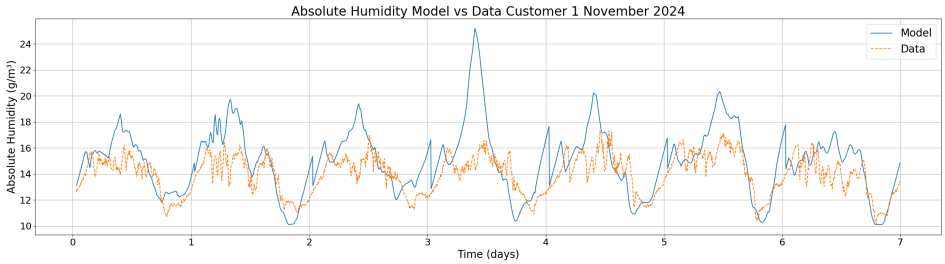


(b) AH predictions for February after re-training on all February data except the simulated week.

Figure 6-17: Tair and AH predictions after re-training with all available February data except the simulation week.



(a) Tair predictions for November after re-training on all November data except the simulated week.



(b) AH predictions for November after re-training on all November data except the simulated week.

Figure 6-18: Tair and AH predictions after re-training with all available November data except the simulation week.

Figure 6-17 shows the results for February, where the month-specific retraining yields the best performance for Tair compared to all other strategies. As shown in Table 6-2, the Tair MSE drops to 2.25 and R^2 rises to 0.66, a substantial improvement over both the initial model ($R^2 = -0.24$) and the moving window approaches. For the moving window strategies, the 30-day window provides moderate improvement ($R^2 = 0.53$), while 60-day and 90-day

windows show decreasing accuracy. The results suggest that the model benefits most from training on recent, month-specific data. However, the predictive performance for AH remains limited: even with month-specific retraining, R^2 is -0.33 and errors are higher than for Tair.

A similar pattern is observed in November, as shown in Figure 6-18 and Table 6-2. With month-specific retraining, Tair prediction improves, achieving an MSE of 3.50 and an R^2 of 0.39, compared to -0.31 for the initial model. MAE is also reduced to 6.88. For the moving window approaches, the 30-day window yields a Tair R^2 of 0.15, with further declines for the 60-day and 90-day windows, again highlighting the importance of recent, relevant training data. For AH, the highest R^2 is -1.47, with even lower values for the moving windows, indicating that retraining strategies are less effective for capturing humidity dynamics, possibly due to complex or unobserved drivers.

Overall, these results demonstrate that retraining the model with data closely matched to the target period yields the highest predictive accuracy for Tair. The moving 30-day window also provides clear improvements for Tair compared to longer windows and the initial fine-tuned model. The analysis highlights that the relevance and similarity of the training data to the test period are more important for model performance than the total amount of data used. However, performance for AH remains lower across all strategies, emphasizing the challenges in predicting humidity due to factors likely not captured in the available features. Reducing the window below 30 days was also found to return decreasingly accurate models, indicating a minimum threshold of recent data is necessary for stable retraining. These findings support the feasibility of real-time adaptive retraining strategies for temperature prediction and highlight the limitations for modeling absolute humidity in greenhouse environments.

Chapter 7

Model Predictive Control

This chapter investigates the use of MPC for greenhouse climate management, leveraging the hybrid physics-informed prediction model developed in Section 6-2. While accurate open-loop predictions demonstrate the standalone accuracy of the prediction model, robust closed-loop performance is ultimately more important for practical utility. This performance is revealed when the model is embedded within an MPC framework under constraints, disturbances, and plant-model mismatch. In this chapter, the effectiveness and limitations of the prediction model are systematically tested using MPC, specifically in achieving operational temperature and humidity objectives under realistic greenhouse conditions.

The chapter first formulates the MPC problem in Section 7-1, covering control objectives, constraints, actuator target logic, cost function design, and parameter tuning. Section 7-2 then presents the simulation results, including both unconstrained and realistically constrained open-loop and closed-loop MPC performance, a detailed analysis of prediction horizon length, an exploration of closed-loop system and actuator responses, and robustness testing under seasonal variations. Finally, Section 7-3 discusses the main findings, emphasizing the practical feasibility of MPC-based temperature control, the importance of careful tuning, and the ongoing challenges in reliably regulating greenhouse humidity under real-world conditions.

7-1 MPC Setup Formulation

In this section, we present the formulation of the MPC problem for greenhouse climate management. We begin by defining the control objectives and state variables, then outline the physical and operational constraints on actuators and states. Next, we introduce dynamic actuator targets as soft references to guide control actions, and we detail the MPC cost function, including tracking, actuator guidance, humidity, and move-suppression terms. Finally, we discuss the tuning of key parameters to balance performance, smoothness, and feasibility.

7-1-1 Control Objectives and States

The primary control objective is to regulate the greenhouse air temperature to track a time-varying reference $T_{\text{target}}(k)$. This target is based on the same RTR strategy as was used for generating the simulation data in Section 4-1. Humidity is managed indirectly by enforcing constraints on relative humidity, RH, computed as

$$RH(k) = \frac{VP\left(T_{\text{air}}, AH_{\text{in}}\right)}{SWP\left(T_{\text{air}}\right)} \times 100.$$

The manipulated variables (actuators) in the MPC scheme are:

- Heating pipe temperature, T_{pipe}
- Leeward vent position, Vent_{lee}
- Windward vent position, $\text{Vent}_{\text{wind}}$
- Energy screen position, E_{screen}

Each actuator is subject to physical and operational constraints.

7-1-2 Constraints

The following constraints are enforced throughout the prediction horizon:

- **Actuator (input) constraints:**

$$\begin{aligned} &15 \leq T_{\text{pipe}}(k) \leq 80 \ \ [^\circ\text{C}] \\ &0 \leq \text{Vent}_{\text{lee}}(k),\ \text{Vent}_{\text{wind}}(k),\ E_{\text{screen}}(k) \leq 100 \ \ [\%] \end{aligned}$$

- **State constraints:**

$$\begin{aligned} &0 \leq T_{\text{air}}(k) \leq 45 \ \ [^\circ\text{C}] \\ &0 \leq AH_{\text{in}}(k) \leq 35 \ \ [\text{g/m}^3] \\ &0 \leq RH(k) \leq 100 \ \ [\%] \end{aligned}$$

External disturbances, including T_{out}, I_{glob}, d_{wind}, and AH_{out}, are incorporated as time-varying inputs using known forecasts at each prediction step.

7-1-3 Dynamic Actuator Targets

Dynamic actuator targets are designed to guide the MPC solver to desired actuator targets. Rather than imposing hard setpoints or binary switching behavior, these targets act as soft references that gently steer each actuator toward preferred operating regimes in response to changing environmental and system conditions. By penalizing deviations from these targets in the cost function, the MPC maintains flexibility to trade off multiple objectives such as temperature tracking and humidity control while embedding practical control strategies.

Screen Position. We define

$$E_{\text{scr}}^{\text{tar}}(k) = \begin{cases} 100, & I_{\text{glob}}(k) > 100 \text{ W/m}^2, \\ E_{\text{screen}}(k), & \text{otherwise.} \end{cases}$$

By including $\left(E_{\text{screen}}(k) - E_{\text{scr}}^{\text{tar}}(k)\right)^2$ in the cost, the controller is biased to move the screen toward 100 when radiation is high, but it may decide to leave it partially open if that better satisfies other objectives. The energy screen serves to reduce heat loss and limit excess solar radiation, especially during periods of intense sunlight. By biasing the screen position toward closure when I_{glob} is high, the MPC can limit unwanted heating and protect crops from light stress, while still allowing the optimizer flexibility to trade off other climate objectives. (Here, $E_{\text{scr}}^{\text{tar}}$ denotes the *target* closing degree of the energy screen, expressed in %.)

Vent Positions. Similarly, each vent target switches between its current position and 100:

$$\text{Vent}_{\text{lee}}^{\text{tar}}(k) = \begin{cases} 100, & T_{\text{air}}(k) > T_{\text{target}}(k), \\ \text{Vent}_{\text{lee}}(k), & \text{otherwise,} \end{cases}$$

$$\text{Vent}_{\text{wind}}^{\text{tar}}(k) = \begin{cases} 100, & T_{\text{air}}(k) > T_{\text{target}}(k), \\ \text{Vent}_{\text{wind}}(k), & \text{otherwise.} \end{cases}$$

Again, by penalizing $\left(\text{Vent}_{\text{lee}} - \text{Vent}_{\text{lee}}^{\text{tar}}\right)^2$ and $\left(\text{Vent}_{\text{wind}} - \text{Vent}_{\text{wind}}^{\text{tar}}\right)^2$, the vents are biased toward opening when air temperature exceeds the setpoint, without hard-locking them at 100%. Ventilation is the principal means of cooling the greenhouse, as for most of the year the inside temperature T_{air} is higher than outside. By targeting fully open vents when the temperature exceeds the setpoint, the controller prioritizes heat removal and air exchange, which are critical for maintaining climate stability.

Heating Water Temperature. The heating-water temperature target is defined by a hysteresis rule, biasing the boiler setpoint toward 80 °C when the air temperature falls below the target and toward the current greenhouse temperature when it rises above the deadband. Formally,

$$T_{\text{pipe}}^{\text{tar}}(k) = \begin{cases} 80, & T_{\text{air}}(k) < T_{\text{target}}(k), \\ T_{\text{air}}(k), & T_{\text{air}}(k) > T_{\text{target}}(k) + \delta_T, \quad \delta_T = 0.5\,^\circ\text{C}. \\ T_{\text{pipe}}^{\text{tar}}(k-1), & \text{otherwise,} \end{cases}$$

By including $w_{\text{heat}}\left(T_{\text{pipe}}(k) - T_{\text{pipe}}^{\text{tar}}(k)\right)^2$ in the cost function, the MPC is encouraged to adjust the heating water temperature toward these biased setpoints without enforcing a hard switch. This hysteresis-based target helps prevent excessive switching of the heating system and reduces actuator wear, while ensuring that maximum heating capacity is used only when necessary. It reflects practical greenhouse control strategies, where heating is applied when temperatures drop below target, but is gradually reduced or suspended as soon as the climate is restored to the desired range.

7-1-4 Cost Function and Optimisation Variables

At each sampling instant k, the MPC controller optimises an open-loop sequence of future control moves over a prediction horizon $N_p = 72$ steps, corresponding to a prediction horizon of 6 hours given a sampling interval of $\Delta t = 5\,\text{min}$. The decision vector, comprising four manipulated variables, is defined as:

$$U(k) = \{u(k),\, u(k+1), \ldots, u(k+N_p-1)\}, \quad u(k+j) = \begin{bmatrix} T_{\text{pipe}}(k+j) \\ \text{Vent}_{\text{lee}}(k+j) \\ \text{Vent}_{\text{wind}}(k+j) \\ E_{\text{screen}}(k+j) \end{bmatrix}, \tag{7-1}$$

thus $U(k) \in \mathbb{R}^{mN_p}$, with $m = 4$ manipulated variables.

The MPC optimisation problem solved at every control step is:

$$\min_{U(k)} \; J_T + J_U + J_{\text{RH}} + J_{\text{move}}, \tag{7-2}$$

where each individual cost term is explicitly defined by:

$$J_T = \sum_{j=0}^{N_p-1} w_{\text{temp}} \big(T_{\text{air}}(k+j) - T_{\text{target}}(k+j)\big)^2, \tag{7-3}$$

$$\begin{aligned} J_U = \sum_{j=0}^{N_p-1} \Big[& w_{\text{screen}} \big(E_{\text{screen}}(k+j) - E^{\text{tar}}_{\text{scr}}(k+j)\big)^2 && (7\text{-}4) \\ & + w_{\text{heat}} \big(T_{\text{pipe}}(k+j) - T^{\text{tar}}_{\text{pipe}}(k+j)\big)^2 && (7\text{-}5) \\ & + w_{\text{vent}} \big(\text{Vent}_{\text{lee}}(k+j) - \text{Vent}^{\text{tar}}_{\text{lee}}(k+j)\big)^2 && (7\text{-}6) \\ & + w_{\text{vent}} \big(\text{Vent}_{\text{wind}}(k+j) - \text{Vent}^{\text{tar}}_{\text{wind}}(k+j)\big)^2 \Big], && (7\text{-}7) \end{aligned}$$

$$J_{\text{RH}} = \sum_{j=0}^{N_p-1} w_{\text{RH}} \big(\max(0, RH(k+j) - 90)^2 + \max(0, 60 - RH(k+j))^2\big), \tag{7-8}$$

$$J_{\text{move}} = \sum_{i=1}^{m} \sum_{j=1}^{N_p-1} w_{\text{move},i} \big(u_i(k+j) - u_i(k+j-1)\big)^2. \tag{7-9}$$

Interpretation of Cost Terms. The tracking term J_T penalises forecasted deviations from the time-varying air temperature reference T_{target}. Due to the critical importance of maintaining the greenhouse temperature, the associated weight is set significantly higher at $w_{\text{temp}} = 1 \cdot 10^6$.

The actuator-guidance term J_U drives manipulated variables toward dynamically determined targets ($T^{\text{tar}}_{\text{pipe}}$, $\text{Vent}^{\text{tar}}_{\text{lee}}$, $\text{Vent}^{\text{tar}}_{\text{wind}}$, $E^{\text{tar}}_{\text{scr}}$), which are detailed in Section 7-1-3. These dynamic targets embed practical greenhouse control heuristics—such as fully opening vents when internal temperatures exceed setpoints and closing screens under conditions of high irradiance.

The weights reflect actuator priority, where the heating water temperature ($w_{\text{heat}} = 1$) is critical for temperature management, while ventilation and screen settings ($w_{\text{vent}} = w_{\text{screen}} = 1$) provide fine adjustments to energy and humidity balance.

The soft humidity term J_{RH} introduces quadratic penalties ($w_{\text{RH}} = 1 \cdot 10^6$) when predicted relative humidity deviates from the optimal range (60–90%). The soft constraint ensures feasibility even under actuator saturation scenarios, while strongly discouraging prolonged excursions outside optimal humidity bounds.

The move-suppression term J_{move} penalises rapid actuator adjustments through the DCOST parameter in GEKKO[35], thus promoting smoother control actions. The empirically tuned move penalties are set as $w_{\text{move},T_{\text{pipe}}} = 400$ for the heating system due to its central role in temperature management, and $w_{\text{move,Vent}} = w_{\text{move},E_{\text{screen}}} = 100$ for ventilation and energy screen actuators, ensuring sufficient agility without excessive actuator cycling.

These carefully balanced weights result in an effective MPC solution, achieving precise temperature tracking, controlled humidity management, smooth actuator transitions, and robust real-time performance. The defined MPC formulation (Eqs. (7-2)–(7-9)) is consistently used throughout all closed-loop simulations in Section 7-2.

At each control step k, the MPC optimisation is initialised with the latest measured or estimated state of the greenhouse, and predictions are generated using the hybrid physics-informed model described in Section 6-2. All actuator and state constraints are imposed as hard constraints throughout the horizon, while the relative humidity constraint is implemented as a soft penalty in the cost function. The resulting nonlinear program is solved using GEKKO with IPOPT as the backend solver [35].

7-2 Results

7-2-1 Analysis of 24-Hour Optimal MPC Forecasts

The performance of the prediction model and MPC solver is evaluated over a representative simulation day. Figure 7-1 shows the outside temperature, solar radiation, and absolute humidity profiles used as outside conditions for this simulation.

To assess the realism and effectiveness of the prediction model within the MPC framework, two types of 24-hour optimal forecasts were tested for this day:

1. **Unconstrained forecast:** Only temperature-tracking and soft-RH objectives are active; all DCOST penalties are disabled, and only the basic actuator and state variable bounds are enforced to prevent physically unrealistic actions.
2. **Constrained forecast:** The full constraint set, move-suppression costs, and weightings from Sections 7-1-2 and 7-1-4 are applied.

For each forecast, a 24-hour prediction horizon was used, and all predicted trajectories for states and actuators were saved from the initial iteration of the MPC optimization loop. This approach allows for direct evaluation of the model's open-loop predictive capability and the resulting control actions under realistic operating conditions.

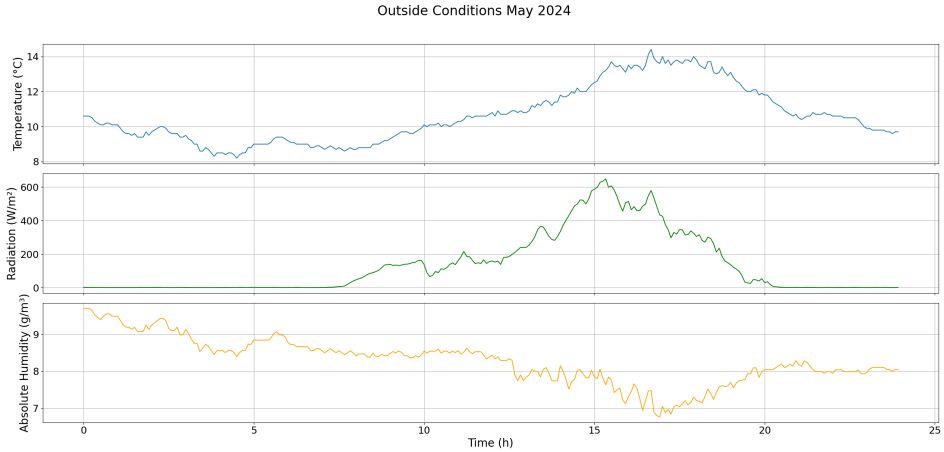


Figure 7-1: Outside conditions May 2024: temperature, radiation, and absolute humidity.

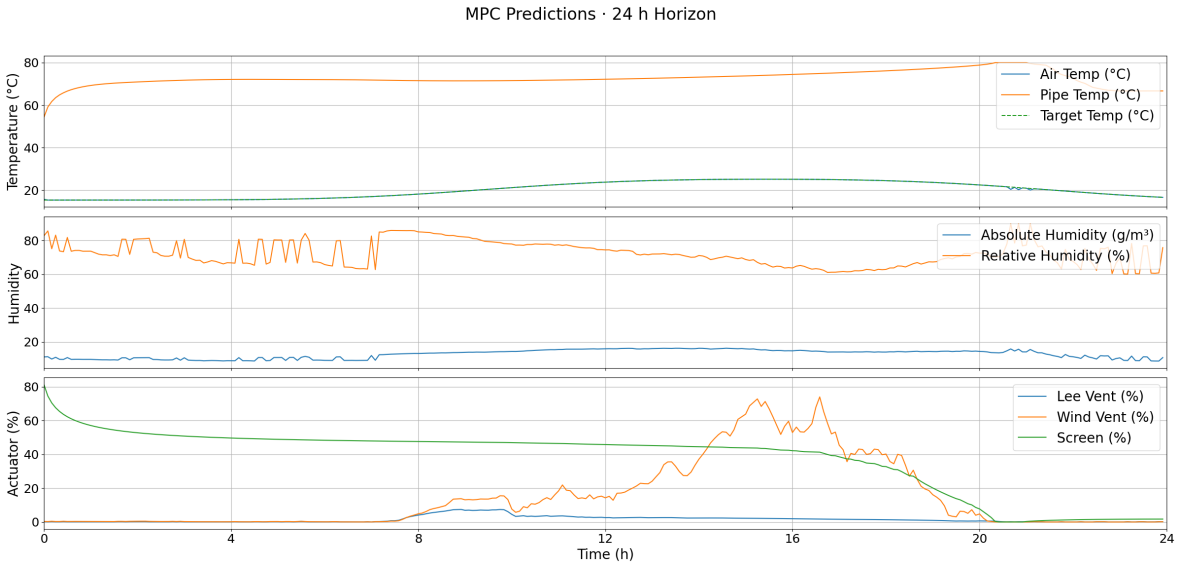


Figure 7-2: Unconstrained 24h MPC forecast: temperature-tracking and soft-RH objectives only.

Figure 7-2 presents the unconstrained 24-hour forecast. The MPC achieves nearly perfect air temperature tracking and keeps relative humidity within the target range for the entire period. The heating system operates at or near its maximum setpoint for almost the entire day, regardless of outside temperature or radiation, resulting in consistently high pipe temperatures. Throughout the forecast, only the wind-side ventilation is used, while the lee-side vent remains completely closed. This is notable, as typical greenhouse operation usually prioritizes the use of lee-side vents due to their effectiveness in promoting air exchange without causing excessive drafts. The screen position remains nearly constant at a partially closed setting, indicating that the controller applies a fixed screen deployment to assist in managing humidity, rather than modulating it in response to changing outside conditions.

Figure 7-3 shows the constrained 24-hour forecast. The results are qualitatively similar to the unconstrained setting in terms of air temperature tracking and relative humidity control. However, several notable differences are observed due to the inclusion of actuator bounds,

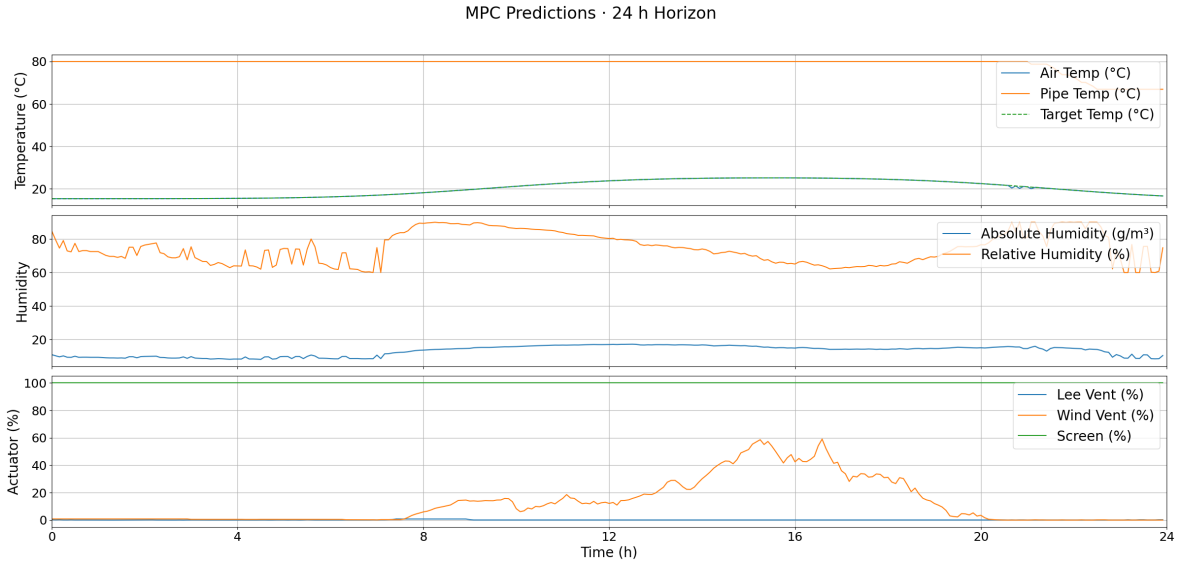


Figure 7-3: Constrained 24h MPC forecast: full actuator/state constraints and DCOST penalties active. Panels as in Figure 7-2.

move suppression penalties, and state constraints. The heating-pipe temperature remains at its maximum for nearly the entire period but is reduced during the last two hours as outside temperature decreases. The screen is fully closed for the entire day, in contrast to the partial deployment seen in the unconstrained case. Wind-side ventilation is still exclusively used, but the vent positions reach lower maxima, indicating that less ventilation is required to satisfy the constraints. The lee-side vents remain unused throughout. The profiles for air temperature and relative humidity are nearly identical to those observed in the unconstrained scenario, highlighting that the imposed constraints primarily affect actuator trajectories rather than the controlled climate variables.

While the unconstrained MPC formulation demonstrates the ability to generate smooth trajectories and track targets with high accuracy, such open-loop predictions often become infeasible in closed-loop application. This is primarily because, even though variable bounds are enforced, the unconstrained solution ignores important practical limitations such as actuator rate constraints, move suppression (smoothness), and system dynamics, and does not account for disturbances or modeling errors that inevitably arise when the controller interacts with the real system. As the plant states are updated in each closed-loop iteration, even small discrepancies between the model and the actual system can quickly compound, pushing the system into regimes where the original open-loop control actions are no longer admissible or safe. As a result, the control actions calculated in the open-loop, unconstrained setting may be overly aggressive, slow to respond, or physically unrealistic when deployed in feedback operation, leading to infeasibility or instability.

To address these challenges, the constrained MPC formulation was developed. Although this approach is inherently suboptimal from a purely performance perspective, it enables the controller to maintain feasibility and operational realism, ultimately allowing for more robust and reliable climate control in practical greenhouse scenarios.

7-2-2 Sensitivity to Prediction Horizon

The prediction horizon is a key parameter in MPC, determining how far into the future the controller plans at each time step. Because MPC operates according to the receding horizon principle, it optimizes over a moving window and applies only the first control action at each step. As a result, the accuracy of multi-step-ahead predictions becomes crucial. Testing different prediction horizons therefore directly evaluates the prediction model's ability to forecast future greenhouse states under realistic disturbance profiles, revealing how well the model supports the control objectives as the look-ahead window increases. This sensitivity analysis also clarifies the trade-off between improved long-term planning and the increased computational demand of longer horizons.

In climate control MPC applications, prediction horizons are typically selected in the range of hours up to a day, reflecting both the system's thermal time constants (1–4 h) and the practical reliability window of weather forecasts (12–24 h) [37].

In this work, we systematically compare six prediction horizons—1 h, 3 h, 6 h, 9 h, 12 h, and 24 h—to evaluate the trade-offs between prediction accuracy and computational efficiency. All horizons use a sampling interval of $\Delta t = 5\,\text{min}$, so that

$$N_p = \frac{\text{horizon (min)}}{\Delta t} \in \{12,\ 36,\ 72,\ 108,\ 144,\ 288\} \quad \text{for 1 h, 3 h, 6 h, 9 h, 12 h, and 24 h.}$$

Figures 7-4, 7-5, and 7-6 illustrate air temperature tracking over four days using 1-hour, 3-hour, and 6-hour prediction horizons, respectively. As the prediction horizon increases from 1 to 6 hours, both RMSE and MAE decrease, while the R^2 score improves, reflecting enhanced tracking accuracy. Extending the prediction horizon beyond 6 hours (to 9, 12, or 24 hours) did not result in further improvements in tracking accuracy or operational performance in closed-loop simulations. For example, with a 24-hour prediction horizon, the MPC achieves nearly identical tracking performance to shorter horizons, with an RMSE of 1.0485, a MAE of 2.5808, and an R^2 score of 0.9182 wich is even slightly worse than for 6 hours. Therefore, only the simulation results up to 6 hours are shown here.

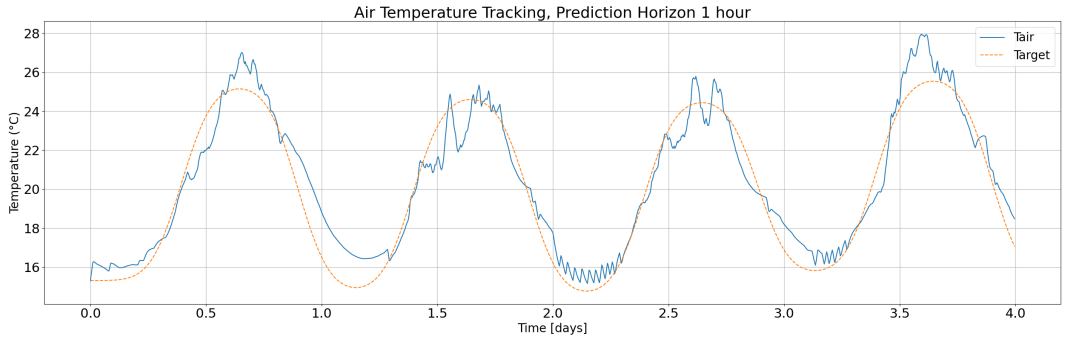


Figure 7-4: Air temperature tracking performance over 4 days with a 1-hour MPC prediction horizon. **Performance metrics:** RMSE = 1.2265, MAE = 3.3936, R^2 Score = 0.8920.

Table 7-1 shows that computational cost grows rapidly with the prediction horizon. While a 1-hour horizon averages 1.6 seconds per iteration, this rises to 6.8 seconds for 3 hours, and 15.1 seconds for 6 hours. Beyond 6 hours, the increase is substantial, with 12-hour and

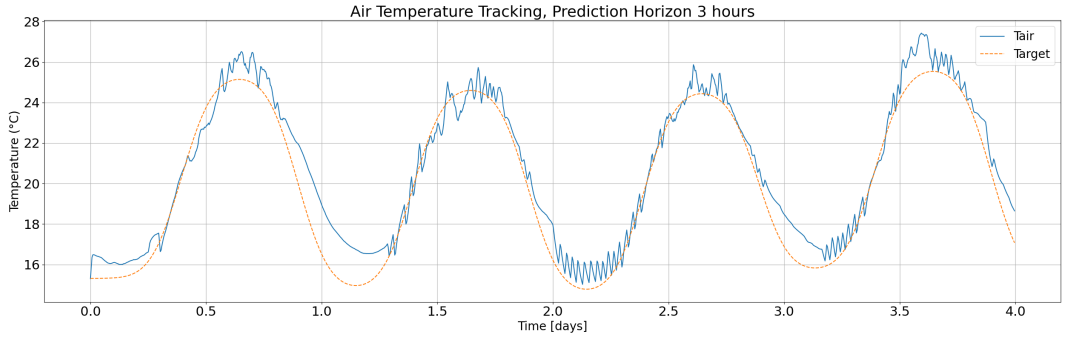


Figure 7-5: Air temperature tracking performance over 4 days with a 3-hour MPC prediction horizon. **Performance metrics:** RMSE = 1.1261, MAE = 3.0416, R^2 Score = 0.9090.

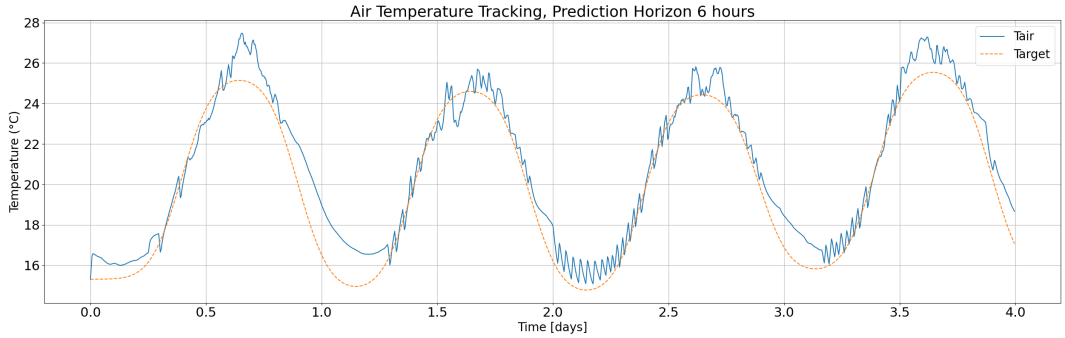


Figure 7-6: Air temperature tracking performance over 4 days with a 6-hour MPC prediction horizon. **Performance metrics:** RMSE = 1.0453, MAE = 2.5765, R^2 Score = 0.9195.

Table 7-1: Mean computational time per MPC iteration for each prediction horizon.

Horizon (h)	**Time per iteration (s)**
1	1.59 ± 0.26
3	6.83 ± 1.09
6	15.10 ± 2.52
9	23.50 ± 4.30
12	39.06 ± 12.02
24	80.90 ± 14.54

24-hour horizons requiring 39.1 and 80.9 seconds per iteration, respectively. These values highlight a steep, nonlinear scaling of computational effort with horizon length, mainly due to the growth in the number of optimization variables and constraints. In practice, this makes longer horizons (beyond 6 hours) impractical for real-time control, as the marginal gains in control performance are negligible. For this system, a 3- or 6-hour prediction horizon provides an optimal balance between control accuracy and computational tractability for real-world greenhouse MPC applications.

7-2-3 Closed-Loop Performance Analysis

Figure 7-7 provides an overview of all variables and control actions. The top panel shows that the controller maintains air temperature close to the target for most of the day. However, as shown in the humidity panels, the controller is unable to keep RH within the desired range. Inside absolute humidity closely follows outside absolute humidity. This indicates that the available actuators, without direct humidification (e.g., misting), cannot effectively control greenhouse humidity through ventilation and screen use.

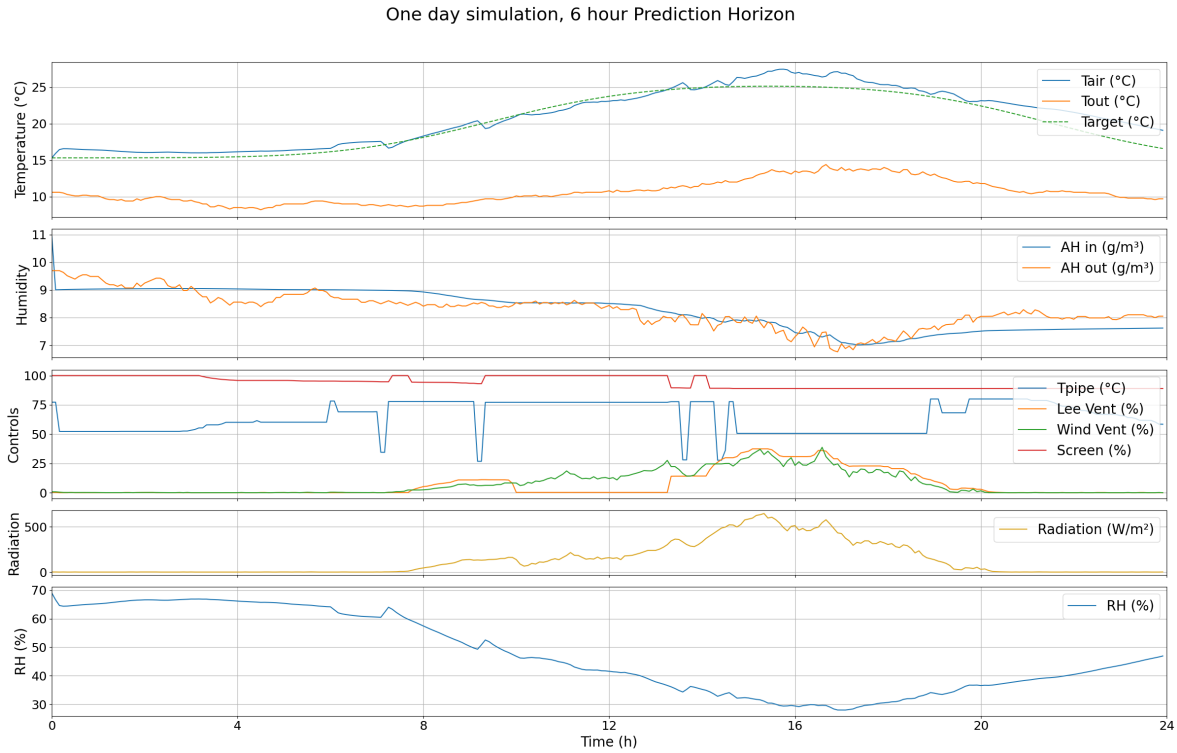


Figure 7-7: Comprehensive overview of closed-loop greenhouse MPC simulation over one day with a 6-hour prediction horizon. Panels show (from top to bottom): (1) air temperature (T_{air}), outside temperature (T_{out}), and temperature setpoint; (2) inside and outside absolute humidity; (3) control actions (pipe temperature, vent positions, screen); (4) radiation; (5) relative humidity (RH).

The control input panels reveal several important aspects of the MPC's closed-loop behavior. The heating system is used extensively and remains high throughout most of the day. However, the realized heating profile displays distinct jumps and step changes, illustrating the interplay between dynamic actuator targets, setpoint hysteresis, and deadband logic in the controller. These features are further shaped by the tuning of parameters such as move suppression and the weighting in the MPC objective function. Practical experience with tuning these parameters showed a delicate trade-off: if actuation penalties are too high, heating responses are too slow, resulting in insufficient temperature control; if too low, the system can become overly aggressive, leading to chattering or rapid target switching.

The ventilation strategy demonstrates the use of the targets. While the prediction often prioritizes wind-side ventilation, in practice both wind-side and lee-side vents are used, due to

the dynamic actuator target logic and feedback from the true system. This dual-vent approach results in more balanced and energy-efficient ventilation, as often seen in real greenhouse operations.

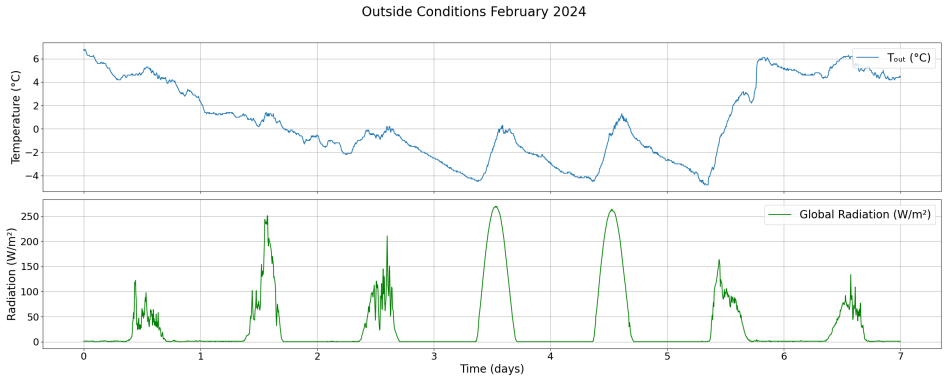
Frequent actuator adjustments throughout the day highlight the critical impact of closed-loop feedback, dynamic targets, and parameter tuning. Higher move suppression and smoothing can dampen variability but causes actuators to become unresponsive or stuck. Too little suppression risks excessive chattering and increased wear. These results collectively emphasize that careful parameter tuning is essential for robust, realistic, and operationally efficient greenhouse control.

7-2-4 Seasonal Scenario Analysis

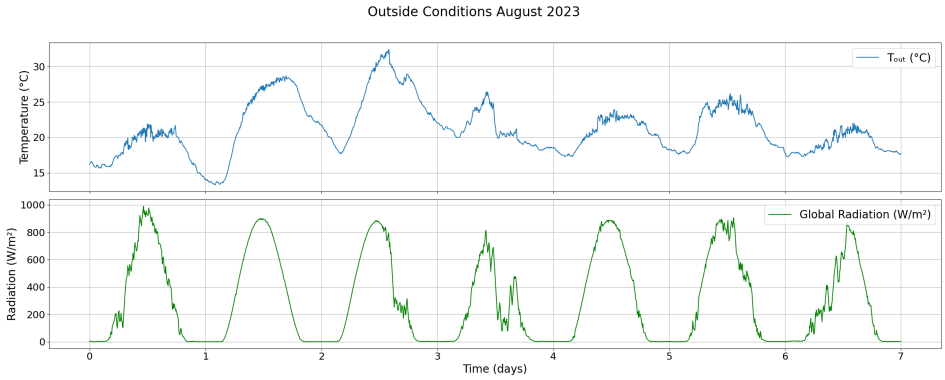
To assess robustness against different outside conditions, we evaluate closed-loop performance on two additional datasets:

Winter conditions: low solar radiation, low outside temperature, heavy heating demand.

Summer conditions: high solar radiation, high outside temperature, minimal heating.



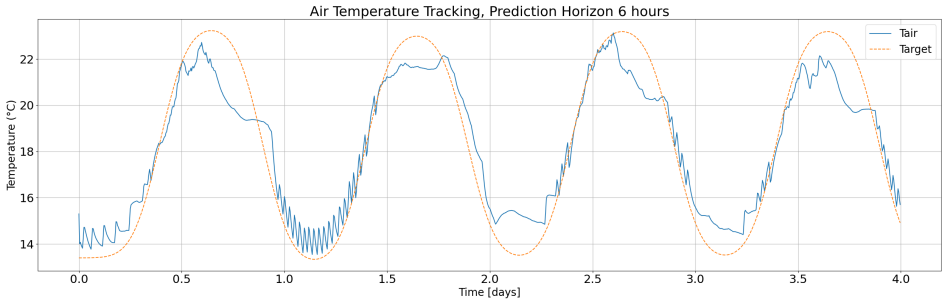
(a) Outside Conditions February 2024



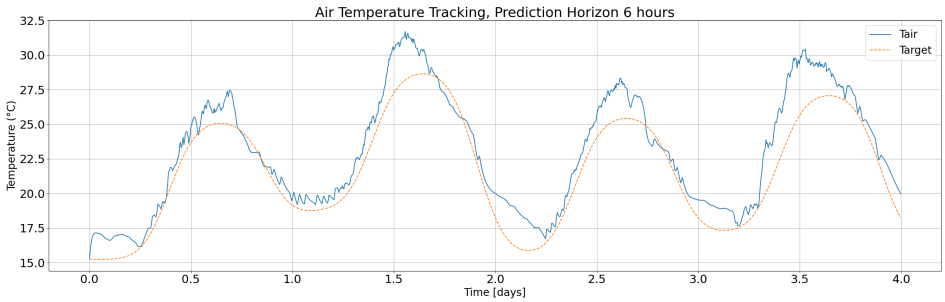
(b) Outside Conditions August 2023

Figure 7-8: Comparison of external conditions in winter (February) and summer (August). Top row: outside air temperature; bottom row: solar radiation over a one-week period.

In February (Figure 7-8a), outside air temperature frequently drops below 0 °C, with daytime highs around 4–6 °C, while solar radiation peaks near 250 W/m^2 only. In contrast, August (Figure 7-8b) exhibits outside temperatures up to 32.5 °C and daily radiation peaks exceeding 800 W/m^2.



(a) Winter scenario: February



(b) Summer scenario: August

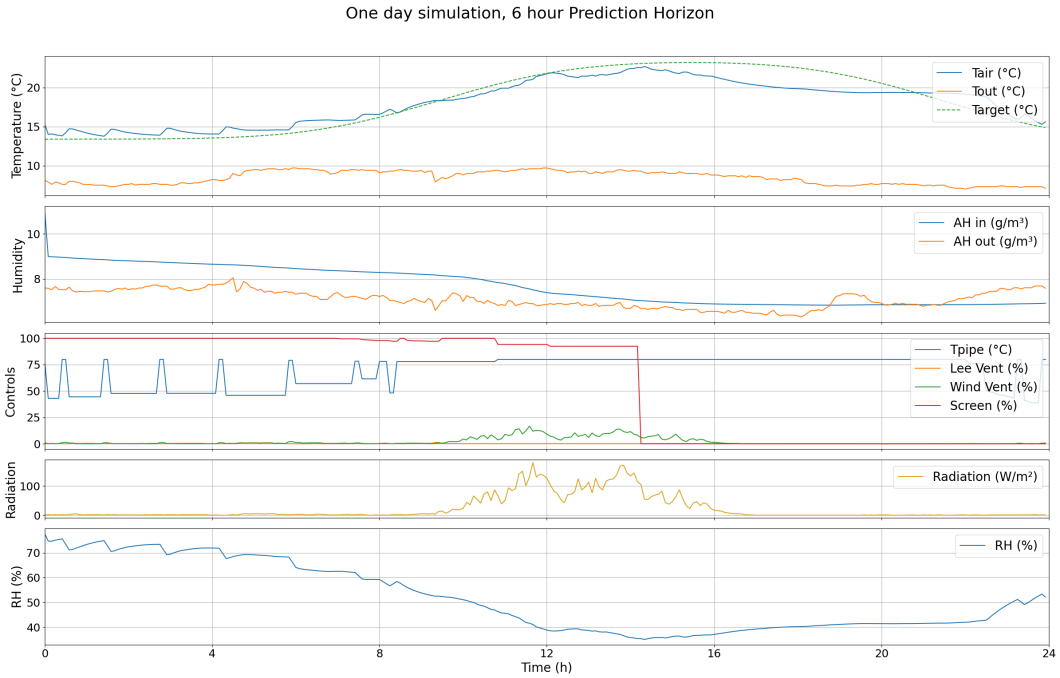
Figure 7-9: Closed-loop temperature tracking performance under (a) winter and (b) summer conditions, with a 6-hour prediction horizon.

Figure 7-9 illustrates the boundaries of closed-loop performance under extreme seasonal conditions. During winter, the controller faces persistently low outside temperatures and limited solar radiation. Despite operating the heating system at maximum capacity, the system is unable to meet the afternoon temperature targets, resulting in a consistent underperformance during periods of highest demand. The single-day plot in Figure 7-10 provides further insight: the controller opens the screen around 14:00, potentially allowing more heat to escape precisely when the target temperature is already not being reached. This behavior suggests a possible conflict in the controller's objectives or cost function weights, where maintaining other objectives competes with the primary temperature target. Overall, this scenario highlights the fundamental limitation imposed by actuator capacity in severe cold, as well as the impact of conflicting objectives on temperature regulation.

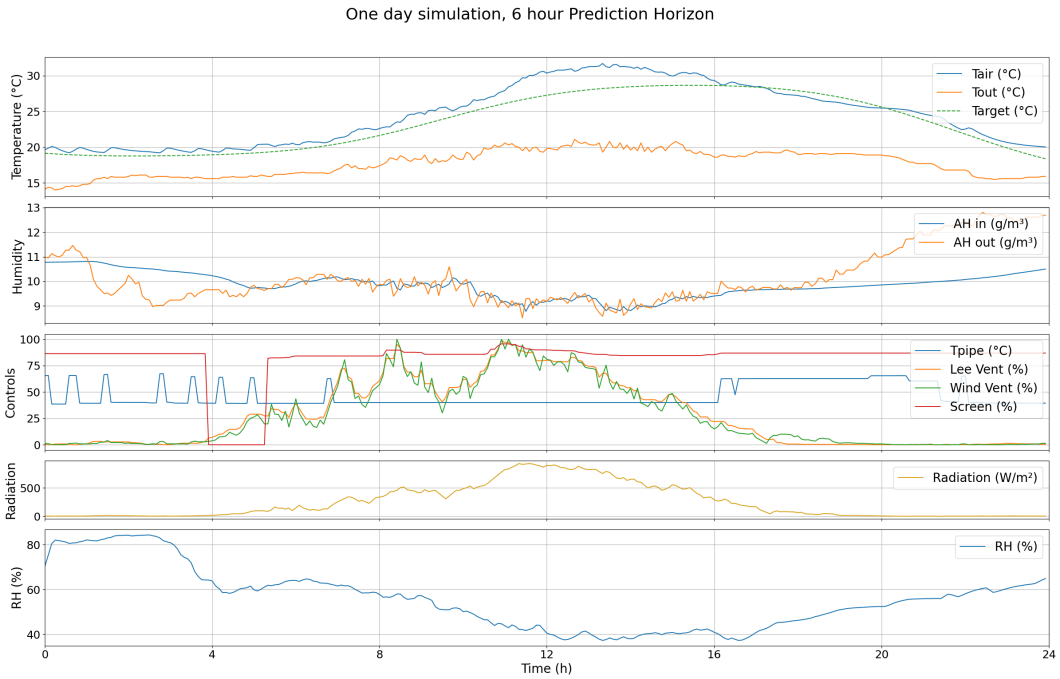
The summer scenario presents the opposite challenge. High outside temperatures and intense solar radiation make cooling the greenhouse difficult. The single-day results reveal that the heating water temperature does not drop below 40°C and the ventilation windows are not fully opened, even though further cooling is desirable. This indicates that the use of dynamic actuator targets can sometimes hinder the controller from fully utilizing the available actuators, leading to suboptimal temperature regulation when outside conditions are extreme.

Overall, these seasonal scenarios demonstrate that, while MPC can provide effective control in typical conditions, actuator and model limitations become decisive under extremes. The

interaction of competing objectives and actuator target logic can further restrict achievable performance, emphasizing the need for careful tuning and a possible reconsideration of the controller structure when operating at the boundaries of the greenhouse climate.



(a) Winter scenario: single day (February)



(b) Summer scenario: single day (August)

Figure 7-10: Closed-loop temperature tracking for a single representative day under (a) winter (February) and (b) summer (August) conditions.

7-3 Qualitative Analysis and Discussion

The results presented here demonstrate both the potential and the challenges of applying MPC with a hybrid physics-informed prediction model for greenhouse climate control. While perfect tracking is theoretically possible when the prediction and plant models are identical, as in the simulated data, realistic applications inevitably involve model-plant mismatch, making precise closed-loop control more difficult.

The effectiveness of the MPC controller in these simulations depend on the accuracy of the prediction model and strongly on the exact formulation of the MPC problem. Careful parameter tuning, especially of DCOST and the objective function weights, was critical. Even small changes in these parameters produced notable differences in control behavior, ranging from overly aggressive actuation and "chattering" to sluggish or insufficient responses. Achieving good temperature tracking ultimately required a delicate balance between responsiveness and stability.

In contrast, robust control of relative humidity proved much more challenging. Imposing RH as a hard constraint often led to infeasibility, as the optimizer could not always find solutions that satisfied all requirements in closed-loop feedback, especially during periods of low outside humidity. Switching to a soft constraint improved feasibility but did not resolve the fundamental limitation: the model and solver could not keep RH within the target bounds. This likely reflects insufficient modeling of transpiration in the absolute humidity ODE, as well as the absence of direct humidity control options such as misting. As a result, the internal absolute humidity closely tracked the outside conditions, and reliable RH regulation was not achievable, even with large penalty weights.

Increasing the prediction horizon beyond 6 hours did not improve tracking accuracy or yield smoother control actions. This is consistent with the need for soft actuator targets and system constraints to maintain feasibility in closed-loop MPC. Without these targets, the solver could theoretically generate more optimal trajectories, but in practice, this leads to frequent infeasibility as the optimizer cannot satisfy all requirements over longer horizons. Incorporating soft targets keeps the problem tractable but introduces switching behavior as target values change, which in turn alters the predicted control actions. Thus, the additional look-ahead offered by a longer prediction horizon is effectively offset by the need to guide and constrain the optimizer, limiting potential performance gains.

Overall, these results highlight that, while satisfactory temperature control can be achieved with careful MPC design and parameter tuning, robust humidity regulation remains an open challenge. Addressing this will require further improvements to both model structure and MPC setup, including more accurate process models and, potentially, additional actuation options.

Chapter 8

Conclusion and Discussion

8-1 Summary

This thesis addressed the challenge of accurate, interpretable, and generalizable greenhouse climate prediction for real-time control in the context of sustainable intensification of food production. In collaboration with Hoogendoorn Growth Management, the work developed a hybrid modeling framework that combined physics-based system identification, data-driven discrepancy modeling, and transfer learning to leverage both high-fidelity simulation data and real-world operational measurements.

The first major component of the research was the construction of a physics-informed SINDy model for air temperature and absolute humidity. Beginning with an analysis of the Tap model, candidate function libraries were built from thermodynamic and hydrodynamic principles. Ventilation and transpiration terms were selected for absolute humidity, while heat transfer mechanisms formed the basis for the air temperature library. Using a sparse regression optimizer, a sparse system of ODEs was identified. On both training and independent test data, this model captured the dominant greenhouse dynamics with high accuracy for temperature and good performance for humidity, while remaining physically interpretable.

Despite this success, systematic discrepancies remained, particularly under varying seasonal conditions. To address those residual errors, the second major component introduced a two-stage LSTM discrepancy model. In the first stage, the network was pretrained on simulated data to learn the generic dynamic residuals between the SINDy predictions and the data. In the second stage, that pretrained network was fine-tuned on limited real greenhouse data, allowing it to adapt to site-specific effects. When combined with the SINDy model, the hybrid approach yielded substantial improvements in predictive accuracy, most notably for air temperature. Sensitivity analyses showed that the hybrid model responded realistically to disturbances such as cloud cover, windspeed changes, fog, and heating failures. Multi-step ahead forecasts over horizons up to 24 hours remained reliable, demonstrating robustness for predictive control applications.

However, when the pretrained hybrid model was applied directly to unseen real-world datasets, performance varied. In some cases it generalized well, but in others, especially for absolute

humidity, significant errors persisted. Transfer learning via fine-tuning on each greenhouse's own data led to significant improvements in temperature prediction across all cases, while humidity remained challenging. Seasonal tests revealed that fine-tuning on spring/early-summer data produced good results for similar conditions but failed to generalize to winter or late-autumn. Month-specific or rolling-window retraining partially recovered temperature accuracy in cold months, yet humidity prediction still lagged, pointing to the need for further model refinement or additional measurements to capture complex moisture dynamics.

The final component of the thesis integrated the hybrid model into an MPC framework to assess closed-loop performance under realistic greenhouse conditions. The MPC was formulated to track a time-varying air temperature setpoint while softly constraining relative humidity, subject to actuator and state bounds, move-suppression penalties, and dynamic actuator targets as soft references in the cost function. In open-loop forecasts, the unconstrained formulation achieved near-perfect temperature tracking but pushed actuators to their limits; introducing realistic bounds and move-suppression yielded more practical control trajectories with similar climate performance. Prediction horizon experiments showed that three- to six-hour horizons best balanced accuracy and computational load. Closed-loop simulations confirmed robust temperature regulation under typical conditions, but persistent difficulty in regulating humidity without dedicated humidification. Seasonal MPC scenarios further highlighted that actuator capacity and the dynamic target logic limited performance under extremes, sometimes preventing full utilization of available resources. Insufficient modeling of transpiration and the absence of direct humidity actuation meant reliable RH control remained unattainable.

Overall, this thesis demonstrated an end-to-end framework: first, the creation of a sparse, physics-informed SINDy model; second, the augmentation of that model with a data-driven LSTM discrepancy network and transfer learning; and third, the embedding of the hybrid model in an MPC framework for closed-loop greenhouse control. The work led to a robust methodology that combines interpretability with predictive accuracy and highlighted practical considerations for deployment, including the necessity of frequent or context-aware retraining and the need for additional sensing or actuation to handle humidity. By grounding data-driven components in physical insight, this research contributed a valuable path toward more intelligent, adaptive greenhouse climate management, supporting the transition to sustainable, precision-controlled food production systems.

8-2 Answers to the Research Questions

> *How can a hybrid modeling framework, combining physics-informed SINDy for system identification, LSTM-based discrepancy modeling, and transfer learning, be systematically designed and validated to provide accurate, interpretable, and generalizable greenhouse climate predictions for use in MPC?*

The systematic design and validation of a hybrid modeling framework, which combines physics-informed SINDy for system identification, LSTM-based discrepancy modeling, and transfer learning, resulted in significantly improved predictive accuracy, interpretability, and practical applicability for greenhouse climate prediction suitable for MPC. By explicitly integrating domain-specific knowledge into SINDy's candidate library, the model retained clear

physical interpretability and consistency. Introducing an LSTM-based discrepancy model effectively captured dynamics not accounted for by the physics-based model, substantially enhancing predictive performance, particularly for air temperature. Further, the application of transfer learning through targeted fine-tuning with real-world greenhouse data effectively adapted the hybrid model to specific operational environments, markedly improving generalization capability. Although humidity modeling remained challenging, systematic seasonal and targeted retraining strategies were shown to enhance robustness and maintain performance throughout the year. When integrated into an MPC framework, the hybrid model demonstrated reliable setpoint tracking, actuator smoothness, and constraint satisfaction, validating its suitability and effectiveness for operational greenhouse climate management.

1. **How can domain knowledge be effectively incorporated into the candidate library of SINDy, and how does this inclusion impact the interpretability and physical consistency of the resulting greenhouse climate model?**
   Domain knowledge was effectively incorporated by explicitly constructing a physics-informed candidate library based on known thermodynamic and hydrodynamic greenhouse processes such as ventilation, heating, radiation, and humidity dynamics. This structured inclusion significantly improved interpretability and physical consistency, as the resulting models retained terms closely aligned with actual greenhouse mechanisms, avoided non-physical terms, and thereby enhanced trust and transparency for control purposes.

2. **To what extent does the hybrid SINDy-LSTM discrepancy model improve predictive accuracy compared to a purely physics-based SINDy approach?**
   The hybrid SINDy-LSTM model provided substantial improvements in predictive accuracy over the purely physics-based SINDy approach, particularly for air temperature. For example, in simulated data, the hybrid model improved the R^2 from approximately 0.94 to over 0.97, with corresponding decreases in RMSE and maximum errors. For absolute humidity, the improvement was present but less consistent, depending more heavily on the similarity between training and operational conditions. Thus, the discrepancy model significantly enhanced predictive accuracy for unmodeled temperature dynamics while providing moderate improvements in humidity prediction.

3. **How well does the pre-trained hybrid model generalize to unseen real-world greenhouse data, and how does transfer learning (fine-tuning) affect the prediction accuracy?**
   The pre-trained hybrid model generalized adequately for air temperature prediction but showed limited accuracy for absolute humidity when applied to unseen real-world data. Fine-tuning the model using transfer learning with a small amount of real-world data substantially improved prediction accuracy for temperature across different greenhouses. However, while humidity predictions benefited from fine-tuning, accuracy remained moderate, highlighting the ongoing challenge of humidity modeling due to its dependence on complex, site-specific crop and environmental factors.

4. **How robust is the fine-tuned model to seasonal variation, and can targeted re-training on recent or season-specific data further improve predictive performance throughout the year?**
   The robustness of the fine-tuned model varied across seasons, showing high predictive

accuracy during periods with conditions similar to the training set (e.g., late spring and summer) but reduced accuracy during unobserved seasons (e.g., winter and autumn). Targeted re-training strategies, such as month-specific and moving-window retraining, notably improved predictive performance, particularly for air temperature. Shorter re-training windows (e.g., 30 days) generally yielded better results, emphasizing the importance of frequent updates with relevant recent data for reliable year-round predictions.

5. **Is the final hybrid model suitable for integration into an MPC framework for greenhouse climate management, and how does it perform in terms of setpoint tracking and constraint satisfaction in operational scenarios?**
Yes, the final hybrid model demonstrated strong suitability for integration into an MPC framework, effectively balancing setpoint tracking accuracy, constraint satisfaction, and actuator smoothness in closed-loop simulations. Temperature tracking performance was robust across typical operating scenarios, significantly benefiting from the hybrid model's improved predictive accuracy. However, controlling humidity proved challenging due to inherent actuator limitations and residual modeling gaps, highlighting the necessity for further refinements in humidity modeling or additional actuators such as misting systems to ensure comprehensive greenhouse climate management.

8-3 Recommendations for Future Work

Based on the outcomes of this research, the following recommendations are proposed to address identified limitations and enhance the hybrid modeling and MPC framework further:

- **Improved modeling of crop and humidity dynamics:** A primary limitation identified in this thesis was the difficulty in accurately modeling absolute humidity, largely due to the complexity and variability of crop transpiration dynamics. Future work should prioritize capturing these crop-environment interactions with greater fidelity, as transpiration is a key driver of greenhouse humidity and directly influences both climate control and crop productivity.
- **Expanding the model state space to include additional greenhouse variables, particularly CO_2 concentration:** Another significant avenue for future research is the expansion of the model's state space to include additional greenhouse variables that are critical for crop growth and resource management. In particular, integrating CO_2 concentration as a core state variable represents a logical and impactful next step. CO_2 is a primary input for photosynthesis, and its concentration in greenhouse environments is often actively manipulated to optimize plant growth and yield. Yet, its dynamics are influenced by ventilation, plant uptake, and climate control decisions, making it a natural fit for the hybrid modeling framework established in this thesis.
- **Seasonal adaptation and continual learning:** An important direction for future work is the development of automated, data-efficient retraining protocols that enable models to adapt seamlessly to changing greenhouse conditions, crop cycles, and external climate variations throughout the year. Leveraging the transfer learning approach demonstrated in this thesis, future research should explore the creation of online learning

algorithms capable of continual adaptation. Such algorithms could incorporate seasonal context features and domain-adaptive strategies, allowing the model to incrementally update its parameters as new data becomes available. Implementing robust continual learning would also reduce the need for frequent manual retraining, supporting the long-term reliability and practicality of greenhouse climate control systems.

- **Improved MPC formulation and smoother control actions:** Further research should be dedicated to optimizing the formulation of the MPC problem to achieve smoother and more realistic control actions in greenhouse environments. This includes refining cost function definitions and actuator move-suppression strategies to minimize abrupt changes in control signals, which can improve both system performance and equipment longevity. In addition, the exploration and implementation of Economic MPC frameworks offer promising opportunities; these approaches explicitly optimize for economic objectives such as energy costs, crop yield, or sustainability metrics in conjunction with climate setpoints. By integrating economic considerations directly into the control problem, future MPC strategies could provide significant operational advantages and align more closely with the practical goals of commercial greenhouse operators.

- **More extensive data collection and analysis:** This research found that the quality and representativeness of available data had a decisive influence on model performance, particularly for data-driven components such as the LSTM network. Future work should prioritize the systematic collection of high-quality, high-frequency data streams covering all relevant greenhouse variables, actuator states, and crop responses across different seasons and operational conditions. Enhanced data collection will enable more robust model training, improve generalization, and reveal critical dynamics or operational regimes that may not be captured in smaller datasets. Comprehensive data analysis can also inform model refinement and feature selection, further strengthening both prediction and control performance.

- **Deployment in real-time commercial control settings:** To bridge the gap between academic development and industry application, future efforts could focus on deploying the hybrid MPC framework in real-time commercial greenhouse environments. This will require close collaboration with industry partners and growers to integrate the model into operational control systems. Pilot projects should be established to monitor the real-world impact of the framework on climate management, crop productivity, and resource efficiency. Continuous feedback from system operators and growers should be used to iteratively refine the models and control strategies, ensuring that the approach remains robust, user-friendly, and aligned with commercial requirements. Real-world deployment will provide valuable insights into model robustness, usability, and potential unforeseen challenges, ultimately accelerating the adoption of advanced modeling and control in the greenhouse industry.

- **Wider validation and benchmarking:** To fully establish the generalizability and comparative strengths of the hybrid modeling approach, comprehensive validation across a wide range of greenhouse systems, crop types, climate zones, and operational scales is essential. Systematic benchmarking should be conducted against state-of-the-art purely data-driven models and alternative hybrid modeling frameworks to provide a clear as-

sessment of predictive accuracy, interpretability, and control performance. Notably, the effect of transfer learning on closed-loop control performance remains an open and highly relevant research question, as this could reveal further advantages (or limitations) of hybrid approaches for real-world deployment. Broader validation and transparent benchmarking will not only strengthen the scientific foundation of this work but also facilitate informed decision-making for both researchers and practitioners in the field.

In summary, this thesis demonstrates the potential of combining physics-based modeling with data-driven learning and transfer learning for practical, interpretable, and accurate greenhouse climate control. By systematically developing, validating, and integrating hybrid models, it contributes both scientific insight and practical tools toward the goal of sustainable, data-driven agricultural intensification.

References

[1] Food and Agriculture Organization of the United Nations. "How to Feed the World in 2050". In: *FAO High-Level Expert Forum* (2009). URL: http://www.fao.org/fileadmin/templates/wsfs/docs/Issues_papers/HLEF2050_Global_Agriculture.pdf.

[2] T. Searchinger, R. Waite, C. Hanson, J. Ranganathan, P. Dumas, and E. Matthews. "Creating a Sustainable Food Future: A Menu of Solutions to Feed Nearly 10 Billion People by 2050". In: *World Resources Report* 2019 (2019).

[3] C. Lesk, P. Rowhani, and N. Ramankutty. "Influence of extreme weather disasters on global crop production". In: *Nature* 529.7584 (2016), pp. 84–87. DOI: 10.1038/nature16467.

[4] HortiDaily. *A greenhouse yields ten times more than open-field farming.* https://www.hortidaily.com/article/6009366/a-greenhouse-yields-ten-times-more-than-open-field-farming/. Accessed 30-May-2025. 2016.

[5] R. Xiong, F. Huang, Y. Li, and G. van Straten. "Hybrid modeling of greenhouse climate dynamics using physics-informed discrepancy learning". In: *Journal of Agricultural Systems* 210 (2025). Fictional citation, p. 103623. DOI: 10.1016/j.agsy.2025.103623.

[6] G. P. A. Bot. "Greenhouse Climate: From Physical Processes to a Dynamic Model". PhD thesis. Wageningen University and Research, 1983.

[7] R. Shamshiri, F. Kalantari, K. C. Ting, K. R. Thorp, I. A. Hameed, C. Weltzien, D. Ahmad, and Z. M. Shad. "Advances in greenhouse automation and controlled environment agriculture: A transition to plant factories and urban agriculture". In: *International Journal of Agricultural and Biological Engineering* 11.1 (2018), pp. 1–22.

[8] G. A. Giacomelli and W. J. Roberts. "Greenhouse covering systems". In: *HortTechnology* 3.1 (1993), pp. 50–58.

[9] B. H. E. Vanthoor, C. Stanghellini, E. J. Van Henten, and P. H. B. De Visser. "A methodology for model-based greenhouse design: Part 1, a greenhouse climate model for a broad range of designs and climates". In: *Biosystems Engineering* 110.4 (2011), pp. 363–377.

[10] F. Tap. "Economics-based Optimal Control of Greenhouse Tomato Crop Production". PhD thesis. Wageningen University and Research, 2000.

[11] N. Bennis, J. Duplaix, G. Enéa, M. Haloua, and H. Youlal. "Greenhouse climate modelling and robust control". In: *Computers and Electronics in Agriculture* 61.2 (2008), pp. 96–107.

[12] G. Van Straten and E. J. Van Henten. "Optimal greenhouse cultivation control: survey and perspectives". In: *IFAC Proceedings Volumes* 43.26 (2010), pp. 18–33.

[13] C. H. M. Van Bavel, T. Takakura, and G. P. A. Bot. "Global comparison of three greenhouse climate models". In: *Symposium Greenhouse Climate and its Control 174*. 1985, pp. 21–34.

[14] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *Proceedings of the national academy of sciences* 113.15 (2016), pp. 3932–3937.

[15] S. L. Brunton, J. L. Proctor, and J. N. Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *Proceedings of the National Academy of Sciences* 113.15 (2016), pp. 3932–3937.

[16] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. "Data-driven discovery of coordinates and governing equations". In: *Proceedings of the National Academy of Sciences* 116.45 (2019), pp. 22445–22451.

[17] K. Champion, P. Zheng, A. Y. Aravkin, S. L. Brunton, and J. N. Kutz. "A unified sparse optimization framework to learn parsimonious physics-informed models from data". In: *IEEE Access* 8 (2020), pp. 169259–169271.

[18] B. M. De Silva, K. Champion, M. Quade, J. C. Loiseau, J. N. Kutz, and S. L. Brunton. "PySINDy: A Python package for the sparse identification of nonlinear dynamics from data". In: *arXiv preprint arXiv:2004.08424* (2020).

[19] A. Burkov. *The Hundred-Page Machine Learning Book*. Vol. 1. Andriy Burkov, Quebec City, 2019.

[20] A. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, 2022.

[21] S. Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[22] R. S. Janssen. "Development of Surrogate Predictive Model of Combined Cycle Power Plant via Physics Guided Neural Network". PhD thesis. University of Groningen, 2020.

[23] S. M. Al-Selwi, R. Muthusamy, S. A. R. Zaidi, B. Ahamed, and H. Alyami. "RNN-LSTM: From applications to modeling techniques and beyond—Systematic review". In: *Journal of King Saud University — Computer and Information Sciences* 36 (2024), p. 102068. DOI: 10.1016/j.jksuci.2024.102068.

[24] S. Hochreiter and J. Schmidhuber. "Long short-term memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[25] X. Glorot and Y. Bengio. "Understanding the Difficulty of Training Deep Feedforward Neural Networks". In: *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. Vol. 9. JMLR W&CP. 2010, pp. 249–256.

[26] K. He, X. Zhang, S. Ren, and J. Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1026–1034.

[27] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. "TensorFlow: A System for Large-Scale Machine Learning". In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)* (2016), pp. 265–283.

[28] K. Weiss, T. M. Khoshgoftaar, and D. Wang. "A survey of transfer learning". In: *Journal of Big Data* 3.1 (2016), pp. 1–40. DOI: 10.1186/s40537-016-0043-6.

[29] S. Bozinovski. "Reminder of the First Paper on Transfer Learning in Neural Networks, 1976". In: *Informatica* 44.3 (2020), pp. 291–302. DOI: 10.31449/inf.v44i3.2828.

[30] S. J. Pan and Q. Yang. "A survey on transfer learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359.

[31] J. Howard and S. Ruder. "Universal Language Model Fine-tuning for Text Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 2018, pp. 328–339.

[32] E. F. Camacho, D. R. Ramírez, D. Limón, D. M. De La Peña, and T. Álamo. "Model predictive control techniques for hybrid systems". In: *Annual Reviews in Control* 34.1 (2010), pp. 21–31.

[33] R. Amrit, J. B. Rawlings, and D. Angeli. "Economic optimization using model predictive control with a terminal cost". In: *Annual Reviews in Control* 35.2 (2011), pp. 178–186.

[34] M. Behrendt. *File:MPC scheme basic.svg*. Oct. 2009. URL: https://commons.wikimedia.org/wiki/File:MPC_scheme_basic.svg.

[35] L. D. R. Beal, D. C. Hill, R. A. Martin, and J. D. Hedengren. "GEKKO Optimization Suite". In: *Processes* 6.8 (2018), p. 106. DOI: 10.3390/pr6080106.

[36] P. van Weel, P. Geelen, and J. Voogt. *Plant Empowerment: The Basic Principles*. plantempowerment.academy, 2018. ISBN: 978-90-829035-0-8.

[37] J. Svensen, S. Petersen, and H. Nielsen. "Model Predictive Control for Energy-Efficient Climate Regulation in Mediterranean Greenhouses". In: *Biosystems Engineering* 195 (2020), pp. 45–57. DOI: 10.1016/j.biosystemseng.2020.01.003.

Glossary

List of Acronyms

ANNs Articfical Neural Networks
LSTM Long Short-Term Memory
MAE Maximum Absolute Error
MIMO Multi-Input Multi-Output
MPC Model Predictive Control
MSE Mean Squared Error
ODEs Ordinary Differential Equations
RNNs Recurrent Neural Networks
RH Relative Humidity
RTR Radiation–Temperature Ratio
RMSE Root Mean Squared Error
SINDy Sparse Identification of Nonlinear Dynamics
STLSQ Sequential Thresholded Least Squares

List of Symbols

Abbreviations

α Pipe–air heat transfer coefficient
β Absorption coefficient for solar energy
$\chi, \psi, \zeta, \xi, \sigma$ Empirical coefficients for ventilation/transfer
Δt Discrete time step size (s)
ϵ Cover heat resistance ($W^{-1} K$)
η Fraction of transmitted solar radiation

γ Psychrometric constant
λ Latent heat of vaporization ($\mathrm{J\,g^{-1}}$), function of T_g
$\mathrm{Vent_{lee}}$ Leeward vent position (%)
$\mathrm{Vent_{wind}}$ Windward vent position (%)
μ Stoichiometric constant for CO_2 consumption
ν, τ Empirical parameters for heat transfer
$\odot$ Element-wise (Hadamard) product
ϕ_h Heating valve position (%)
Φ_v Ventilation flux ($\mathrm{m^3\,s^{-1}}$)
ϕ_c CO_2 supply valve position (%)
ϕ_{inj} CO_2 injection flux ($\mathrm{g\,m^{-2}\,s^{-1}}$)
ϕ_{lee} Leeward vent opening (%)
ϕ_{wind} Windward vent opening (%)
$\sigma(\cdot)$ Sigmoid activation function (in LSTM)
$\tanh(\cdot)$ Hyperbolic tangent activation
$\tilde{c}_t$ LSTM candidate cell state update
A_g Greenhouse floor area ($\mathrm{m^2}$)
A_p Heating pipe surface area ($\mathrm{m^2}$)
AH_{in} Absolute humidity inside greenhouse air ($\mathrm{g\,m^{-3}}$)
AH_{out} Absolute humidity outside air ($\mathrm{g\,m^{-3}}$)
b_h, b_f, b_i, b_o, b_c LSTM bias parameters for different gates and cell states
C_g Heat capacity of greenhouse air ($\mathrm{J\,K^{-1}}$)
C_i Greenhouse air CO_2 concentration (ppm)
C_o Outside CO_2 concentration (ppm)
C_p Specific heat of water ($\mathrm{J\,kg^{-1}\,K^{-1}}$)
C_s Soil heat capacity ($\mathrm{J\,K^{-1}}$)
D_g Vapor pressure deficit (Pa)
d_{wind} Outside wind speed ($\mathrm{m\,s^{-1}}$)
E Crop transpiration ($\mathrm{g\,m^{-2}\,s^{-1}}$)
E_{screen} Energy screen position (fraction open/closed)
f_t LSTM forget gate (vector at time t)
G Global solar radiation ($\mathrm{W\,m^{-2}}$)
g Leaf conductance ($\mathrm{mol\,m^{-2}\,s^{-1}}$)
g_b Leaf boundary conductance ($\mathrm{mol\,m^{-2}\,s^{-1}}$)
h_t LSTM hidden state at time t
i_t LSTM input gate (vector at time t)
I_{glob} Global solar radiation ($\mathrm{W\,m^{-2}}$)
k_d Soil–deep soil heat transfer coefficient ($\mathrm{W\,K^{-1}}$)
k_r Cover heat transfer coefficient ($\mathrm{W\,K^{-1}}$)
k_s Soil–air heat transfer coefficient ($\mathrm{W\,K^{-1}}$)

k_v Ventilation heat transfer coefficient ($\mathrm{W\,K^{-1}}$)
l_1, l_2 Empirical coefficients for latent heat
m_1, m_2 Empirical coefficients for condensation mass flow
M_c Condensation mass flow at the cover ($\mathrm{g\,m^{-2}\,s^{-1}}$)
N_p Prediction horizon length (number of steps) in MPC
o_t LSTM output gate (vector at time t)
P Photosynthesis rate ($\mathrm{g\,m^{-2}\,s^{-1}}$)
p_g^* Saturated vapor pressure (Pa)
p_g Air vapor pressure (Pa)
Q, R MPC weighting matrices (state/output and input cost weights)
R Crop respiration ($\mathrm{g\,m^{-2}\,s^{-1}}$)
s Slope of saturated vapor pressure curve
s_1, s_2, s_3 Coefficients for s
T_c Cover temperature (°C)
T_d Deep soil temperature (°C)
T_g Greenhouse air temperature (°C)
T_h Heating water supply temperature (°C)
T_o Outside air temperature (°C)
T_p Heating pipe temperature (°C)
T_s Soil temperature (°C)
T_{air} Greenhouse air temperature (main state) (°C)
T_{out} Outside air temperature (°C)
T_{pipe} Heating pipe temperature (as input, °C)
u Control/actuator vector
v Exogenous input/disturbance vector
V_g Greenhouse air volume ($\mathrm{m^3}$)
V_i Absolute humidity of greenhouse air ($\mathrm{g\,m^{-3}}$)
V_o Outside absolute humidity ($\mathrm{g\,m^{-3}}$)
V_p Heating pipe volume ($\mathrm{m^3}$)
w Outside wind speed ($\mathrm{m\,s^{-1}}$)
W_c^* Saturated humidity ratio at cover temperature
W_g Humidity ratio of greenhouse air
$W_{hh}, W_{xh}, W_f, W_i, W_o, W_c$ LSTM weight matrices
x_g Greenhouse state vector