energies

MDPI

Article

A Physics-Based Modelling and Control of Greenhouse System Air Temperature Aided by IoT Technology

Beatrice Faniyi and Zhenhua Luo *

School of Water, Energy, Environment and Agrifood, Cranfield University, Cranfield MK43 0AL, UK
* Correspondence: z.luo@cranfield.ac.uk

Abstract: The need to reduce energy consumption in greenhouse production has grown. Thermal heating demand alone accounts for 80% of conventional greenhouse energy consumption; this significantly reduces production profit. Since microclimate affects crop metabolic processes and output, it is essential to monitor and control it to achieve both quantity and quality production with minimum energy consumption for maximum profit. The Internet of Things (IoT) is an evolving technology for monitoring and controlling environments that have recently been adopted to boost greenhouse efficiency in many applications by integrating hardware and software solutions; therefore, its adoption is thus critical in enabling greenhouse energy consumption minimisation. The first objective of this study is to improve and validate a greenhouse dynamic air temperature model required to simulate or predict indoor temperature. To achieve the first objective, therefore, an existing model was enhanced and a closed loop test experimental data from the IoT cloud-based control system platform deployed in the prototype greenhouse built in Cranfield University was used to validate the model using an optimisation-based model fitting approach. The second goal is to control the greenhouse air temperature in simulation using relatively simple PI and on-off control strategies to maintain the grower's desired setpoint irrespective of the inevitable disturbances and to verify the potential of the controllers in minimising the total energy input to the greenhouse. For the second objective, the simulation results showed that the two controllers maintained the desired setpoint; however, the on-off strategy retained a sustainable oscillation, and the tuned PI effectively maintained the desired temperature, although the average energy used by the controllers is the same.

Keywords: energy model; greenhouse climate; energy minimisation; control algorithm; IoT control-based greenhouse

Citation: Faniyi, B.; Luo, Z. A Physics-Based Modelling and Control of Greenhouse System Air Temperature Aided by IoT Technology. *Energies* **2023**, *16*, 2708. https://doi.org/10.3390/en16062708

Academic Editors: Diego Luna and Andrea Mariscotti

Received: 27 January 2023
Revised: 6 March 2023
Accepted: 7 March 2023
Published: 14 March 2023

1. Introduction

Sustainable energy management is a global priority for smart agriculture. Sustainable energy management hinges on energy use, environmental effect, and cost efficiency in greenhouse applications. Sustainable horticulture is a challenge since a growing population requires better agricultural output yields, which increases the industry's energy requirement [1]. The greenhouse microclimate provides the optimum environment to meet global crop necessities. The greenhouse-controlled environment allows crop production in climates and seasons that would otherwise prohibit growth, aids in prolonging seasonal crop cultivation, and reduces transportation distances, water use, and land usage [1]. The environment provides good growing conditions for the plants, one of which is the inside air temperature [2]. A greenhouse can produce fresh veggies year-round with a 50% higher yield than open-air agriculture. However, energy usage and labor costs contribute to more than 50% of greenhouse production costs [3]. In addition, the commercial greenhouse has the largest energy consumption compared to all other agricultural industries. In this case, energy management in greenhouses is crucial from a broad sustainability standpoint. In warmer regions, the cost of heating (in particular) and cooling a greenhouse can account for approximately 50% of its total operating costs. The cost can account for 70–85% of total

operating expenses in northern latitudes [4]. Approximately 65–85% of the total energy used in greenhouses is for heating, while the remaining is used for transportation and electricity [5]. So, it is clear that heating costs take up a big chunk of the greenhouse's operating budget. Therefore, any steps that make it cheaper to control the temperature inside will reduce the amount of energy used and increase profit. Temperature is a major determinant of crop development, yield, and quality. In addition to the economic benefit associated with proper management of greenhouse heating demand, it is important that the variable be regulated [6].

Over the years, farmers as well as researchers have tried several methods to reduce greenhouse energy consumption, including the use of control algorithms, energy modeling, the control of key environmental parameters, the use of technologies such as information and communication technology (ICT), and other prevailing methods such as the use of a variety of energy storage equipment [7]. Several systems have been used in closed and semi-closed greenhouses to control the climate. These closed and semi-closed greenhouse systems store heat in aquifers, buffers, heat pumps, and heat exchangers. However, controlling these systems is difficult because of, e.g., the different types of energy, complex energy facilities, etc. [6]. Precision agriculture (PA) has been identified as a viable solution to the problem. PA relies on automated sensing and control solutions to measure and control internal parameters to boost efficiency by limiting environmental effects, permitting costs, water savings, etc. [8]. The Internet of Things (IoT) is a fast-growing technology that has recently been used to automate and remotely control the environmental parameters in greenhouses by using hardware resources (sensors, controllers, and actuators) and control algorithms.

This work is an attempt to address two main gaps in the literature. First, although IoT hardware resources, as mentioned above, have been implemented for optimisation in various greenhouse applications up until now, such as optimal control of soil nutrient levels in hydroponics [9], intelligent lighting control systems [10,11], the optimisation of greenhouse irrigation systems [12,13], and the monitoring of crop-sensitive data, that gives the farmer real-time performance information to make informed decisions [14]. Despite these application areas, the use of IoT hardware resources to reduce total heating energy demand in greenhouses is uncommon in the literature; only a few works address the subject. However, greenhouse energy models are not used to implement the control system solution, as evidenced by these works [15,16]. Because of this, it is hard to determine if the total energy used is reduced in these applications since all the fluxes that affected the greenhouse climate were not accounted for in simulation and control system design.

Second, despite the availability of proprietary control systems for indoor agriculture, growers with small- to medium-sized greenhouses or indoor growth facilities need affordable solutions. The increased attention to the use of advanced control algorithms, such as model predictive control (MPC), for the optimisation of greenhouse energy use is noted in the literature [1,17]. And it has been determined that there is a need for a straightforward, inexpensive control algorithm that can be used by the average farmer. ON/OFF control is a simple and affordable temperature control strategy that is rarely considered for reducing energy use in greenhouses. To assess its suitability for minimising energy consumption, this paper compared its performance to that of the PI control algorithm.

Consequently, the objective of this study was to meets these two needs by modifying a physics-based dynamic air temperature model that included an energy minimisation module to simulate a greenhouse with IoT hardware resources for indoor temperature prediction and to test how well two simple controllers maintained the grower's desired indoor temperature while saving energy via simulation. In this context, the primary objective of this project was to modify and evaluate a dynamic greenhouse air temperature model. The model was validated using data from the experimental prototype greenhouse with an IoT-based control system. The second objective was to evaluate how well two control algorithms (a PI controller and an ON/OFF controller) maintained the setpoint while minimising the total input to the greenhouse.

Criterion for Model Selection

When energy balance equations are used to construct the differential equation of an energy model, the approach is termed "physics-based" or "first principles" modelling. The use of a physics-based air temperature dynamic model of a greenhouse is essential to describing the climate inside the prototype greenhouse and the functionality of the control equipment. This is important because to control the greenhouse a model-based control design is necessary to understand the behaviour of the system and for prediction for control purposes [18]. A dynamic energy simulations model helps to simulate the temperature dynamics, and facilitate understanding of the complex mechanisms involved in the thermal process of greenhouse operation and thus contribute to a more energy-efficient greenhouse operation.

Although there are studies on optimal control strategies that adopt an economic criterion in the simulation model to either minimise both heating and ventilation expenses [19] or maximise profit, as seen in these works [20,21]. However, the model that directly addresses the minimisation of the total input energy use is more applicable for this work without the inclusion of economic criteria. This approach was implemented in [7,22]. They targeted the minimisation of the total energy input, so they are suitable physics-based models for this study; however, the latter used a steady-state model, which is not fitting for our work. Aside from the selection of an acceptable model, the frequent use of advanced control strategies and the necessity to balance the link between inexpensive and high-accuracy control systems were identified in the study [17]. Therefore, our current research aims to verify the potential of two relatively simple to implement algorithms in maintaining the grower's desired indoor temperature, as well as the total energy-saving potentials of the two algorithms using an appropriate energy model. The remainder of the paper is as follows: in Section 2, the model and experimental setup were described. In Section 3, we used the optimisation method to simulate and validate the model. In Section 4, we presented and analysed the results and drew a conclusion based on the control performance and energy consumption of the two controllers.

2. Methodology

As stated previously, the physics-based temperature model based on greenhouse climate models described in [7,23,24] serves as the basis for this work. The following changes were made to these models: the model described in [7] has been modified so that it can be easily applied to simulate the indoor temperature of different greenhouse setups that are equipped with IoT hardware resources and used to build model-based controllers that can maintain the desired temperature of different crops. The original differential equation for air temperature is given below [7]:

$$\frac{dTin}{dt} = \frac{1}{Ccap}\left(Q_{sun} - Q_{cover} - Q_{trans} + Q_{lamps} + Q_{energy}\right) \quad (1)$$

where the Q_{energy} is described as:

$$Q_{energy} = \left(-Q_{vent} \pm Q_{he} + Q_{pipe}\right) \quad (2)$$

However, we removed the effect of (a) the heat transfer through crop transpiration Q_{trans} from Equation (1) and replaced it with $Q_{soil} = K_{coe}(T_s - Tin)A_g$ [24], which accounts for the heat transfer between the internal air and soil as shown in Equation (3) below. (b) The effect of the heat imposed on the greenhouse through solar radiation Q_{sun} and heat loss through the cover Q_{cov} were both replaced with the equations in [23]. (c) The effect of the heat gain due to artificial lighting Q_{lamps} in Equation (1) was ignored in this work. (d) In Equation (2), from the energy input term Q_{energy}, the heat exchange with outdoor air due to the natural ventilation Q_{vent} from [7] was used; however, the effect of heating by the pipe rail heating system Q_{pipe} was ignored. In the model simulation, the effect of active heating or cooling by the heat exchangers, $\pm Q_{he}$ was replaced with $+Q_{power}$ as

shown in Equation (3) below. As a result, the heating and vent were combined into a single manipulating input variable Q_{energy}, implying that a negative energy flux removes heat from the greenhouse air (cooling), whereas a positive energy flux adds heat to the greenhouse air (heating). The air temperature model that resulted is:

$$\frac{dTin}{dt} = \frac{1}{Ccap}\left(Q_{sun} - Q_{cov} - Q_{soil} + Q_{energy}\right) \tag{3}$$

For simulation reasons, we replaced Q_{energy} in Equation (3) with Equation (4) below:

$$Q_{energy} = \left(-Q_{vent} + Q_{power}\right) \tag{4}$$

where $Ccap$ represents the heat capacity of the greenhouse air. The model for the specific heat capacity of the greenhouse air is adopted from [24], given as:

$$Ccap = R_{hoa} * C_a * V_g \tag{5}$$

The models in Equations (6)–(9) below represent the individual fluxes that affect greenhouse air temperature (note that the descriptions of all the variables and parameters are given in Nomenclature), as described in Equation (3). They are expressed in detail below as: Q_{sun}, which represents the heat load imposed on the greenhouse by the sun, is computed using the solar radiation model derived in [23], which was simplified by [24] in Equation (6) as:

$$Q_{sun} = C_{rad} * Q_{rad} * A_g \tag{6}$$

The convective heat loss via the cover Q_{cov} is described below as [24]:

$$Q_{cov} = Q_{coe} * (T_{out} - T_{in}) * A_c \tag{7}$$

The heat transfer between the internal air and soil is described as [24]:

$$Q_{soil} = K_{coe} * (T_s - T_{in}) * A_g \tag{8}$$

Lastly, for simulation purposes, since Q_{vent} is the heat lost through natural ventilation, it is realised in the experiment through the opening of the roof window and computed using the model below [7] as:

$$Q_{vent} = Q_{ven} * R_{hoa} * C_a * (T_{out} - T_{in}) \tag{9}$$

A simulation algorithm developed in Python was used to numerically solve Equation (3). The result and validation procedure are presented in Section 3. Few assumptions were made in the system model simulation. We assumed that: (1) the greenhouse air is homogeneous, i.e., it was counted as a perfectly stirred tank; and (2) the floor area is assumed to have been directly planted on, instead of the pots that were used. (3) Furthermore, the feedback effects of crop growth on air temperature were ignored.

2.1. Experimental Set-Up

The 4-by-6-foot prototype greenhouse shown in Figure 1 below is located at Cranfield University, United Kingdom (52° N, 0° E). The greenhouse has a floor area of 12.5 m^2. The roof consisted of a single aluminium frame vent mounted at a slope of 60°. These ventilation windows measured 0.6 × 0.6 m^2 and had a maximum opening angle of 60°. An experiment was set up in May 2022 to measure the temperature dynamics in the greenhouse air and obtain data for the calibration and validation of the improved air temperature model in Equation (3).

Figure 1. Experimental greenhouse.

Although the indoor data that was used for the validation was from this greenhouse system, the exogenous data were obtained from the Cranfield weather station as shown in Figure 2.

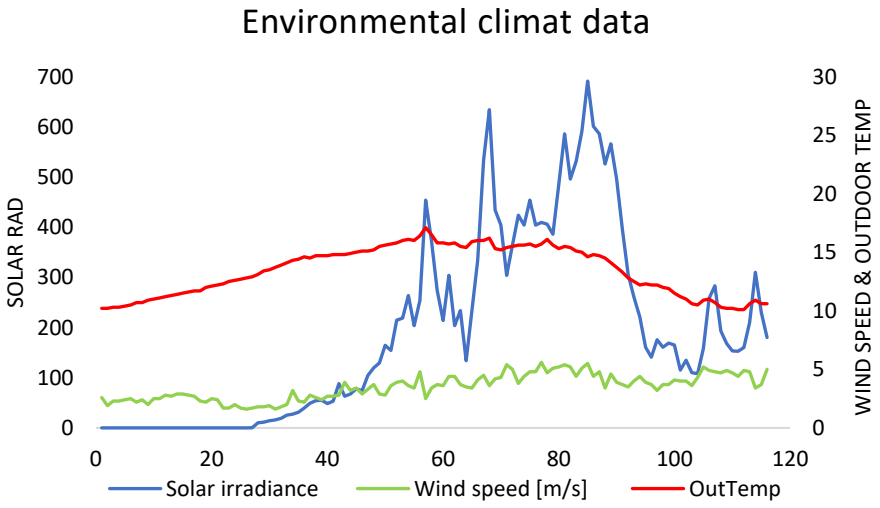


Figure 2. Solar irradiance (left scale), outdoor temperature, and wind speed (right scale).

The greenhouse is equipped with a 2 kW heater (placed in the upper middle green rack), artificial lighting, natural ventilation, and three green planting racks consisting of two 50 W bulbs and one 25 W bulb. These heating bulbs were used as an additional IoT-supported heating system to maintain the desired microclimate temperature around the plant and were thus placed in the lower racks on the greenhouse floor. However, the greenhouse is simulated as a single input, single output (SISO) system. Only the centre sensor temperature reading was used as the output variable measurement of the system as shown in the control system schematic diagram in Figure 3 below.

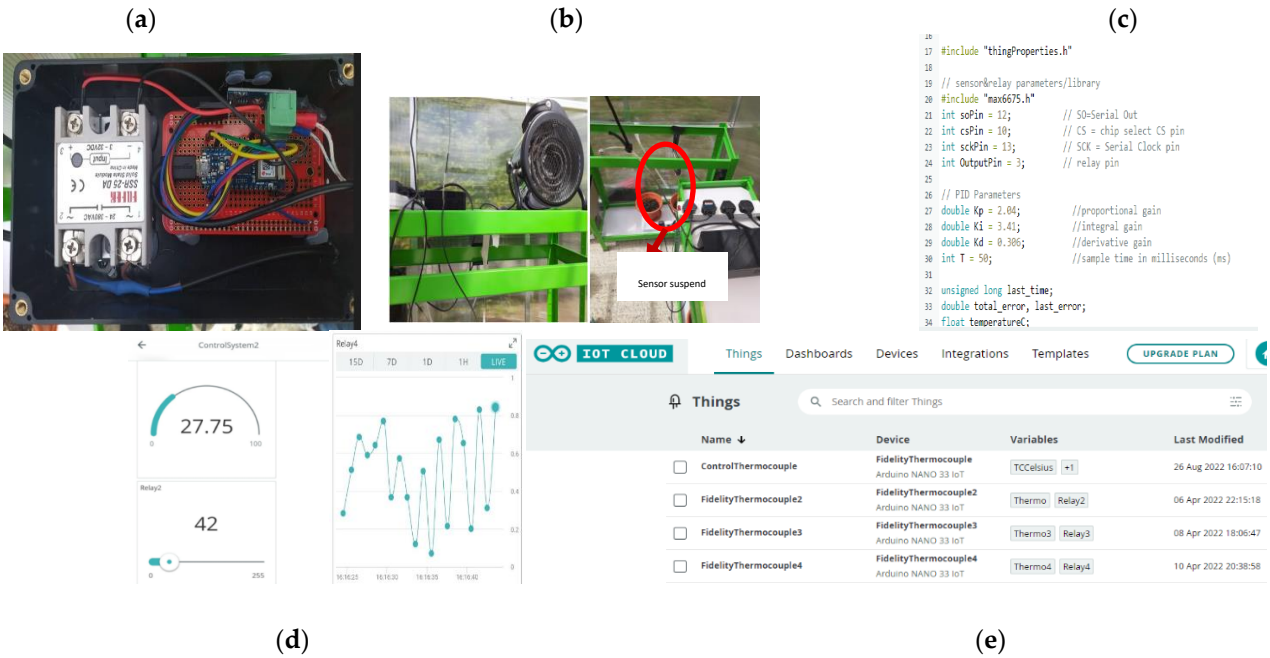


Figure 3. From the left, (**a**–**e**) is one of the IoT control system content; the interior view of the greenhouse showing the 2 KW, a suspended sensor in the air and planting pots; Arduino IoT cloud customisable software; user interface mobile app; and web-based app.

2.2. IoT System Architecture

The simple and efficient IoT cloud-based control system is shown in the schematic diagram in Figure 4, which consists of a sensing part, a controlling part, and an actuating part (four of the IoT cloud-based control systems in total were deployed in the greenhouse, one in the upper middle green rack and three others placed in the lower racks as previously described). The main units of the established IoT cloud-based control system are as follows:

- Arduino Nano 33 IoT microcontroller;
- K-type thermocouple sensor connected to the microcontroller;
- A Wi-Fi connection;
- Heater and electric bulbs;
- Arduino IoT cloud platform;
- User interface.

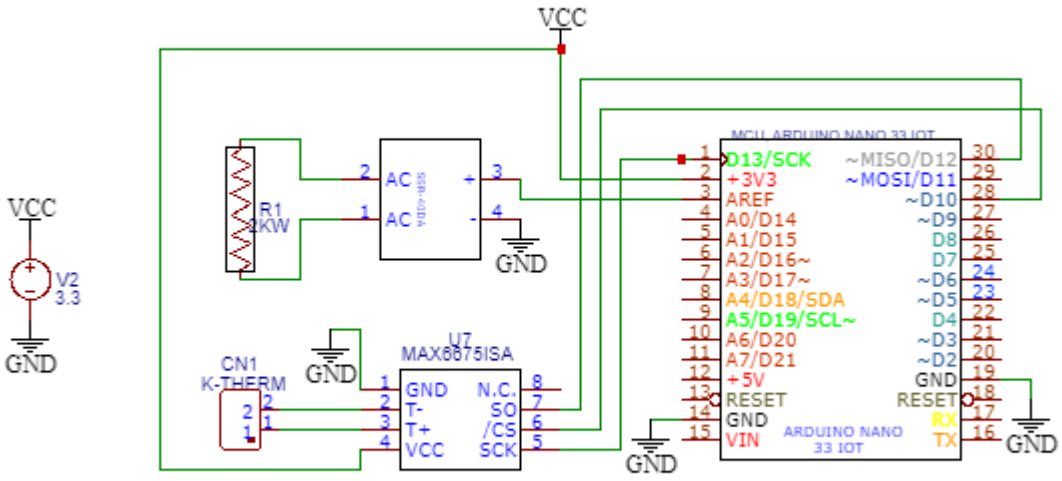


Figure 4. IoT-based control system schematic diagram.

Four of the IoT-based control systems were deployed in the greenhouse. Each of them starts operation by activating the sensor attached to the Arduino 33 IoT microcontroller. The K-type thermocouple sensor is positioned to measure the air (microclimate) temperature. The sensor reports its collected data to the microcontroller in real-time. Consequently, the microcontroller directly transmits the collected data to the Arduino IoT cloud platform through the phone that was turned into a Wi-Fi hotspot. The user gets access to the real-time data at any time of the day or in real-time, either through the Arduino IoT Cloud platform via the mobile app Figure 3d or through the computer web-based user interface as shown in Figure 3e.

Regarding the real-time decisions, we wrote the code partly shown in Figure 3c to read data from the sensor and to control the heater in real-time. This code was written using Arduino IDE and modified using the Arduino Create web editor, and we uploaded the sketch directly to the Arduino microcontroller from the web browser. The real-time decisions are then made by the microcontroller based on the software that is running on the Arduino microcontroller.

2.3. Schematic Diagram Design

The control system schematic diagram was designed using EasyEDA online design tool. EasyEDA is a web-based circuit design, circuit simulator, and PCB design tool that is free and simple to use. The IoT-based control system schematic diagram was designed using this tool and presented in Figure 4.

2.4. Hardware Design

We chose the Arduino Nano 33 IoT microcontroller board to manage the sensor and actuator that were placed in the greenhouse. The Nano 33 IoT board is a low-power microcontroller with integrated Wi-Fi, Bluetooth, an inertial measurement unit (IMU), and a real-time clock (RTC). It has the same pins and form factor as the previous 8-bit Nano board and consists of 22 digital input and output pins, including 8 analogue input pins and 14 digital pins, 11 of which can be used for PWM output. The board has one 5 V, one 3.3 V, and two ground pins. As shown in Figure 3a, to connect the sensor and actuator (relay) to the Arduino microcontroller, we used a total of 4 pins. This provides an opportunity to extend the scale of the control system in the future.

MAX6675 was used in the design of the control system. MAX6675 is an analog-to-digital converter that converts the K-type thermocouple's voltage output into a digital signal that the Arduino Nano 33 IoT board can read. The Arduino Nano 33 IoT board is compatible with the MAX6675, which operates at 3.3 V and has a maximum output current of 50 mA, well within the maximum output current of the Arduino Nano 33 IoT board.

We used a 40 A SSR-40DA relay. This solid-state relay can switch up to 40 A of 24–380 V AC loads. Although the maximum current that can be sourced by a single microcontroller pin is significantly less than 200 mA, the SSR is designed to operate directly from logic-level outputs. This simplifies the design, reduces the number of components, and eliminates the need for additional circuitry, making it ideal for switching power in industrial and household appliances. Nonetheless, if available, a simple transistor or optocoupler circuit will serve as a buffer between the microcontroller and the device.

The 2-kW heater used in the IoT-based control system operates at 240 V AC and draws approximately 8.33 A of current. This falls within the rating of the SSR-40DA 40 A relay, making it suitable for the heater.

2.5. Software Design

After developing a PID algorithm in the Arduino IDE and modifying it with the Arduino Create web editor, we uploaded the sketch to the Arduino microcontroller directly from the web. Figure 3c depicts a portion of the code that was developed in the IDE. The IoT cloud configuration procedure is described in detail below.

1. The first step involves setting up a local laptop with an Arduino Create plugin. This plugin allows the Arduino 33 IoT board to communicate with the Arduino IoT Cloud via a micro-USB connection;
2. The hardware setup in Figure 3a shows the connection of the Arduino 33 IoT board to the thermocouple sensor and the SSR;
3. The board is then configured through the Arduino Create plugin mentioned in step 1. This process involves following simple instructions to register the device in the cloud and establish communication with the cloud via Wi-Fi;
4. In the Arduino Cloud platform, the microcontrollers are referred to as "things." Figure 3e shows the four connected microcontrollers called "things." Each of these "things" has properties (a property is a cloud variable) that can be viewed and modified from the Arduino cloud platform;
5. For each of the microcontrollers, two properties are set up: TCCelcius or thermo (which represents temperature measured in degrees Celsius) and a relay (to vary the control input signal to the heater). The value of the temperature variables is set to range from 0–100 in the cloud, and the variable permission that is set is "Read Only" (this means the variable can only be read from the Arduino pin it is connected to but cannot be used to output voltage);
6. The relay variable is set to range from 0–255 in the cloud, which is the maximum output that the Arduino PWM pin is allowed to output, and the variable permission that is set is "ReadWrite" (which means that it can both receive and send data to the Arduino pin that controls the relay. This allows the Arduino Cloud platform to modify the relay's control input signal and turn the relay on and off as needed.);
7. The data updating status is set for both the TCCelcius or thermo and relay variables to update data "on change," which means that data is updated as soon as it becomes available.

Although the temperature range was set on the cloud from 0–100, the PID algorithm running on the device is already programmed to a setpoint of 23 °C that is required for the plant in our work. The control system input and output data are given in Figure 5.

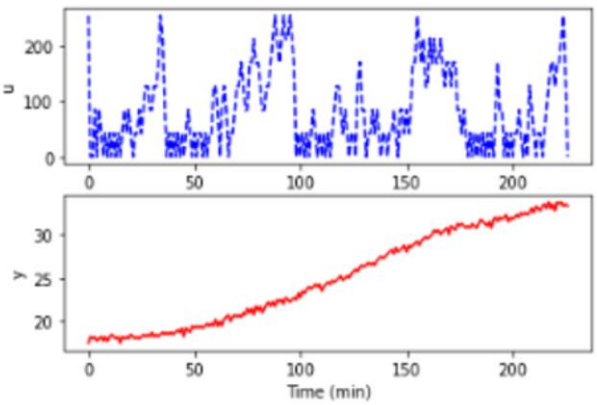


Figure 5. IoT-based control system measured input/output data.

3. Simulation and Validation

In this section, we performed a simulation to validate the improved model in Equation (3) using the IoT-based control system's (closed loop) measured indoor temperature data depicted in Figure 5. Validation entails determining if the improved model adequately represents the pilot greenhouse. In the following section, a method for validating the applicability of the proposed model is described using the energy balance model presented in Equation (3).

The work done in this section aims to estimate two model parameters in Python that can predict indoor temperature based on Equation (3). This is achieved by optimising the predicted output to match the experimental measurements of the real system shown in

Figure 5 using an optimisation technique. The following steps were followed to formulate the parameter estimation problem as an optimisation problem. The least squares method was used, and the objective function to be minimised is the sum of squared prediction errors given below.

$$SSE = \sum_{k=1}^{N} e(k)^2 \tag{10}$$

where $e(k)$ represents the prediction error, which is the difference between the observed temperature $Tin_{obs}(k)$ and the predicted indoor temperature (Tin_{pred}), given as:

$$e(k) = Tin_{obs}(k) - Tin_{pred}(k) \tag{11}$$

where Tin_{obs} is observed temperature values which are taken from Figure 5 and Tin_{pred} is the predicted indoor temperature that was calculated through simulations. The goal is to optimize the prediction errors and make the predicted output as similar as possible to the experimental measurements. The optimization problem was solved using the Python library SciPy, which provides a function called "curve_fit" that implements the least squares method. The "curve_fit" function requires as input the model function (Equation (3)), the experimental data (indoor temperature measurements), and an initial guess for the parameters. The output of the "curve_fit" function is the estimated parameters that optimize the objective function (i.e., the sum of squared prediction errors).

The estimated model parameters were used to simulate the Indoor temperature of the greenhouse over a period of one day (using the 25 May 2022 data). The simulation results were compared to the experimental measurements to assess the accuracy of the model. The comparison showed that the model with the estimated parameters was able to predict the indoor temperature of the greenhouse with reasonable accuracy. This confirmed the validity of the physics-based dynamic model and the effectiveness of the parameter estimation technique used in this work.

Overall, the work in this section demonstrates the importance of parameter estimation in model-based control of indoor temperature in greenhouses. The optimizationn technique used in this work can be applied to other models and experimental data to estimate model parameters and improve the accuracy of the model predictions.

3.1. Simulation Conditions

The model in Equation (3) was solved numerically in Python; we used a static environmental data of the solar radiation [W/m^2], the wind speed [m/s], and outdoor temperature [°C] measured on 25 May 2022 shown in Figure 2 (obtained from the Cranfield weather station as previously discussed) as inputs to the model.

The known thermophysical properties of the greenhouse cover, soil, vent and heater parameters are used as input parameters to the model. Values shown below in Table 1 are the other required parameter values that were obtained from the literature.

Table 1. Parameters adopted in the literature [24].

Parameters	Values
Crad	0.1
C_a	1000
Rhoa	1.29

Since the dynamic model for the greenhouse air temperature is modified from an existing energy balance equation as previously explained, assuming the greenhouse is a perfectly stirred tank (homogeneous condition) and that the model is a single differential equation with a separate energy input module in Equation (4). We modified the model so that it can predict the air temperature accurately. Two unknown model parameters, which

are the total ventilation flux from indoor to outdoor air Q_{ven} in Equation (9), and the soil surface temperature, T_s, in Equation (8) are estimated, and the values are shown in Table 2.

Table 2. Optimised model parameters.

Estimated Parameters	Identified Value
Q_{ven}	0.29
T_s	11.05

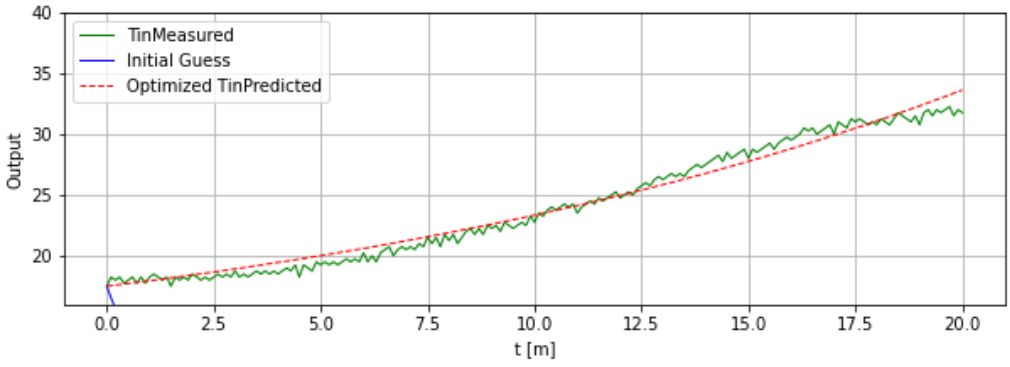
As previously discussed, the least square method is applied to estimate these parameters using the dynamic model; in this method the unknown parameters are adjusted until the difference between the measured and the simulated temperature is minimised in the least square sense. This method gives good estimates after several iterations with an initial guess ($T_s = 20.58$ and $Q_{ven} = -5.88$). The evaluated model from the experimental data with a minimum sum of squared errors (SSE) is realised.

3.2. Validation Results

Figure 6 shows the results of the simulation for the validation of the model in Equation (3). In the time interval 0.0–20.0 min, the simulated indoor air temperature based on the estimated model is plotted together with the real measured indoor temperature. The simulation runs with the initial state value set equal to the real measured state at t = 0.0. The highest discrepancy is found within 12.5–17.5 min between the simulated indoor air temperature and the actual measured temperature during this period, which is roughly 2 °C. The modified physics-based model can correctly estimate the inside temperature. The initial SSE objective in the values is Figure 6a below and the final SSE objective after optimisation is depicted. The optimization-based model adaption strategy for simulating greenhouse air temperature offers promising results: it produces a model that fits well and yields predictions that are approximately equal to the values of the measured air temperature data.

```
Initial SSE Objective: 39807.692303073345
Final SSE Objective: 91.1413391989764
Ts: 11.05362082839811
Qven: 0.2949219722610455
```

(**a**)



(**b**)

Figure 6. (**a**) Optimisation result; (**b**) measured versus simulated temperature [°C].

The optimised model was used to calculate the required static operating point ($Q_{energy}0$, $T_{in}0$). We calculated the operating point from the model to determine the required static control input energy ($Q_{energy}0$) needed to bring the indoor temperature to a static (constant value) response of 23 °C in Python by setting the obtained air temperature dynamics equation to zero. This simulation was intended to test whether the model can be used in

varying weather conditions, as shown in Figure 7 below. The operating point was calculated using one day's data; three simulations were done in all. The input disturbances (outdoor temperature and solar irradiation) were adjusted to 10.5 °C and 108.2 W/m^2, 16.0 °C and 333.0 W/m^2 and finally 22.0 °C and 400.0 W/m^2, respectively, for 7000 s. Figure 7a–c below show what is required to maintain the reference indoor value under different circumstances. Results show that even if the outdoor conditions change, provided there is a correct actuator (either a heating or cooling system), the desired indoor temperature can be maintained.

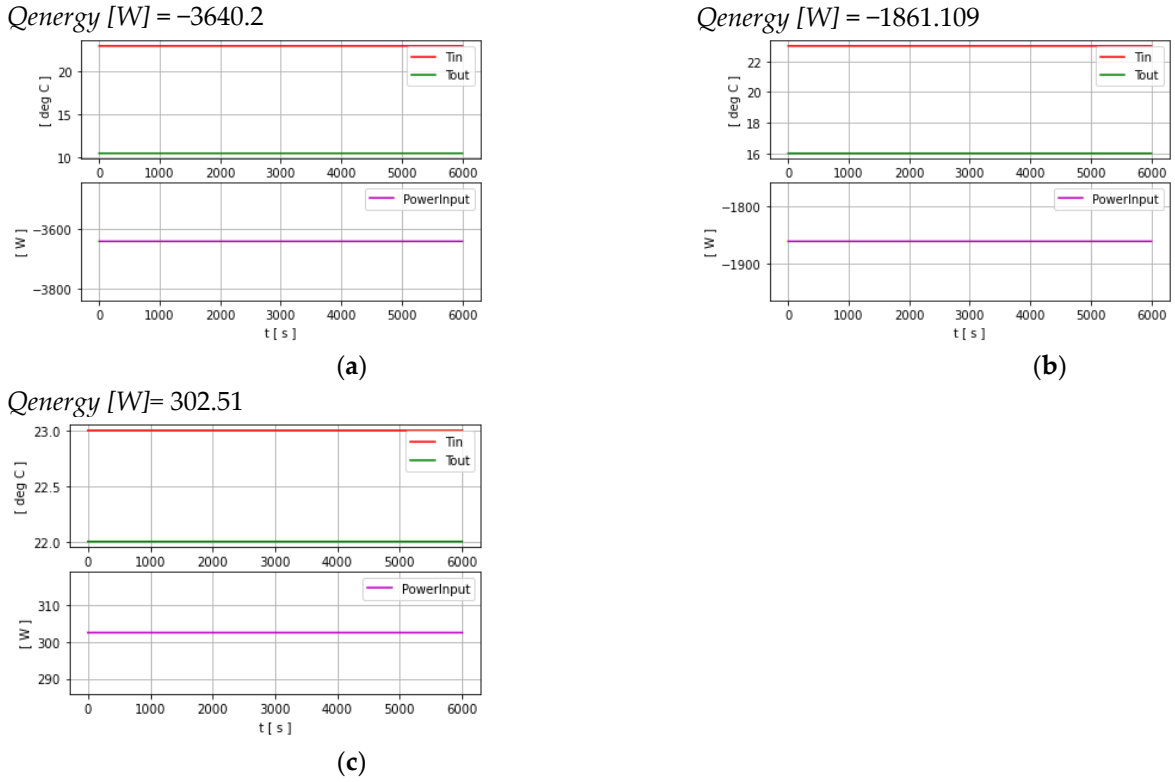


Figure 7. (**a**–**c**) Simulated results.

4. Controller Design

The dynamic behaviour of a process defines how the measured process variable changes over time as a result of the controller output and any disturbance factors; understanding process dynamics is critical when designing and tuning a controller. The controller output in our experiment is a pulse-width modulated (PWM) signal from the Arduino microcontroller.

This signal varies between 0 and 255, and is used to control the state of a relay that operates the actuator (heating system). Depending on the value of the PWM signal, the relay may switch on or off, thereby affecting the process variable. However, due to external disturbance factors (Figure 2) that highly influenced the measured air temperature, we needed to calculate the required static input energy to enforce the indoor temperature to the desired static (steady state) temperature value of 23 °C using the system model.

To design a PI control system, we needed to estimate the dynamic properties of the process. For this requirement, the calculated steady-state operating point in Figure 7a was used in the simulation of a Doublet test to confirm how the indoor air temperature responds to changes in the input signal and disturbances, as shown in Figure 8. Generally, in a Doublet test, a system is allowed to stabilize at some step input signal and corresponding temperature. The equilibrium is then disrupted either by raising or lowering the step input signal, and the resulting dynamic response time is measured [25].

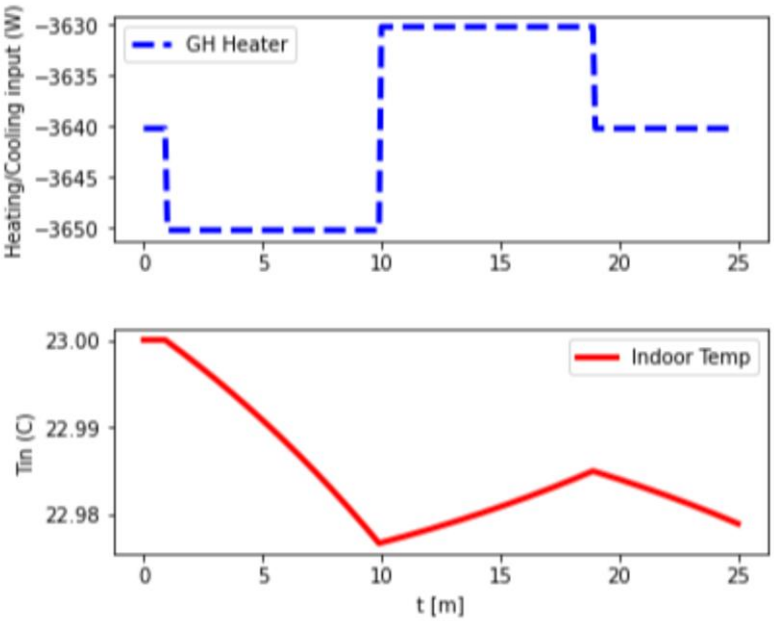


Figure 8. Simulated process response to Doublet test.

With the method described, we were able to get the dynamics process data for the design of the control system, as shown in Figure 8 above. By fitting a first-order plus dead-time (FOPDT) model to the dynamic process response data, we estimated the key properties of the process dynamics. The FOPDT model parameters were used in Section 4.5 to design the tuned controller and are shown in Table 3. Although we designed two controllers, we evaluated the results of three controllers in the simulation, resulting in a total of three controllers described in this work: an ON/OFF control, a PI controller with gains determined by trial and error, and a tuned PI controller design. We provide a brief discussion of the two-control method below, along with an evaluation of their performance.

Table 3. FOPDT process parameters.

Parameters	Values
K_p	0.04
τ_p	151.9
θ_p	11.8

4.1. ON/OFF Controller

An ON/OFF controller is a simple control based on two rules, and it may be used as a substitute for a PID controller, particularly in temperature control. A thermostat is a typical example of an ON/OFF controller that is usually used in room temperature control. The ON/OFF controller function is given in [26]. We set the control signal input value to vary between a maximum of 3640.2 when in heating mode and −3640.2 (W) minimum when in cooling mode, as calculated from the static test that required energy input. The function is given as:

$$u_{(t)} = \begin{cases} u_{max} & if\ e \geq 0 \\ u_{min} & if\ e < 0 \end{cases} \tag{12}$$

where

$$e = r(t) - y(t) \tag{13}$$

$e(t)$ is the control error, $r(t)$ is the setpoint or reference signal and $y(t)$ is the process output measurement, u_{max} is the maximum control signal, usually 100%, and u_{min} is usually 0%. However, u_{max} is set as 3640.2 and since we used u_{min} as cooling in our simulation, it was

set as −3640.2, The output of the control algorithm is the power supply set as the Q_{energy} for the simulation of the control system.

$$Q_{energy} = Qenergy_{haeting} = 3640.2 \text{ or } Qenergy_{cooling} = -3640.2 \tag{14}$$

4.2. PI Controller

According to the authors in [27], a traditional PI controller accelerates response. However, PI controllers require tuning of the proportional gain and integral time to set their appropriate value based on the thermophysics of the particular heating equipment [28], in this research we used the trial-and-error method to obtain the first PI control output and later the estimated process constants from the FOPDT model fitting were used in a correlation to obtain initial estimates of the second PI control tuning parameters. The standard PI controller function is given in this form:

$$u(t) = K_c e(t) + \frac{K_c}{T_i} \int_0^t e(t) d\theta \tag{15}$$

where K_c is the proportional gain, T_i is the integral time constant and $e(t)$ is the control error. For the PI controller, to avoid the windup generated by the interplay of integral action and saturations, we limit the control law variations so that the controller output never exceeds the actuator limit u_i^{lim} given as follows [29]:

$$u_i = \begin{cases} u_i^{lim} & \text{if } |u_{min}| \geq u_i^{lim} \\ u_{min} & \text{otherwise} \end{cases} \tag{16}$$

4.3. Controls Simulation Result and Discussion

The controller design simulation was executed in Python for 1440 min each; these were intended to create a 1-day test from 6 a.m. to 6 a.m. of the next day. For the trial-and-error controller tuning, the parameters were given as; K_c set to 80 (W/K) and the T_i value was set to 5.9 (s). We used the trial-and-error method to obtain these values. However, these controller parameters can be changed to different values. The trial-and-error PI controller's parameters are shown in Table 4 below.

Table 4. PI tuning trial and error values.

Parameters	Values
K_c	80
T_i	5.9

The work aims to evaluate the performance of the controllers in disturbance rejection and to determine the control algorithm that minimises the total input energy (heat). To achieve this, both controllers were given the same parameter values as shown in Table 5. The static value of solar radiation Q_{rad}, outdoor temperature T_{out}, the desired temperature set point T_{in_sp} and initial temperature $T_{in\_initial}$ were used in the simulation. The control energy input (heat) was limited to the derived static control input energy of 3640.2 (W) *u_max* and *u_min* = −3640.2 as the actuator limit (given in Equation (17)). The Q_{rad} and T_{out} values were obtained from the outdoor disturbance data.

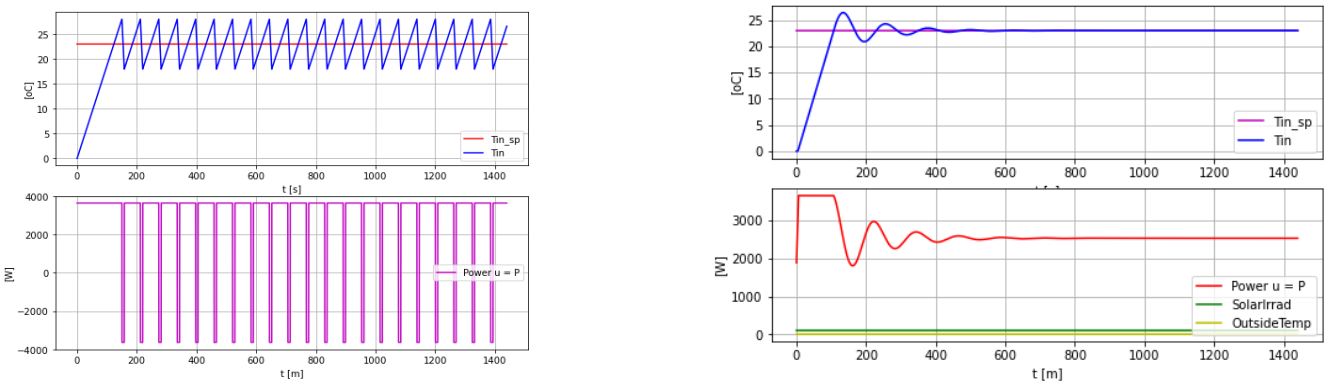
Table 5. Trial and error PI and ON/OFF control parameters.

Parameters	Values
Q_{rad}	108.2
T_{out}	10.5
T_{in_sp}	23
$T_{in\_initial}$	0 °C

The remaining values used in the simulation for the ON/OFF and PI controller were given below in Table 5.

4.4. Controller Results

Figure 9a,b depicts the result of the ON/OFF and PI controller, with a static input disturbances value.



(**a**) (**b**)

Figure 9. (**a**) ON/OFF control output; (**b**) PI control output.

Figure 9a,b above, shows the outcome of the ON/OFF controller and the PI controller, respectively, during a 24-h simulation period. The ON/OFF control is based on a simple rule that turns the heating system on and off based on the difference between the desired temperature and the actual temperature. As depicted in Figure 9a, the ON/OFF controller was employed to regulate the indoor temperature. Although both controllers achieved the targeted indoor temperature, the ON/OFF controller had a non-zero control error. On the other hand, the PI controller, which was tuned through a trial-and-error approach, resulted in oscillatory and marginally delayed closed-loop responses during the transient phase, which eventually yielded stable results. To compare the energy-saving potential of the two controllers, Figure 9a,b shows that the average control input (heat) required to regulate the temperature for the two controllers is nearly the same. However, the PI controller shows superior performance in terms of control error and oscillations. In the next section, we discuss the simulation procedure for tuning the PI controller to achieve better control performance in comparison to the two controllers designed above.

4.5. Tuning the PI Controller

In this subsection, the PI controller was tuned following the six-step process described in [30], which was summarised into three steps.

First, the controller output was simulated to approach the design level of operation and the data were recorded as process responses shown in Figure 8.

Second, the process data was analsed to estimate the process gain (K_p), process time constant (τ_p), and process dead time (θ_p) using a first order plus dead time (FOPDT) model given by Equation (17) below.

$$\tau_p \frac{d_{Tin(t)}}{dt} = -Tin(t) + K_p U(t - \theta_p) \quad (17)$$

Lastly, the resultant FOPDT model was used to obtain the initial tuning parameters for the PI controller given in Table 3.

The FOPDT model parameters estimated from the process data were used to generate initial estimations of the controller tuning parameters based on a correlation. The Internal Model Control (IMC) tuning correlations with moderate tuning [31] were then applied

to obtain the PI controller parameters. The moderate tuning involves setting τ_c to be the larger of $0.1.\tau_p$ or $0.8\,\theta_p$ i.e., $\tau_c = max(0.1\tau_p,\ 0.8\theta_p)$:

$$K_c = \frac{1}{K_p}\frac{\tau_p}{(\theta_p + \tau_c)} \tag{18}$$

$$T_i = \tau_p \tag{19}$$

where τ_c is the closed loop time constant, K_c is the proportional gain and T_i is the integral time constant. Table 6 below shows the resulting tuned controller parameters.

Table 6. Tuned PI controller parameters.

Parameters	Values
K_c	132.4
τ_i	151.9

The above controller parameters were used as the new PI algorithm controller constants and the resultant response is given in Figure 10 below.

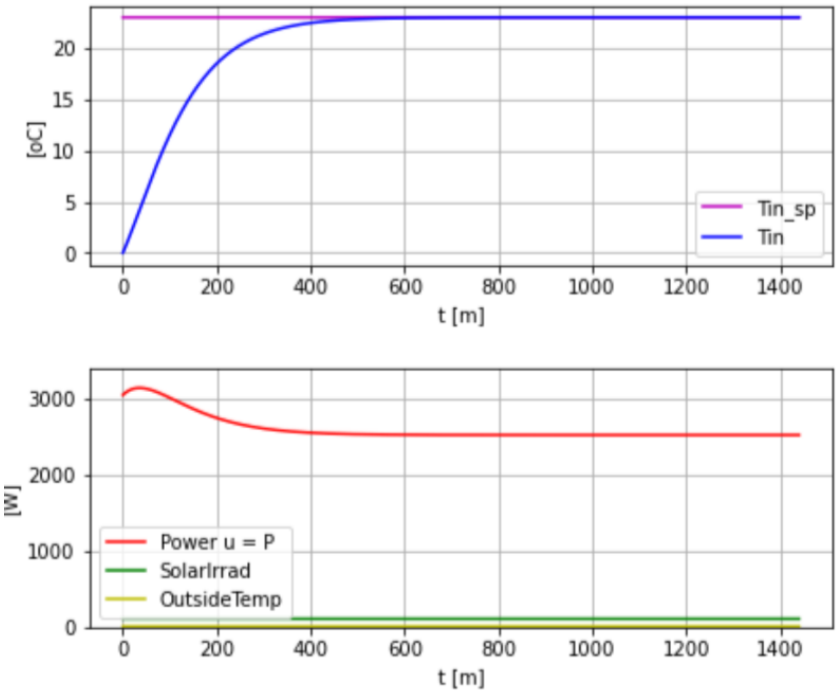


Figure 10. Tuned PI control output.

The ON/OFF controller is known for its simplicity and the fact that it does not require tuning, which makes it a popular choice for many applications. However, its inherent sustained oscillations can be a drawback. In this study, a comparison between the ON/OFF controller and the tuned PI controller (as shown in Figure 10) revealed that the PI controller offers smoother control, making it the preferred option for the greenhouse. The tuned PI controller not only provides excellent control with zero steady-state control error but also offers good system stability when the controller gain and integral time are appropriately tuned. In cases where heuristic tuning methods are required, appropriate values should be selected to ensure a good result. It is worth noting that the average steady-state control input signal (heat) of the two controllers is almost identical.

5. Conclusions

The greenhouse sector faces significant challenges with high energy consumption that require practical solutions. Precision agriculture, using IoT technology to monitor and control greenhouses, offers a promising solution. By incorporating efficient control strategies and hardware resources, real-time monitoring, and management of the heating system, which is the primary source of excessive energy use, becomes possible. Although IoT technology has been used in greenhouses, there is limited research on its use for minimising heating energy demand. This study presents an improved physics-based air temperature model that simulates a greenhouse equipped with IoT hardware resources for predicting indoor air temperature. The model was validated using measured data from an IoT cloud-based control system platform deployed in the greenhouse. The modified model, adapted through an optimisation-based model fitting technique, had a maximum discrepancy of 2 °C between simulated and actual measured indoor air temperature.

To control the greenhouse's air temperature, an on-off control and a PI control algorithm were implemented. While both controllers efficiently maintained the desired indoor temperature in simulation, the on-off controller had a non-zero mean control deviation from the setpoint, while the tuned PI controller offered smoother control and was chosen as the best option. However, the on-off controller's low cost makes it a simple and workable control design for a small- or medium-sized greenhouse IoT control system solution. Its performance in a real greenhouse has not been verified. Regarding energy-saving potential, the simulated results confirmed that the two control strategies required the same steady-state control input signals (heat) to maintain the desired indoor temperature. However, further experimentation or research is needed to verify the IoT cloud-based control system platform's use in a commercial-size greenhouse.

Future work will optimise the designed on-off control to achieve the plant's maximum tolerable temperature, enabling temperature range variations instead of a precise temperature setpoint. This approach, using optimisation techniques, will achieve energy minimisation. The performance of the optimised on-off controller will be compared with PI, PID, and an MPC controller, using optimisation methods for disturbance rejection and energy minimization.

Author Contributions: Conceptualization, Z.L.; Methodology, B.F. and Z.L.; Software, B.F.; Validation, B.F.; Formal analysis, B.F.; Data curation, B.F.; Writing—original draft, B.F.; Writing—review & editing, Z.L.; Visualization, Z.L.; Supervision, Z.L.; Funding acquisition, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data has been submitted to the university open access portal. The link is 10.17862/cranfield.rd.22265872.

Acknowledgments: We acknowledge Finn Haugen at the Faculty of Technology, Natural Sciences and Maritime Sciences, University of South-Eastern Norway. We express our sincere appreciation.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

$Ccap$	Heat capacity of the greenhouse air (J/m^2 °C)
R_{hoa}	The air density (~1.29 kg·m^{-3})
C_a	Specific heat capacity of air (at 5500 J·kg^{-1}·°C^{-1})
V_g	Greenhouse volume (m^3)
Q_{sun}	Heat load imposed on the greenhouse by the sun
C_{rad}	Transmitted (coefficient) solar radiation by structural components
Q_{rad}	Outdoor solar radiation (W m^{-2})

A_g	Greenhouse surface area in (m^2)
Q_{cover}	Convective heat loss via the cover
Q_{coe}	Convective heat transfer coefficient of polycarbonate
T_{out}	Outdoor temperature (°C)
T_{in}	Indoor temperature (°C)
A_c	Greenhouse cover material area (m^2)
Q_{soil}	Heat transfer between the internal air and the soil
K_{coe}	Soil heat transfer coefficient (W $°C^{-1}$)
T_s	Soil surface temperature in (°C)
$Qenergy$	Energy input term (W)
$Qenergy_{on}$	Heating system ON (W)
$Qenergy_{off}$	Heating system OFF (W)
$Qvent$	Heat lost through natural ventilation
$Qven$	Total ventilation flux from indoor to outdoor air (m^3 s^{-1})

Model parameters and values
$Crad = 0.1$
$Qrad = 108.2$
$Ag = 12.5$
$Qcoe = 0.21$
$Ac = 2.2$
$Ts = 11.05$
$Rhoa = 1.29$
$Kcoe = -7.88$
$Vg = 3.87$
$Ca = 1000.0$
$Qven = 0.29$

References

1. Iddio, E.; Wang, L.; Thomas, Y.; McMorrow, G.; Denzer, A. Energy efficient operation and modeling for greenhouses: A literature review. *Renew. Sustain. Energy Rev.* **2020**, *117*, 109480. [CrossRef]
2. Frausto, H.U.; Pieters, J.G. Modelling greenhouse temperature using system identification by means of neural networks. *Neurocomputing* **2004**, *56*, 423–428. [CrossRef]
3. Ullah, I.; Fayaz, M.; Aman, M.; Kim, D.H. An optimization scheme for IoT based smart greenhouse climate control with efficient energy consumption. *Computing* **2022**, *104*, 433–457. [CrossRef]
4. Rorabaugh, P.; Jensen, M.; Giacomelli, G. Introduction to Controlled Environment Agriculture and Hydroponics. *Control. Environ. Agric. Cent.* **2002**, 1–130. Available online: https://www.canr.msu.edu/floriculture/resources/energy/assets/GreenhouseEnergyConservationandAlternativesbyRorabaughetal.pdf (accessed on 9 March 2022).
5. Ahamed, M.S.; Guo, H.; Tanino, K. Energy saving techniques for reducing the heating cost of conventional greenhouses. *Biosyst. Eng.* **2019**, *178*, 9–33. [CrossRef]
6. Singh, M.C.; Singh, J.P.; Singh, K.G. Development of a microclimate model for prediction of temperatures inside a naturally ventilated greenhouse under cucumber crop in soilless media. *Comput. Electron. Agric.* **2018**, *154*, 227–238. [CrossRef]
7. van Beveren, P.J.M.; Bontsema, J.; van Straten, G.; van Henten, E.J. Minimal Heating and Cooling in a Modern Rose Greenhouse. *IFAC Proc. Vol.* **2013**, *4*, 282–287. [CrossRef]
8. Bersani, C.; Fossa, M.; Priarone, A.; Sacile, R.; Zero, E. Model Predictive Control versus Traditional Relay Control in a High Energy Efficiency Greenhouse. *Energies* **2021**, *14*, 3353. [CrossRef]
9. Zamora-Izquierdo, M.A.; Santa, J.; Martínez, J.A.; Martínez, V.; Skarmeta, A.F. Smart farming IoT platform based on edge and cloud computing. *Biosyst. Eng.* **2019**, *177*, 4–17. [CrossRef]
10. Jiang, J.; Moallem, M. Development of an Intelligent LED Lighting Control Testbed for IoT-based Smart Greenhouses. In Proceedings of the IECON 2020 the 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 18–21 October 2020; pp. 5226–5231. [CrossRef]
11. Afzali, S.; Mosharafian, S.; van Iersel, M.W.; Velni, J.M. Development and implementation of an iot-enabled optimal and predictive lighting control strategy in greenhouses. *Plants* **2021**, *10*, 2652. [CrossRef]
12. Vineela, T.; NagaHarini, J.; Kiranmai, C.; Harshitha, G.; AdiLakshmi, B. IoT Based Agriculture Monitoring and Smart Irrigation System Using Raspeberry Pi. *Int. Res. J. Eng. Technol.* **2018**, *5*, 1417–1420. Available online: https://www.irjet.net/archives/V5/i1/IRJET-V5I1307.pdf (accessed on 2 September 2020).
13. Navarro-Hellín, H.; Martínez-del-Rincon, J.; Domingo-Miguel, R.; Soto-Valles, F.; Torres-Sánchez, R. A decision support system for managing irrigation in agriculture. *Comput. Electron. Agric.* **2016**, *124*, 121–131. [CrossRef]

14. Thomopoulos, V.; Bitas, D.; Papastavros, K.N.; Tsipianitis, D.; Kavga, A. Development of an integrated IoT-based greenhouse control three-device robotic system. *Agronomy* **2021**, *11*, 405. [CrossRef]
15. Šaš, M.; Pejić, D. The application of IoT solutions in greenhouses with the aim of reducing electrical energy consumption. *J. Process. Energy Agric.* **2021**, *25*, 156–159. [CrossRef]
16. Sagheer, A.; Mohammed, M.; Riad, K.; Alhajhoj, M. A cloud-based IoT platform for precision control of soilless greenhouse cultivation. *Sensors* **2021**, *21*, 223. [CrossRef] [PubMed]
17. Zhang, S.; Guo, Y.; Zhao, H.; Wang, Y.; Chow, D.; Fang, Y. Methodologies of control strategies for improving energy efficiency in agricultural greenhouses. *J. Clean. Prod.* **2020**, *274*, 122695. [CrossRef]
18. Speetjens, S.L.; Stigter, J.D.; van Straten, G. Physics-based model for a water-saving greenhouse. *Biosyst. Eng.* **2010**, *105*, 149–159. [CrossRef]
19. Gutman, P.O.; Lindberg, P.O.; Ioslovich, I.; Seginer, I. A non-linear optimal greenhouse control problem solved by linear programming. *J. Agric. Eng. Res.* **1993**, *55*, 335–351. [CrossRef]
20. Seginer, I.; Ioslovich, I. Seasonal optimization of the greenhouse environment for a simple two-stage crop growth model. *J. Agric. Eng. Res.* **1998**, *70*, 145–155. [CrossRef]
21. van Ooteghem, R.J.C.; van Willigenburg, L.G.; van Straten, G. Receding horizon optimal control of a solar greenhouse. *Acta Hortic.* **2005**, *691*, 797–806. [CrossRef]
22. Chalabi, Z.S.; Bailey, B.J.; Wilkinson, D.J. Computers and electronics in agriculture A real-time optimal control algorithm for greenhouse heating. *Comput. Electron. Agric.* **1996**, *1*, 1–13. [CrossRef]
23. Henten, V.E.J. *Greenhouse Climate Management: An Optimal Control Approach*; Wageningen University and Research: Wageningen, The Netherlands, 1994.
24. Liang, M.H.; Chen, L.J.; He, Y.F.; Du, S.F. Greenhouse temperature predictive control for energy saving using switch actuators. *IFAC-PapersOnLine* **2018**, *51*, 747–751. [CrossRef]
25. Cook, F. Development of Apparatus for Ice Nucleation Studies. 2020. Available online: http://research-information.bristol.ac.uk (accessed on 29 December 2022).
26. Edgar, T.F.; Himmelblau, D.M.; Lasdon, L.S. *Process Dynamics and Control*, 2nd ed.; Wiley: New York, NY, USA, 2004.
27. Sangwan, A.; Gupta, A.; Tiwari, D.; Kumar, V.; Rana, K.P.S. Comparative Study of Optimization Techniques for Tuning of PI Gains for Greenhouse Climate Control. In Proceedings of the 2019 International Conference on Computing, Power and Communication Technologies, GUCON 2019, New Delhi, India, 27–28 September 2019; pp. 274–280.
28. Ulpiani, G.; Borgognoni, M.; Romagnoli, A.; di Perna, C. Comparing the performance of on/off, PID and fuzzy controllers applied to the heating system of an energy-efficient building. *Energy Build.* **2016**, *116*, 1–17. [CrossRef]
29. Hu, H.; Xu, L.; Goodman, E.D.; Zeng, S. NSGA-II-based nonlinear PID controller tuning of greenhouse climate for reducing costs and improving performances. *Neural. Comput. Appl.* **2014**, *24*, 927–936. [CrossRef]
30. Cooper, D.J. Practical Process Control® by Control Control Control Station® Station Station® Innovative Solutions from the Process Control Professionals Practical Process Control Using LOOP-PRO Software. 2005. Available online: https://vdocuments.net/practical-process-control-569de4b12a55a.html?page=1 (accessed on 29 December 2022).
31. Rice, R.C. *PID Tuning Guide A Best-Practices Approach to Understanding and Tuning PID Controllers*, 1st ed.; Technical Contributions from: Also Introducing: Simplifying PID Control, Optimizing Plant Performance; Control Station Inc.: Manchester, CT, USA, 2010.