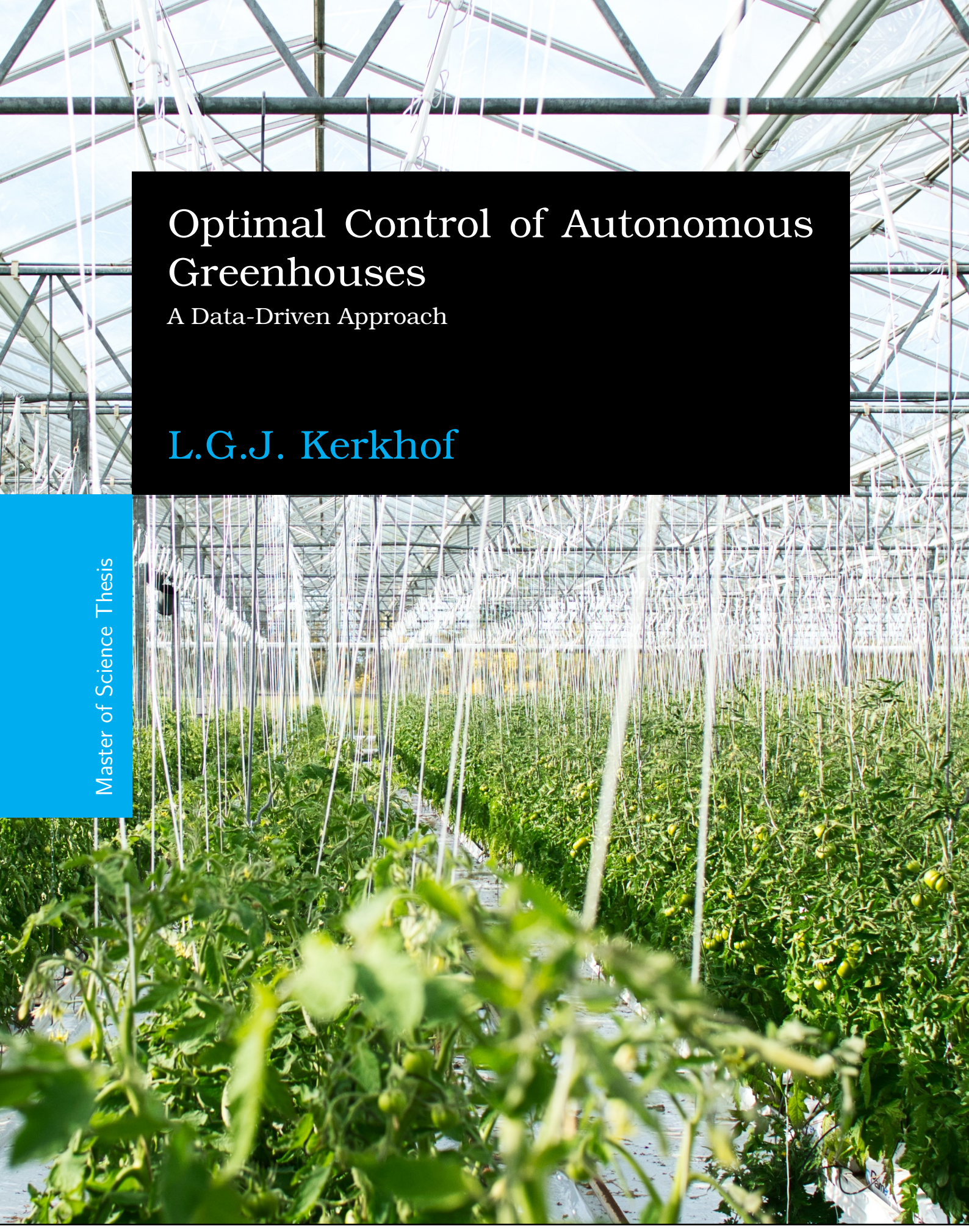

Optimal Control of Autonomous Greenhouses

A Data-Driven Approach

L.G.J. Kerkhof

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Optimal Control of Autonomous Greenhouses

A Data-Driven Approach

Master of Science Thesis

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

L.G.J. Kerkhof

July 13, 2020

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

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

Cover image by Erwan Hesry (https://unsplash.com/@erwanhesry).

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Optimal Control of Autonomous Greenhouses

by

L.G.J. Kerkhof

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: July 13, 2020

Supervisor(s):

Prof.dr.ir. T. Keviczky

Reader(s):

Dr.ir. S. Wahls

Dr.ir. K. Batselier

Ir. L. Baart de la Faille

Abstract

The world population is growing rapidly and the demand for healthy food grows with it. Greenhouse cultivation provides an efficient way to grow crops in a protected and controlled environment. In the past, many greenhouse control algorithms have been developed. However, the majority of these algorithms rely on an explicit parametric model description of the greenhouse. These models are often based on physical laws such as conservation of mass and energy and contain many parameters which should be identified. Due to the complex and highly non-linear dynamics of greenhouses, these models might not be applicable to control greenhouses other than the one for which the model has been designed and identified. Hence, in current horticultural practice these control algorithms are scarcely used. Therefore, the need rises for a control algorithm which does not rely on a parametric system representation but rather on input/output data of the greenhouse system, hereby establishing a way to control the system with unknown or unmodeled dynamics. A recently proposed algorithm, Data-Enabled Predictive Control (DeePC), is able to replace system identification, state estimation and future trajectory prediction by one single optimization framework. The algorithm exploits a non-parametric model constructed solely from input/output data of the system. This algorithm is employed in order to control the greenhouse climate. It is shown that in numerical simulation the DeePC algorithm is able to control the greenhouse climate. A comparison is made with the conventional Nonlinear Model Predictive Control (NMPC) algorithm in order to show the differences between a predictive control algorithm that has direct access to the non-linear greenhouse simulation model and a purely data-driven predictive control algorithm. Both algorithms are compared based on reference tracking accuracy and computational time. Furthermore, it is shown in numerical simulation that the DeePC algorithm is able to cope with changing dynamics within the greenhouse system throughout the crop cycle.

Table of Contents

List of Figures

List of Tables

Acknowledgements

This thesis in front of you is the result of the project I worked on for the past year and concludes my studies in Systems and Control. Also, it marks the end of my time as a student at the Delft University of Technology after six years. Working on this project has been quite a journey during which I learnt a tremendous amount, not only on professional but also on personal level.

Of course, this journey would not have been possible without the help of many people. First and foremost, I would like to thank my supervisor prof.dr.ir. Tamás Keviczky for his advice, time and inspiration. Your constructive feedback has been crucial during the process of this thesis project.

Furthermore, I would like to thank my fellow team members of team Automatoes: Evripidis, Rene, Gerdine, Leonard, Neil, Niek and Godfrey. Without them, this project would not have been such joyful as it has been now.

Finally, I would like to thank my friends and family who supported me throughout the past six years of my studies. Without all of you, I think it would have been nearly impossible for me to finish my studies. Especially, I would like to thank my parents for always supporting me and the decisions I made during my studies. Lastly, I would like to thank Gabrielle for her unconditional love, support and patience during this project.

Delft, University of Technology — L.G.J. Kerkhof
July 13, 2020

Chapter 1

Introduction

1-1 Motivation

The world population is increasing rapidly. According to the United Nations, the world population will increase to nearly 10 billion people in 2050 [1]. Currently an estimated amount of 821 million people is undernourished worldwide and this amount has been growing since 2014 [2]. Hence, as the world population is growing, the demand for healthy and fresh food grows as well.

Greenhouse cultivation plays an important role in providing fresh and healthy foods, such as fruits and vegetables. Due to their enclosure, greenhouses enable the control of climate conditions inside the greenhouse. Hence, this controlled indoor climate enables the manipulation of crop production by improving crop quality and decreasing the crop cultivation period [3]. Furthermore, the greenhouse provides a protection against insects, pests and diseases.

An important concern within the greenhouse industry is sustainability. According to the European Union, the share of energy consumption of agriculture with respect to the final energy consumption was 8.2% in 2017 in the Netherlands [4]. In addition, according to the Dutch government, the 2020 target CO_2 emission for the Dutch greenhouse industry that was agreed upon in 2014 is probably not going to be reached [5]. Hence, advanced greenhouse control algorithms could play an important role in reaching higher resource use efficiency and hence decrease the total energy and CO_2 consumption in the greenhouse industry, leading to a more sustainable horticultural sector.

Besides the food shortage and sustainability issues, one of the biggest problems within the greenhouse industry nowadays is finding sufficient experienced labor to manage crop production because the amount of experienced growers worldwide is declining [6]. A solution to overcome the shortfall of experienced labor is to increase the level of autonomy in greenhouse crop production. Hence, since the worldwide greenhouse vegetable production is increasing [7] and the greenhouse equipment is becoming more advanced [8], the necessity for advanced greenhouse control algorithms grows.

In order to increase the level of autonomy in the greenhouse industry, various model based control algorithms have been developed [9]. However, these algorithms usually exploit an explicit parametric model of the system dynamics. Difficulties in modelling the greenhouse dynamics arise in the fact that greenhouses exhibit complex and highly nonlinear behaviour [10]. Therefore, data-driven control algorithms, e.g. Data-Enabled Predictive Control (DeePC) [11], could provide a solution to this problem since such algorithms do not rely on an explicit parametric system representation but rather on input-output data of the system.

Furthermore, recent advancements in the field of Artificial Intelligence (AI) have led to major breakthroughs in fields such as autonomous driving [12], healthcare [13] and robotics [14]. Within the greenhouse industry, AI has been applied to yield prediction, disease detection, weed detection, crop quality classification and species recognition [15]. However, AI is still scarcely used in greenhouse climate and irrigation control and in making crop management decisions. Therefore, AI could contribute to a fully autonomous greenhouse cultivation period with yield levels compared to commercial practice [9]. For this reason, Wageningen University and Research and Tencent organise the Autonomous Greenhouse Challenge. During this Challenge, 5 teams get the opportunity to grow cherry tomatoes in a remotely controlled greenhouse compartment for six months. Since the greenhouse compartments are being remotely controlled, the teams are not allowed to enter their compartments and are ought to use advanced greenhouse control and/or AI algorithms which determine optimal settings for the greenhouse climate based on sensor measurements and weather predictions. The goal of this Challenge is to produce high quality tomatoes at a high level of production and with a high resource use efficiency. Together with researchers from Delft University of Technology and employees from Hoogendoorn Growth Management, Van der Hoeven Horticultural Projects and Keygene we will team up and participate during this Challenge.

1-2 Research Question

This thesis aims to develop a data-driven way to control greenhouses. This is done by employing the DeePC algorithm. This algorithm will be used to control the greenhouse model from [16] and its performance will be compared to the Nonlinear Model Predictive Control (NMPC) algorithm. Hence, the research question is stated as follows:

How does Data-Enabled Predictive Control perform compared to Nonlinear Model Predictive Control when controlling an autonomous greenhouse?

In order to support the research question, several sub-research questions with a more limited scope are formulated next:

- *Which model should be used to include in the NMPC algorithm and to generate data for the DeePC algorithm?*
- *Which of the available actuators are going to be controlled and which inputs could be retrieved from auxiliary controllers or forecasts?*
- *How can feedback of the crop status be included in the control loop?*
- *How is the performance of the DeePC algorithm going to be bench-marked against the NMPC algorithm?*

1-3 Thesis Outline

This thesis is organized as follows: in Chapter 2 an introduction is given to the greenhouse system in general and the dynamical greenhouse/crop models that are used in this thesis. In Chapter 3, the NMPC algorithm is introduced together with its application to the greenhouse system. In Chapter 4, the DeePC algorithm is introduced together with the mathematical background from behavioural systems theory on which the algorithm is relies. Thereafter, this algorithm will be applied to the greenhouse system. In Chapter 5, both the NMPC and DeePC algorithms will be used in a case-study where a reference is tracked during a particular day of which the outside conditions are available. Afterwards, both algorithms will be compared on their reference tracking accuracy and computation time. Furthermore, in this Chapter a comparison is made in the case that the DeePC algorithm controls the greenhouse while using data from the beginning of the crop cycle and data from when the crop is fully developed in order to show the relevance of the data that is used in the DeePC algorithm. In Chapter 6 an algorithm is proposed that controls the irrigation based on the water content in the substrate and the total daily solar radiation sum. Finally, in Chapter 7 a conclusion from the results will be drawn, the aforementioned research questions will be answered, the results of the Autonomous Greenhouse Challenge are discussed and recommendations for future work will be made.

Chapter 2

The Greenhouse System

2-1 Introduction

Worldwide, many different types and designs of greenhouses exist. Essential differences are seen in the greenhouse shape, cover material (e.g., glass or polyethylene) and the principles of operation [17]. For this thesis, the Venlo type greenhouse will be considered, which is shown in Figure 2-1. This glass covered, multi-span greenhouse is the most common greenhouse type in the Netherlands. Due to its transparent cover, solar radiation is allowed to enter the greenhouse in order to stimulate photosynthesis. Photosynthesis is the driving factor for plant and fruit growth and hence increasing the photosynthesis rate of plants will increase the crop yield [18]. Photosynthesis is the process where a crop converts solar radiation, CO_2, water and nutrients into assimilates such as carbohydrates which are used for plant maintenance and growth of fruits and leaves. Hence, modern greenhouse equipment allows the injection of CO_2 and additional heating in order to create the ideal plant growth climate. Furthermore, artificial lighting allows to add supplementary light when the solar radiation does not suffice and irrigation and nutrients could be provided as desired through an irrigation system. Even more advanced greenhouses contain shading screens, fogging systems or cooling pads as well.

Figure 2-1: Typical multi-span Venlo type greenhouse [19].

Besides the variations in greenhouse types and designs and control equipment, there is a large diversity of crops which can be grown in a greenhouse as well; ranging from bulk crops such as tomato, cucumber and sweet pepper to ornamental crops such as pot plants or flowers and medicinal plants [20]. Since all these different crops exhibit different dynamics due to their different growing processes, there exists a large variety of control inputs and objectives when it comes to controlling a greenhouse.

Within the greenhouse, many physical, biological and chemical processes take place simultaneously [21]. This results in the fact that the greenhouse system is a complex non-linear dynamical system that mainly consists of two subsystems: the greenhouse climate and the greenhouse crop. The greenhouse crop mainly reacts to the surrounding greenhouse climate, e.g., temperature, humidity and CO_2 concentration of the greenhouse air and light intensity. The greenhouse climate is mainly influenced by the control inputs such as heating, CO_2 injection and artificial lighting and exogenous inputs such as solar radiation, wind speed and outside temperature. Furthermore, the crop influences the climate as well due to e.g., transpiration, evaporation and CO_2 uptake. Another complexity arises due to the fact that the greenhouse crop dynamics are rather slow compared to the greenhouse climate dynamics. This difference in time scales makes it difficult to determine what the long term benefits of the yield will be when making short term decisions on resource usage.

Often, rejecting exogenous inputs is of major importance when controlling a system [22]. However, one of the exogenous signals that acts on the greenhouse system is solar radiation. This radiation is also the main driving force behind photosynthesis and on cold days it could be used to heat up the greenhouse. Furthermore, it might occasionally be necessary to use the cold outside air to cool down the greenhouse. Therefore, the exogenous signals should be exploited instead of rejected. Especially since the greenhouse has a transparent cover, the goal is to exploit the energy from the sun as much as possible. Hence, the optimal way to obtain high yield with a high resource use efficiency is by providing minimum heat and keeping the windows closed as much as possible in order to keep the heat, humidity and CO_2 inside the greenhouse.

In order to validate and benchmark the results of a greenhouse control algorithm, a greenhouse simulation model is required. Therefore, this chapter describes both the greenhouse climate and greenhouse crop subsystems in a schematic and mathematical way. For the greenhouse climate, a general description is given that includes common characteristics that hold for many greenhouse designs and crop varieties. For the greenhouse crop, a description of a tomato crop is given. Many different publications arise in which a mathematical model of the greenhouse climate is developed. For an extensive survey on these models, the reader is referred to [23] and the references therein. For this thesis, the greenhouse climate and crop model from [16] is used since it is developed for control purposes and describes processes such as ventilation and crop evaporation in a non-linear way while many climate models assume a linear relationship for these processes.

2-2 The Greenhouse Climate System

2-2-1 Schematic Description of the Greenhouse Climate

In this section, a description of the greenhouse climate system is given. In Figure 2-2 is schematically shown how the greenhouse climate interacts with the control inputs, the external environment and the greenhouse crop. In this scheme, the greenhouse and crop are considered as two separate subsystems: S_{g} and S_{c} , with states x_{g} and x_{c} , respectively.

The solid arrows in this figure represent the mass and energy fluxes whereas the dashed arrows represent the variables that influence these fluxes. These variables are the control inputs, the greenhouse states, the crop states and the disturbances acting on the greenhouse and will be called 'information flows' for conciseness as is done in [3].

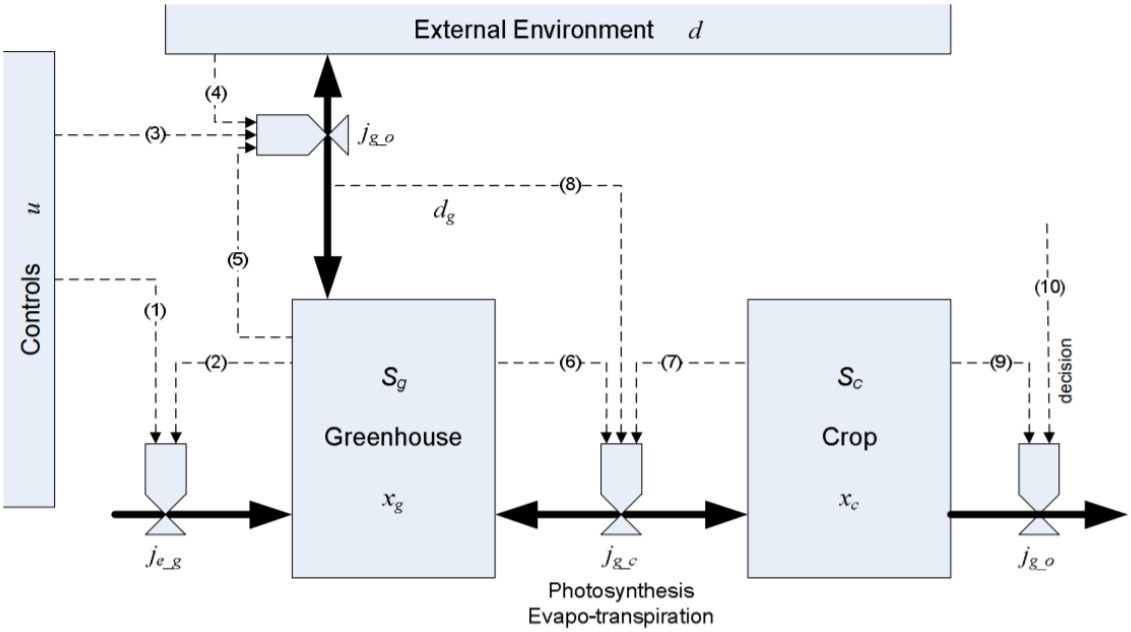

Figure 2-2: Block scheme of the greenhouse climate system [3].

Mass and energy fluxes between four different parts are distinguished: the fluxes between the control equipment and the greenhouse, the fluxes between the greenhouse and the outside environment, the fluxes between the greenhouse and the crop and the fluxes between the crop and output, denoted by j_{e_g}, j_{g_o}, j_{g_c} and j_{c_o}, respectively. Mass fluxes are e.g., the water and CO_2 fluxes from and to the crops and energy fluxes are e.g., the heating and solar energy fluxes from and to the greenhouse.

The information flows, denoted by (1), ... , (10) are described below:

- (1) The control inputs that are used for active climate control, e.g., heating, irrigation and CO_2 supply.

- (2), (5), (6) The greenhouse climate states, e.g., the greenhouse air temperature and humidity, the soil temperature, the temperature of the heating pipes and the CO_2 concentration of the greenhouse air.
- (3) The control inputs that steer the window opening on both the lee and wind side windows of the greenhouse (passive climate control).
- (4) The exogenous signals, i.e., all signals that come from the outside environment such as the outside air temperature and humidity, wind speed, wind direction and solar radiation.
- (7), (9) The states of the crop e.g., the mass content of the assimilate buffer, the weight of fruits and leafs on the crop and the growth stage of the crop.
- (8) The solar radiation input to the crop. The only external input that directly influences the crop since solar radiation is directly involved in the photosynthesis process. Hence, the solar radiation influences e.g., CO_2 uptake and transpiration of the crop and therefore the j_{g_c} flux.
- (10) The decision actions that influence the crop, e.g., picking leafs, pruning and harvesting fruits. In commercial practice, these actions are usually determined by the grower.

Hence, the greenhouse climate and crop subsystems interact in a complex process with each other, with the control inputs and with the external signals. This causes the entire greenhouse climate system to be a complex and highly non-linear dynamical system. Therefore, modeling of the entire greenhouse climate system is a difficult task in which many factors should be taken into account. The next section describes the greenhouse climate model developed in [16].

2-2-2 Greenhouse Climate Mathematical Model

The five greenhouse climate states are greenhouse air temperature T_{g} [°C], soil temperature T_{s} [°C], temperature of the heating pipes T_{p} [°C], CO_2 concentration C_{i} [g m^{-3}] and absolute humidity of the greenhouse air V_{i} [g m^{-3}]. In this subsection the differential equations of the greenhouse climate states are described. Furthermore, a number of parameters that appear in these equations are not constant but are rather defined by other functions. Therefore, these parameter functions are also described at the corresponding differential equations. The parameters that are constant are listed in Appendix B with their corresponding values.

Greenhouse air temperature

$$\dot{T}_{\mathrm{g}} = \frac{1}{C_{\mathrm{g}}}\left((k_{\mathrm{v}} + k_{\mathrm{r}})(T_{\mathrm{o}} - T_{\mathrm{g}}) + \alpha(T_{\mathrm{p}} - T_{\mathrm{g}}) + k_{\mathrm{s}}(T_{\mathrm{s}} - T_{\mathrm{g}}) + \eta G - \lambda E + \frac{\lambda}{1+\epsilon} M_{\mathrm{c}}\right) \tag{2-1}$$

Where:
C_{g} greenhouse air heat capacity [J $°C^{-1}m^{-2}$]

k_v ventilation heat transfer coefficient [W °C^{-1}m^{-2}]
k_r cover heat transfer coefficient [W °C^{-1}m^{-2}]
T_o outside temperature [°C]
α pipe heat transfer coefficient [W °C^{-1}m^{-2}]
k_s soil heat transfer coefficient [W °C^{-1}m^{-2}]
η greenhouse transmission [-]
G solar radiation [W m^{-2}]
λ evaporation energy water [J g^{-1}]
E transpiration rate crop [g s^{-1}m^{-2}]
M_c condensation mass flow [g s^{-1}m^{-2}]
ϵ greenhouse cover heat resistance [-]

The parameters listed above which are not constant but defined by a function instead are described below:

The ventilation heat transfer coefficient k_v is defined by the following function:

$$k_v = \rho_a c_p \Phi_v \tag{2-2}$$

Where, ρ_a is the air density [g m^{-3}], c_p is the air specific heat at constant pressure [J°C^{-1}m^{-2}] and Φ_v is the ventilation flux which on its turn is defined by:

$$\Phi_v = \left(\frac{\sigma \phi_{lee}}{1 + \chi \phi_{lee}} + \zeta + \xi \phi_{wind} \right) w + \psi \tag{2-3}$$

Where, ϕ_{lee} is the lee side opening [%], ϕ_{lee} is the wind side opening [%], w is the wind speed [ms^{-1}] and σ, χ, ζ, ξ and ψ are ventilation rate parameters.

The heat transfer coefficient of the pipes α is defined by:

$$\alpha = \nu \sqrt{\tau + \sqrt{|T_g - T_p|}} \tag{2-4}$$

Where, ν and τ are heat transfer coefficient parameters of the heating pipes.

The vaporisation energy of water is defined by:

$$\lambda = l_1 - l_2 T_g \tag{2-5}$$

Where, l_1 and l_2 are vaporisation energy coefficients.

The crop transpiration rate is defined by:

$$E = \frac{s\eta G + \rho_a c_p D_g g_b}{\lambda \Big(s + \gamma (1 + \frac{g_b}{g}) \Big)} \tag{2-6}$$

Where, s is the slope of the saturated water vapour pressure curve [kPa°C^{-1}], D_{g} is the air vapour pressure deficit [kPa], g_{b} is the leaf boundary layer conductance [ms^{-1}], γ is the apparent psychometric constant [kPa°C^{-1}] and g is the leaf conductance [ms^{-1}].

The slope of the saturated water vapour pressure curve is defined by:

$$s = s_1 T_{\mathrm{g}}^2 + s_2 T_{\mathrm{g}} + s_3 \tag{2-7}$$

Where, s_1, s_2 and s_3 are saturated water vapour pressure curve slope coefficients.

The air vapour pressure deficit is defined by:

$$D_{\mathrm{g}} = p_{\mathrm{g}}^* - p_{\mathrm{g}} \tag{2-8}$$

Where, p_{g}^* is the greenhouse air saturated vapour pressure [kPa] and p_{g} is the greenhouse air vapour pressure [kPa] which on their turn are defined by:

$$p_{\mathrm{g}}^* = a_1 e^{\frac{a_2 T_{\mathrm{g}}}{a_3 + T_{\mathrm{g}}}} \tag{2-9}$$

$$p_{\mathrm{g}} = \Lambda T_{\mathrm{g}} V_{\mathrm{i}} \tag{2-10}$$

Where, a_1, a_2 and a_3 are saturation vapour pressure parameters and Λ is a pressure constant derived from the ideal gas law [Nm°C^{-1}g^{-1}].

The leaf conductance is defined by:

$$g = g_1\Big(1 - g_2 e^{-g_3 G}\Big) e^{-g_4 C_{\mathrm{i}}} \tag{2-11}$$

Where, g_1, g_2, g_3 and g_4 are leaf conductance parameters.

Finally, the greenhouse cover condensation mass flow is defined by:

$$M_{\mathrm{c}} = \begin{cases} m_1 |T_{\mathrm{g}} - T_{\mathrm{c}}|^{m_2} (W_{\mathrm{g}} - W_{\mathrm{c}}^*), & \text{if } W_{\mathrm{g}} > W_{\mathrm{c}}^* \\ 0, & \text{if } W_{\mathrm{g}} \leq W_{\mathrm{c}}^* \end{cases} \tag{2-12}$$

Where, m_1 and m_2 are mass transfer parameters. T_{c} is the temperature at the greenhouse cover [°C]. W_{g} is the greenhouse air humidity ratio [-] at vapour pressure p_{g} and W_{c}^* is the greenhouse air humidity ratio [-] at saturated vapour pressure p_{g}^* at the cover and are calculated by:

$$W(p) = \frac{\omega p}{p_{\mathrm{atm}} - p} \tag{2-13}$$

Where, ω is a humidity ratio parameter and p_{atm} is the atmospheric pressure [kPa]. T_{c} is approximated by:

$$T_{\mathrm{c}} = \frac{\epsilon}{\epsilon + 1} T_{\mathrm{o}} + \frac{1}{\epsilon} T_{\mathrm{g}} \tag{2-14}$$

Heating pipe temperature

$$\dot{T}_{\mathrm{p}} = \frac{1}{V_{\mathrm{p}}}\left(\phi_{\mathrm{h}}(T_{\mathrm{h}} - T_{\mathrm{p}}) + \frac{A_{\mathrm{p}}}{\rho_{\mathrm{w}} C_{\mathrm{p}}}\left(\beta G - \alpha(T_{\mathrm{p}} - T_{\mathrm{g}})\right)\right) \tag{2-15}$$

Where:
V_{p} heating pipe volume [m^3]
ϕ_{h} opening heating valve [-]
T_{h} heating water temperature [°C]
A_{p} heating pipe surface [m^2]
ρ_{w} density water [g m^{-3}]
C_{p} specific heat of water at constant pressure [J°$\mathrm{C}^{-1}\mathrm{m}^{-2}$]
β heat absorption coefficient [-]

Soil temperature

$$\dot{T}_{\mathrm{s}} = \frac{1}{C_{\mathrm{s}}}\left(k_{\mathrm{s}}(T_{\mathrm{g}} - T_{\mathrm{s}}) + k_{\mathrm{d}}(T_{\mathrm{d}} - T_{\mathrm{s}})\right) \tag{2-16}$$

Where:
C_{s} greenhouse soil heat capacity [J °$\mathrm{C}^{-1}\mathrm{m}^{-2}$]
k_{d} deep soil heat transfer coefficient [W °$\mathrm{C}^{-1}\mathrm{m}^{-2}$]
T_{d} deep soil temperature [°C]

Greenhouse air CO_2 concentration

$$\dot{C}_{\mathrm{i}} = \left(\frac{V_{\mathrm{g}}}{A_{\mathrm{g}}}\right)^{-1}\left(\Phi_{\mathrm{v}}(C_{\mathrm{o}} - C_{\mathrm{i}}) + \phi_{\mathrm{inj}} + R - \mu P\right) \tag{2-17}$$

Where:
$\frac{V_{\mathrm{g}}}{A_{\mathrm{g}}}$ average greenhouse height [m]
Φ_{v} ventilation flux [m s^{-1}]
C_{o} outside CO_2 concentration [kg m^{-3}]
ϕ_{c} CO_2 injection flux [g s^{-1} m^{-2}]
R crop respiration [g $\mathrm{s}^{-1}\mathrm{m}^{-2}$]
μ fraction molar weight CO_2 and CH_2O [-]
P crop photosynthesis [g $\mathrm{s}^{-1}\mathrm{m}^{-2}$]

Greenhouse air absolute humidity

$$\dot{V}_{\mathrm{i}} = \left(\frac{V_{\mathrm{g}}}{A_{\mathrm{g}}}\right)^{-1}\left(E - \Phi_{\mathrm{v}}(V_{\mathrm{i}} - V_{\mathrm{o}}) - M_{\mathrm{c}}\right) \tag{2-18}$$

Where:
V_{o} outside humidity [kg m^{-3}]

Combining (2-1), (2-16), (2-15), (2-17) and (2-18) we obtain the following dynamical model for the greenhouse climate:

$$\dot{x}_{\mathrm{g}} = f(x_{\mathrm{g}}, u, v) \tag{2-19}$$

Where x_{g} denotes the greenhouse climate state vector, u denotes the vector with control inputs and v denotes the vector with external signals:

$$x_{\mathrm{g}} = \begin{bmatrix} T_{\mathrm{g}} \\ T_{\mathrm{p}} \\ T_{\mathrm{s}} \\ C_{\mathrm{i}} \\ V_{\mathrm{i}} \end{bmatrix}, \quad u = \begin{bmatrix} T_{\mathrm{h}} \\ \phi_{\mathrm{lee}} \\ \phi_{\mathrm{wind}} \\ \phi_{\mathrm{c}} \end{bmatrix}, \quad v = \begin{bmatrix} T_{\mathrm{o}} \\ T_{\mathrm{d}} \\ C_{\mathrm{o}} \\ V_{\mathrm{o}} \\ w \\ G \end{bmatrix} \tag{2-20}$$

2-3 The Greenhouse Crop System

2-3-1 Schematic Description of the Greenhouse Crop

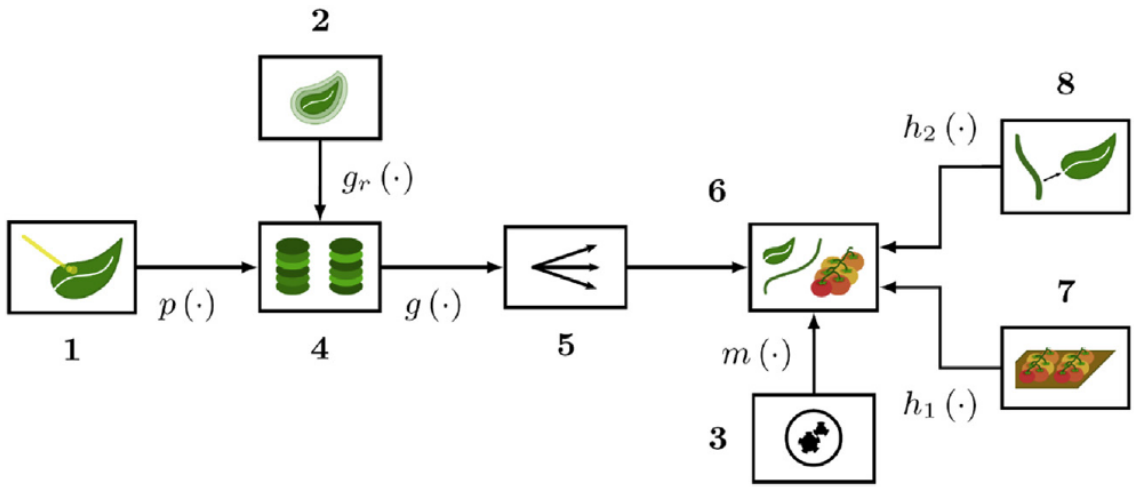

Figure 2-3: Block scheme of the greenhouse tomato crop system [24].

In Figure 2-3, a block scheme is shown which shows the main processes involved in the growth of a tomato crop. The diagram start with photosynthesis (1) where solar radiation is used to produce assimilates. The assimilates produced by photosynthesis are transferred (p) to the assimilate buffer (4) where accumulation of assimilates takes place. From this buffer assimilates are again transferred (g_r, g) in order to be used for growth respiration (2) and distribution (5) among the fruits, stems, leafs and roots as the crop grows. Growth respiration is the process where assimilates are combined with oxygen and converted to energy required for crop growth. After distribution, the assimilates are converted into biomass (6) or used for maintenance respiration (3, m). Finally, the resulting biomass can be harvested in the form of fruits (7, h_1) or leafs (8, h_2).

Obviously, the model here is a simplified version of the real crop growth process. The assumption is done that irrigation is done properly and the influence of fertigation is left out due to the complex, and partly unknown, chemical interaction between different fertilizers and the greenhouse crop. Furthermore, other processes such as pollination and diseases are omitted and assumed to cause no limitations in the growth of the crop.

2-3-2 Greenhouse Crop Mathematical Model

The greenhouse crop model used in this thesis is obtained from [16]. This model is a so called big-fruit big-leaf model, i.e., it considers the total fruit and leaf weight in single states instead of considering every fruit or leaf in a separate state. This approach is convenient for use in control algorithms since it limits the number of states substantially. For example, the tomato crop model developed in [25] uses 2 states per truss, 6 states per fruit, 4 for each vegetative unit (part of the stem and leaf corresponding to a truss) and 4 for the overall plant. Hence,

for a full-grown tomato crop with 6 trusses and 8 fruits per truss, the total number of states would be 328 states. This is in clear contrast with the model from [16], which only uses a total of four states to describe the crop.

The states described in this model are the assimilate buffer dry weight per ground area m_B [g m^{-2}], the weight of the fruits per ground area m_F [g m^{-2}], the weight of the leafs per ground area m_L [g m^{-2}] and the crop development stage D [-]. The crop states are represented by differential equations. Similar to the greenhouse climate states, there are parameters which appear in these equations which are not constant but rather depend on other functions. Again, these parameter functions are described at the corresponding differential equations and the parameters that are constant are listed in Appendix B with their corresponding values.

Each state is influenced by processes such as photosynthesis or fruit/leaf harvesting which are in their turn determined by equations.

Assimilate buffer dry weight

$$\dot{m}_B = P - b\left(f g_F m_F + v g_L \frac{m_L}{z}\right) - b\left(r_F m_F + r_L \frac{m_L}{z}\right) \tag{2-21}$$

Where:
P crop photosynthesis [g $s^{-1}m^{-2}$]
b buffer switching function [-]
f is the fruit assimilate requirement quotient [-]
g_F relative fruit growth rate [s^{-1}]
v is the vegetative requirement quotient [-]
g_L is the relative leaf growth rate [s^{-1}]
z is the leaf fraction of vegetative dry weight [-]
r_F is the relative fruit respiration rate [s^{-1}]
r_L is the relative leaf respiration rate [s^{-1}].

The first term on the right hand side in (2-21) represents the assimilate production by photosynthesis. The middle term on the right hand side in (2-21) represents the assimilates to biomass conversion and distribution where constant distribution parameters are assumed. The right term on the right hand side in (2-21) represents the respiration for both the fruits and the vegetative part of the crop. The parameters listed above which are not constant but defined by a function instead are described below:

The crop photosynthesis is defined by the following function:

$$P = P_m l \frac{I}{p_1 + I} \frac{C}{p_2 + C} \tag{2-22}$$

Where P_m is the maximum photosynthesis rate [g $s^{-1}m^{-2}$], l is the LAI correction function [-], I is the PAR [μmol s^{-1-2}], C is the CO_2 concentration [ppm] and p_1, p_2 are photosynthesis parameters.

The PAR is defined by:

$$I = \eta m_p G \tag{2-23}$$

Where m_{p} is the Watt to μmol conversion factor.

The CO_2 in ppm is calculated by:

$$C = \frac{10^6 R_{\mathrm{g}}}{M_{\mathrm{CO_2}} p_{\mathrm{atm}}} (T_{\mathrm{g}} + T_0) C_{\mathrm{i}} \tag{2-24}$$

Where R_{g} is the gas constant [Jmol^{-1}K^{-1}], $\mathrm{M_{CO_2}}$ is the molar mass of CO_2 [kg] and T_0 is a factor to convert the temperature from Celsius to Kelvin.

The LAI correction function is defined by:

$$l = \frac{\left(\frac{W_{\mathrm{L}}}{w_{\mathrm{R}}}\right)^m}{1 + \left(\frac{W_{\mathrm{L}}}{w_{\mathrm{R}}}\right)^m} \tag{2-25}$$

Where w_{R} and m are LAI correction function coefficients.

The buffer switching function is defined by:

$$b = 1 - e^{-b_1 B} \tag{2-26}$$

This function tends to zero when the assimilate buffer is empty and tends to one when the assimilate buffer is full; hence the name switching function. b_1 is a buffer switching parameter.

The relative fruit and leaf growth rates, respectively, are defined by:

$$g_{\mathrm{F}} = (f_1 - f_2 D_{\mathrm{P}}) Q_{\mathrm{G}}^{\frac{T_{\mathrm{g}} - T_{\mathrm{G}}}{10}} \tag{2-27}$$

$$g_{\mathrm{L}} = g_{\mathrm{F}} v_1 e^{v_2 (T_{\mathrm{g}} - v_3)} \tag{2-28}$$

Where f_1 and f_2 are fruit growth rate coefficients, Q_{G} is the Q_{10}-value for the temperature effect on fruit growth rate, T_{G} is the growth rate reference temperature and v_1, v_2 and v_3 are vegetative fruit growth ratio coefficients.

The relative fruit and leaf respiration rates, respectively, are defined by:

$$r_{\mathrm{F}} = M_{\mathrm{F}} Q_{\mathrm{R}}^{\frac{T_{\mathrm{g}} - T_{\mathrm{G}}}{10}} \tag{2-29}$$

$$r_{\mathrm{L}} = M_{\mathrm{L}} Q_{\mathrm{R}}^{\frac{T_{\mathrm{g}} - T_{\mathrm{G}}}{10}} \tag{2-30}$$

Where Q_{R} is the Q_{10}-value for the temperature effect on the maintenance respiration and M_{F} and M_{L} are the fruit and leaf maintenance respiration coefficients, respectively.

Fruit weight

$$\dot{m}_{\mathrm{F}} = \Big(bg_{\mathrm{F}} - (1-b)r_{\mathrm{F}} - h_{\mathrm{F}}\Big)m_{\mathrm{F}} \tag{2-31}$$

Where:
h_{F} fruit harvest rate [s^{-1}]

Here, the first term on the right hand side represents the growth of the fruit. The middle term on the right hand side represents the respiration necessary for fruit growth and the right term on the right hand side represents the harvest of the fruits.

Leaf weight

$$\dot{m}_{\mathrm{L}} = \Big(bg_{\mathrm{L}} - (1-b)r_{\mathrm{L}} - h_{\mathrm{L}}\Big)m_{\mathrm{L}} \tag{2-32}$$

Where:
h_{L} leaf harvest rate [s^{-1}]

Here, the first term on the right hand side represents the growth of the leafs. The middle term on the right hand side represents the respiration necessary for leaf growth and the right term on the right hand side represents the harvest of the leafs.

The harvest coefficient is defined by:

$$h = \begin{cases} 0, & \text{if } 0 < D < 1 \\ d_1 + d_2 \ln\left(\frac{T_{\mathrm{g}}}{d_3}\right) + d_4 t, & \text{if } D = 1 \end{cases} \tag{2-33}$$

Then, the fruit and leaf harvest coefficients are defined by:

$$h_{\mathrm{F}} = y_{\mathrm{F}} h \tag{2-34}$$

$$h_{\mathrm{L}} = y_{\mathrm{L}} h \tag{2-35}$$

Here, y_{F} and y_{L} are the fruit harvest coefficient parameters.

Crop development stage

$$\dot{D} = d_1 + d_2 + \ln\left(\frac{T_{\mathrm{g}}}{d_3}\right) + d_4 t - h \tag{2-36}$$

Where:
d_1, d_2, d_3 and d_4 are plant development rate parameters
t is the time
h is the harvest coefficient [s^{-1}].

The crop development stage starts at zero and increases to one while the crop develops. The moment when D becomes one means that the crop has developed in such a way that the first fruits can be harvested. At that moment, D remains constant, i.e., $\dot{D} = 0$.

Finally, for a given period from t_0 until t_f the total harvest of fruits and leaves is computed as follows:

$$W_{H_F} = \int_{t_0}^{t_f} h_F W_F dt \tag{2-37}$$

$$W_{H_L} = \int_{t_0}^{t_f} h_L W_L dt \tag{2-38}$$

Combining (2-21), (2-31), (2-32) and (2-36) we obtain the following dynamical model for the greenhouse crop:

$$\dot{x}_c = f(x_c, x_g, t, v) \tag{2-39}$$

Where x_g and v are as defined in (2-20), t is the time and x_c denotes the vector that contains the greenhouse crop states:

$$x_c = \begin{bmatrix} m_B \\ m_F \\ m_L \\ D \end{bmatrix} \tag{2-40}$$

2-4 Greenhouse Simulation

2-4-1 Discretization and Implementation

By combining both the greenhouse climate and greenhouse crop models, the complete greenhouse model is obtained:

$$\dot{x} = f_{gh}(x, u, v, t) \tag{2-41}$$

Where $x = [x_g^T, x_c^T]^T$ and f_{gh} contains the state transition equations: (2-1), (2-16), (2-15), (2-17), (2-18) (2-21), (2-31), (2-32) and (2-36).

Now, for implementation purposes, the continuous time model (2-41) is discretized using Euler's method. Hence, the states are updated in the following discrete way:

$$x(t+1) = x(t) + h * f_{gh}(x(t), u(t), v(t), t) \tag{2-42}$$

where h is the sampling time.

Using this discretization method, the following non-linear discrete time model is obtained:

$$x(t+1) = f_{gh}(x(t), u(t), v(t), t) \tag{2-43}$$

Since the dynamics of the greenhouse are relatively slow, the sampling time h is chosen to be 300 seconds throughout this thesis. Furthermore, this is also convenient due to the fact that the weather condition measurements are also sampled every 5 minutes. The discrete time greenhouse simulation model (2-41) is subsequently implemented in Matlab.

2-4-2 Simulation Example

Next, a numerical simulation is shown where the effect of the crop on the greenhouse climate is clearly visible. Two days are simulated with the exact same (simple) control inputs, see Figure 2-4, and the exact same exogenous inputs. In the first case, the crop development stage D is 0 and in the second case, the crop is fully developed ($D = 1$) and hence has more leaf and fruit weight per square meter.

In Figure 2-5 is shown how the fully developed crop influences the greenhouse climate such that it lowers the greenhouse air temperature (top left), decreases the CO_2 concentration (bottom left) and increases the absolute humidity in the greenhouse (bottom right). These effects can be explained due to the fact that the fully developed crop has a larger leaf area. Therefore, the crop takes up more CO_2 since photosynthesis mainly takes place in the leafs. Also, due to the higher leaf area, the crop evaporates more water which leads to an increase in the absolute humidity of the greenhouse air. Subsequently, the evaporation of the increased humidity leads on its turn to a decrease in temperature. Furthermore, due to the colder greenhouse air, the temperature of the heating pipe decreases slightly as well since it has a constant heating water temperature.

In Figure 2-6 the states of the young and the fully develop crop are shown. As can be seen from the middle two and bottom two plots, the crop has a larger fruit and leaf weight in the fully developed case. From the top two plots can be seen that the fully developed crop creates faster and more assimilates compared to the young crop but is restricted due to CO_2 limitations. Furthermore, from the top two plots can be seen that there is only assimilate production when there is solar radiation, which makes sense since only then photosynthesis can take place. Finally, it displays that the assimilate buffer gets depleted after sunset since there is still a demand for assimilates from the crop but no assimilates are produced anymore.

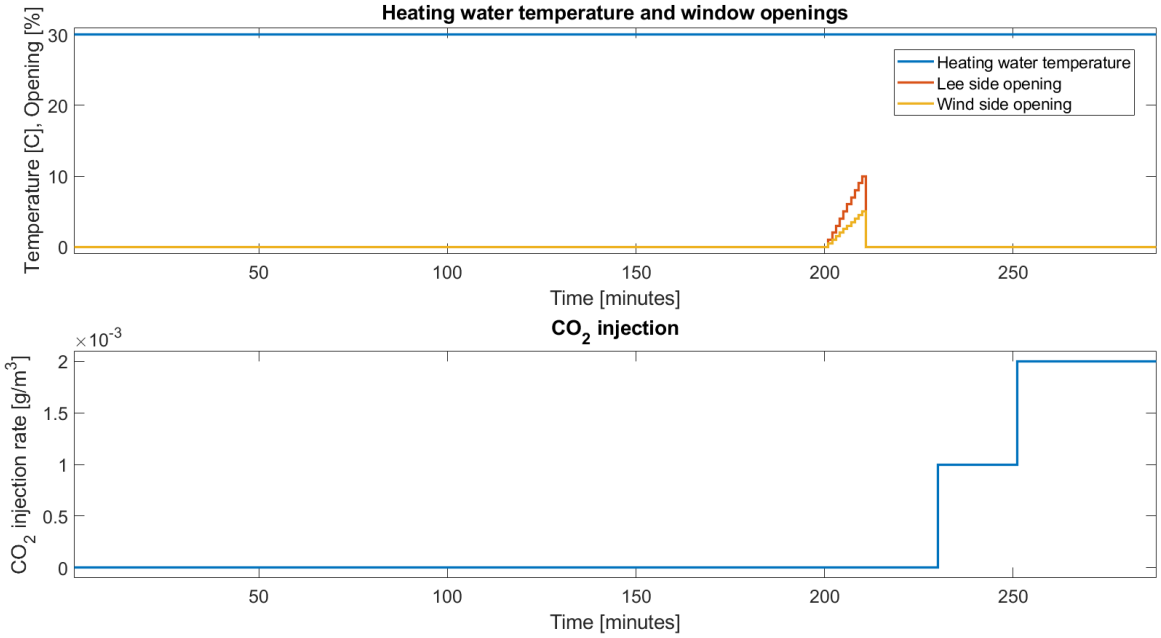

Figure 2-4: The control inputs that are used in both numerical simulations.

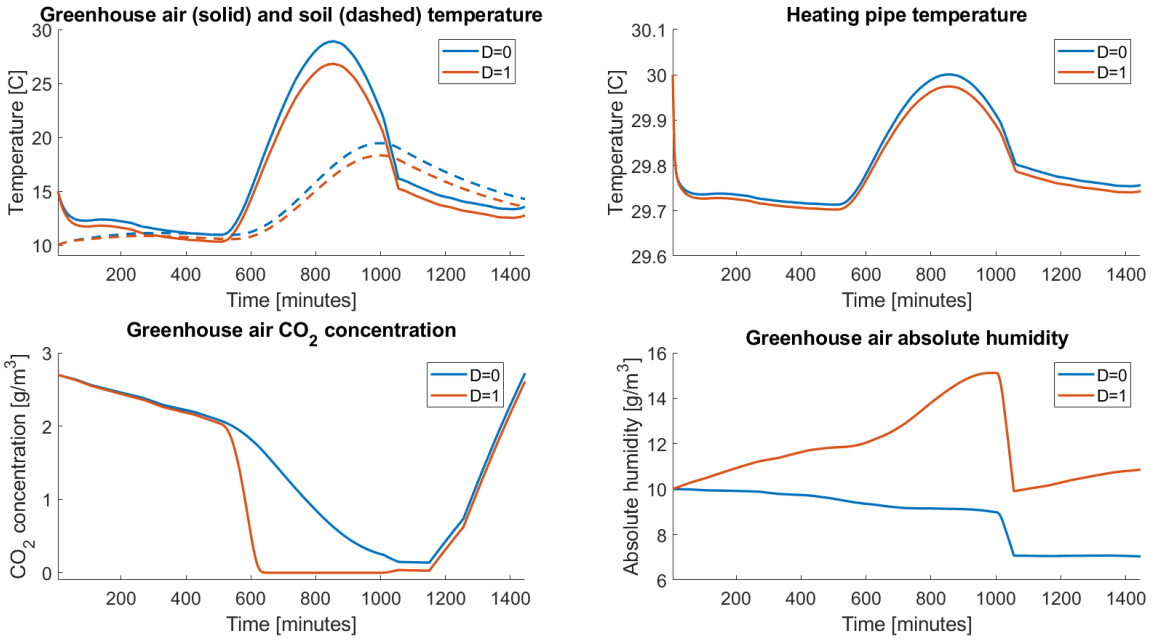

Figure 2-5: Greenhouse climate states that show the effect of a young crop and a fully developed crop on the greenhouse climate.

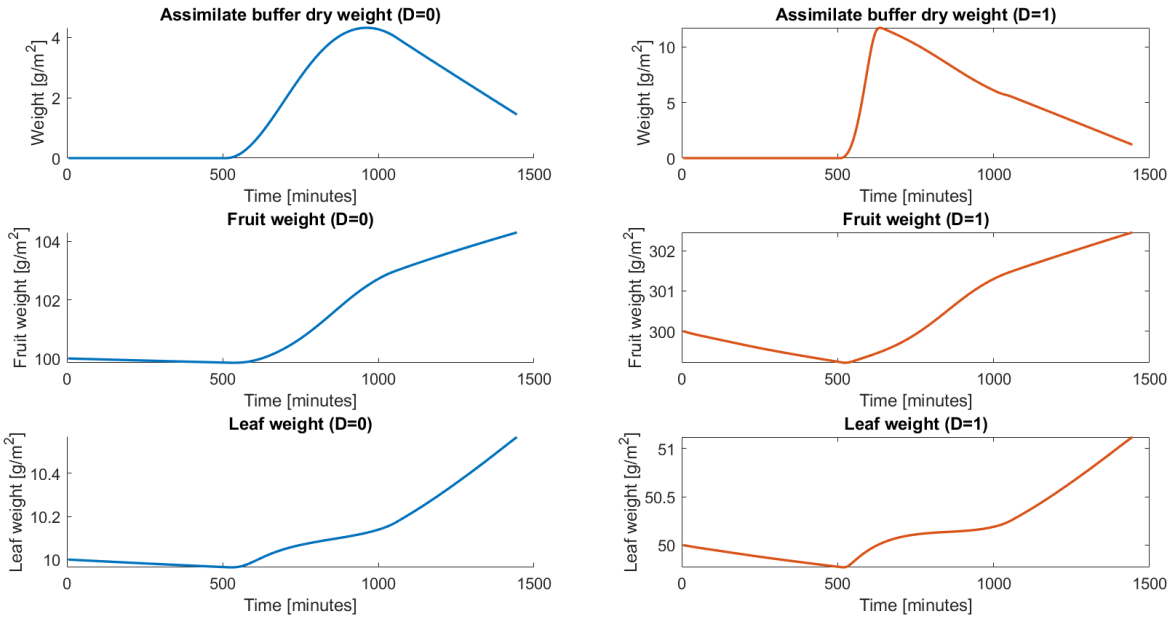

Figure 2-6: Greenhouse crop states that show the states of a young crop (left) and a fully developed crop (right).

Chapter 3

Non-Linear Model Predictive Control

3-1 Introduction

Model Predictive Control (MPC) is a predictive control algorithm that makes use of an explicit model of the system it aims to control. At each time step the algorithm computes the optimal control input signal over a prediction horizon in order to reach a certain objective [26]. A frequently appearing objective in MPC is reference tracking, see Figure 3-1 for a graphical representation. Furthermore, a typical example of an explicit dynamical system model is the discrete time linear state space system:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{3-1}$$

Here $x(t) \in \mathbb{R}^n$ denotes the system state, $u(t) \in \mathbb{R}^m$ denotes the control input and $y(t) \in \mathbb{R}^p$ denotes the system output at time $t \in \mathbb{Z}_+$. Furthermore, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$ and $D \in \mathbb{R}^{p \times m}$ are the system matrices.

Hence, when an explicit model of a system such as (3-1) is available, an optimization problem can be formulated where a cost function is minimized while the problem is subject to the system dynamics, control input constraints and state constraints [11]. Therefore, in (3-2) a general linear MPC optimization framework is shown, which is solved at every time step [27]. Then, in Algorithm 1 the outline of the, in this case linear, MPC algorithm is shown [28].

$$\begin{aligned} \min_{u} \quad & V_{\mathrm{N}}(x,u,r) = \sum_{k=0}^{N-1} \ell(x_k, u_k, r_k) + V_{\mathrm{f}}(x_N, r_N) \\ \text{subject to} \quad & x_{k+1} = Ax_k + Bu_k, \quad \forall k \in \{0,...,N-1\}, \\ & y_k = Cx_k + Du_k, \quad \forall k \in \{0,...,N-1\}, \\ & x_0 = \hat{x}(t), \\ & u_k \in \mathcal{U}, \quad \forall k \in \{0,...,N-1\}, \\ & x_k \in \mathcal{X}, \quad \forall k \in \{0,...,N-1\}. \end{aligned} \tag{3-2}$$

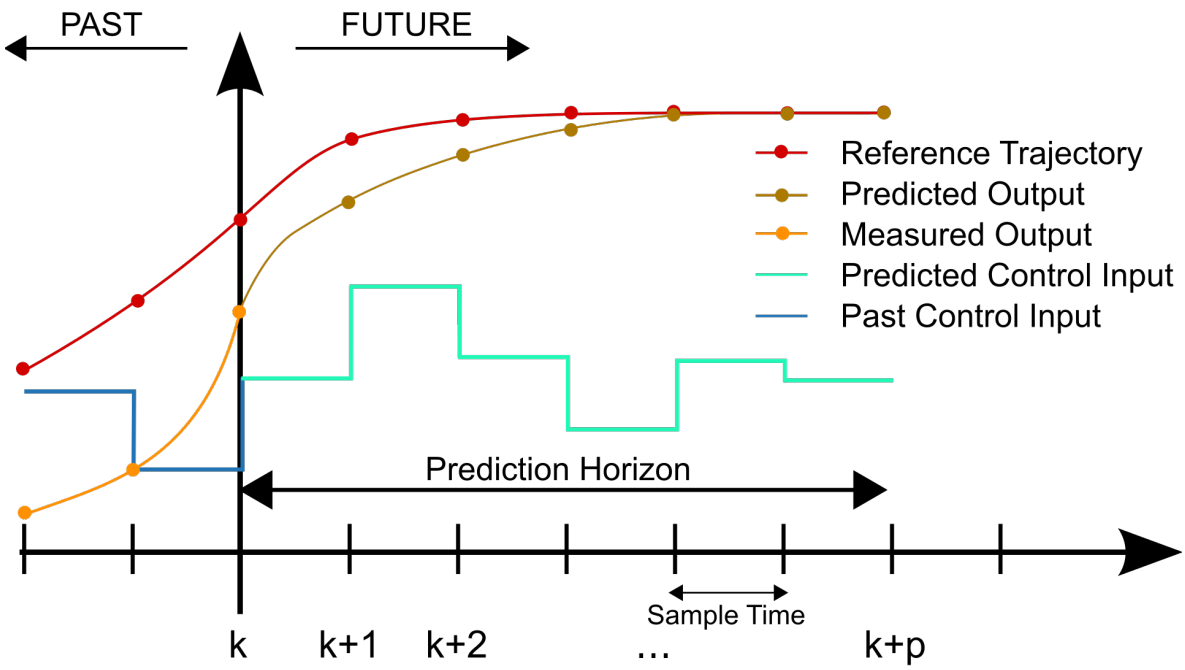

Figure 3-1: MPC scheme [29].

In (3-2), $V_{\mathrm{N}}(x, u, r)$ is the cost function which is minimized. In this cost function, $\ell(x_k, u_k, r_k)$ is the stage cost computed at each time index $k \in \mathbb{Z}_+$ and $V_{\mathrm{f}}(x_N, r_N)$ is the terminal cost computed at the end of the prediction horizon $k = N$.

Furthermore, $N \in \mathbb{Z}_{++}$ denotes the prediction horizon horizon, $u = (u_0, ..., u_{N-1})$ denote the control inputs which are the optimization variables, $x = (x_0, ..., x_N)$ denote the system states as predicted by the model, $y = (y_0, ..., y_{N-1})$ denote the predicted system output as predicted by the model and $r = (r_0, ...r_N)$ is the desired state or output reference trajectory.

Besides, $\hat{x}(t)$ denotes the state estimate at time t where $t \in \mathbb{Z}_+$ is the time step at which the optimization problem is solved. Hence, if full state measurement is available, $\hat{x}(t) = x(t)$. Else, in case full state measurement is not available and the system is observable, an observer is typically used to estimate the state.

Finally, $\mathcal{U} \subseteq \mathbb{R}^m$ and $\mathcal{X} \subseteq \mathbb{R}^n$ denote the sets of input and state constraints, respectively.

Algorithm 1: MPC

Input: System matrices (A, B, C, D), prediction horizon N, constraint sets $\mathcal{U}$ and $\mathcal{X}$;
1) Obtain initial state estimate $\hat{x}(t)$.
2) Solve (3-2) for the optimal input sequence $u^\star = (u_0^\star, ..., u_{N-1}^\star)$.
3) Apply only the first input $u(t) = (u_0^\star)$.
4) Set t to $t + 1$.
5) Return to 1.

3-2 Non-Linear Model Predictive Control

Essentially all systems are inherently stochastic and non-linear. Therefore, using (3-1) might not give a sufficient accurate representation of the true system dynamics. Especially in the case of the highly non-linear dynamics of the greenhouse system, a non-linear system representation (3-3) is inevitable. Therefore, the classic non-linear discrete time system description is given here:

$$\begin{aligned} x(t+1) &= f(x(t), u(t), v(t)) \\ y(t) &= h(x(t), u(t), v(t)) \end{aligned} \tag{3-3}$$

Here, $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q \to \mathbb{R}^p$ denote the nonlinear functions which map the current state and input to the next state and current output, respectively. Furthermore, $v(t)$ denote the exogenous inputs at time t. With this nonlinear system representation, we arrive at the following Nonlinear Model Predictive Control (NMPC) framework [30]:

$$\begin{aligned} \min_{u} \quad & V_{\mathrm{N}}(x,u) = \sum_{k=0}^{N-1} \ell(x_k, u_k, r_k) + V_{\mathrm{f}}(x_N) \\ \text{subject to} \quad & x_{k+1} = f(x_k, u_k), \quad \forall k \in \{0, ..., N-1\}, \\ & y_k = h(x_k, u_k), \quad \forall k \in \{0, ..., N-1\}, \\ & x_0 = \hat{x}(t), \\ & u_k \in \mathcal{U}, \quad \forall k \in \{0, ..., N-1\}, \\ & x_k \in \mathcal{X}, \quad \forall k \in \{0, ..., N-1\}. \end{aligned} \tag{3-4}$$

which is solved at every time step $t \in \mathbb{Z}_+$. Now, the algorithm for NMPC is shown in Algorithm 2.

Algorithm 2: NMPC

Input: Non-linear system equations $f(x(t), u(t), v(t))$ and $h(x(t), u(t), v(t))$, prediction horizon N, constraint sets $\mathcal{U}$ and $\mathcal{X}$;

1) Obtain initial state estimate $\hat{x}(t)$.
2) Solve (3-4) for the optimal input sequence $u^\star = (u_0^\star, ..., u_{N-1}^\star)$.
3) Apply only the first input $u(t) = (u_0^\star)$.
4) Set t to $t+1$.
5) Return to 1.

3-3 Greenhouse Controlled by NMPC

In this section, two numerical simulations will be shown where the NMPC algorithm is used to control the greenhouse system. In the first example, NMPC will be used for tracking a reference temperature with the greenhouse air temperature. In the second example, the goal will be to maximize the yield of the greenhouse crop while keeping the greenhouse air temperature between a minimum and a maximum bound. In both the reference tracking and yield maximization simulations, (2-19) and (2-39) will be used as the non-linear system equations which are required in the NMPC optimization framework (3-4). Furthermore, in both simulations (and the remainder of this thesis) the SNOPT solver from the TOMLAB Optimization Environment is used to solve the non-linear optimization problems [31].

3-3-1 Weather Conditions

The exogenous signals in case of the greenhouse system consists of the outside weather conditions. Hence, in both the simulations here the same day of measured weather conditions will serve as the exogenous inputs. This data is obtained through LetsGrow.com and shown in Figure 3-2. In the bottom plot the solar radiation is shown and in the top plot the outside temperature, outside absolute humidity, wind speed, outside CO_2 concentration and the deep soil temperature are shown. The outside CO_2 concentration and deep soil temperature signals are assumed constant at 0.1 g/m^3 and 10 °C, respectively and at each time index a random variable drawn from a standard normal distribution multiplied with 0.01 is added to these constants.

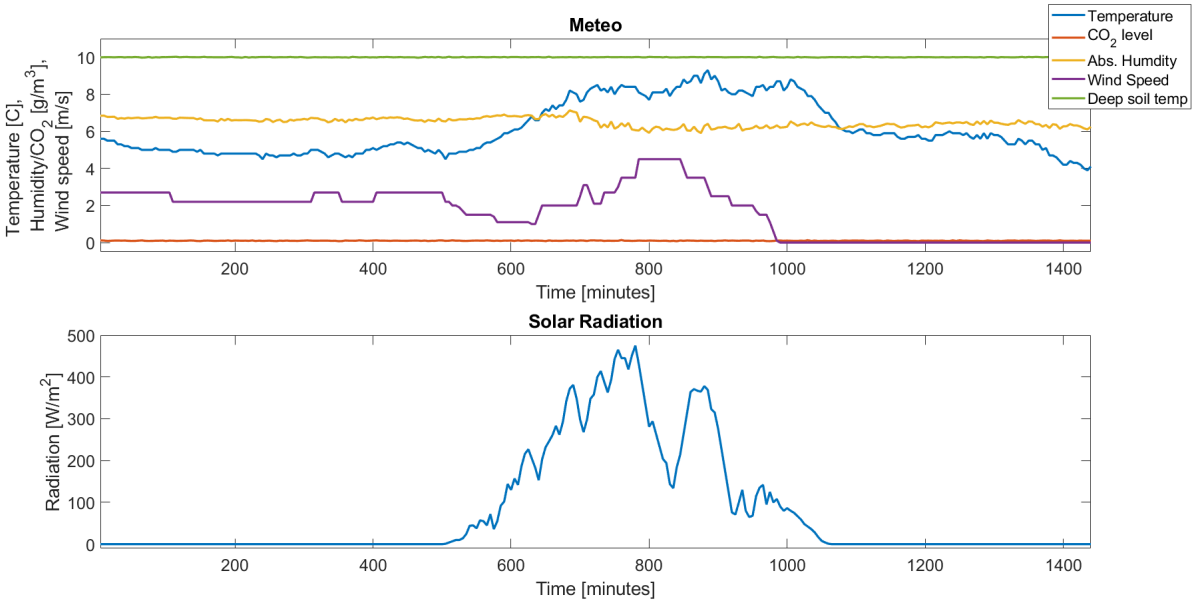

Figure 3-2: Weather conditions used in the NMPC simulations.

3-3-2 Reference Tracking

In this numerical simulation, the objective will be to track a reference with the greenhouse air temperature while minimizing the control input. Furthermore, a penalty is given to the change of in control inputs in order to avoid rapid changes in window position or heating pipe temperatures since this could lead to breakdown of the actuators due to fatigue. The control inputs will be constrained due to the physical limitations of the actuators. This means that the heating water temperature will be constrained between 10°C and 80 °C, the window opening will be constrained between 0% and 100%. The CO_2 injection will be constrained between 0 g $s^{-1}m^{-2}$ and 2*10^{-3} g $s^{-1}m^{-2}$. Hence, the optimization problem for reference tracking becomes:

$$\begin{aligned} \min_{u} \quad & \sum_{k=0}^{N-1} \|T_{g,k} - r_k\|_Q^2 + \|u_k\|_R^2 + \|u_k - u_{k-1}\|_{R_\Delta}^2 \\ \text{subject to} \quad & x_{k+1} = f_g(x_k, u_k), \quad \forall k \in \{0, ..., N-1\}, \\ & x_0 = \hat{x}(t), \\ & u_l \leq u_k \leq u_u, \quad \forall k \in \{0, ..., N-1\}. \end{aligned} \tag{3-5}$$

Here, $Q \succeq 0$, $R \succeq 0$ and $R_\Delta \succeq 0$ are positive semi-definite matrices that represent the cost on reference deviation, control input and change of control input, respectively. Furthermore, $T_{g,k}$ denotes the temperature of the greenhouse air at time k and u_l and u_u are the lower and upper bounds of the set $\mathcal{U}$, respectively. For this numerical simulation, full state measurement is assumed, no terminal cost is used and the parameters are chosen as follows:
$N = 12$
$Q = \text{diag}(500, 0, ..., 0)$
$R = 0.1 I_4$
$R_\Delta = r_\Delta I_4$
$u_l = [10, 0, 0, 0]^T$
$u_u = [80, 100, 100, 2 * 10^{-3}]^T$

Two different simulations are done for $r_\Delta = 10$ and $r_\Delta = 1$ in order to show the differences when the penalty on change of control is larger.

In Figure 3-3 and Figure 3-4 the results of the reference tracking simulations are shown. As can be seen from Figure 3-3, the greenhouse air temperature is driven to the reference as desired. Furthermore, the control input signals are smoother for $r_\Delta = 10$ while better reference tracking is achieved while using $r_\Delta = 1$. Hence, a trade-off needs to be made between the accuracy of the reference tracking and the aggressiveness of the control inputs.

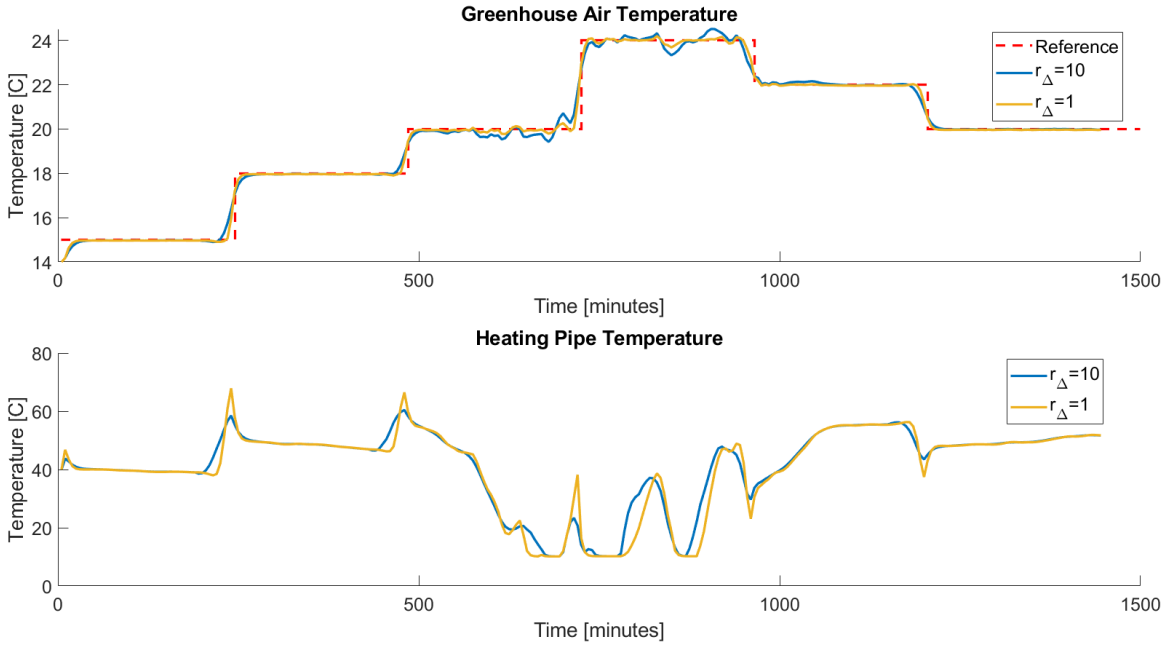

Figure 3-3: Greenhouse air (top) and heating pipe (bottom) temperatures controlled by NMPC for $r_\Delta = 10$ and $r_\Delta = 1$.

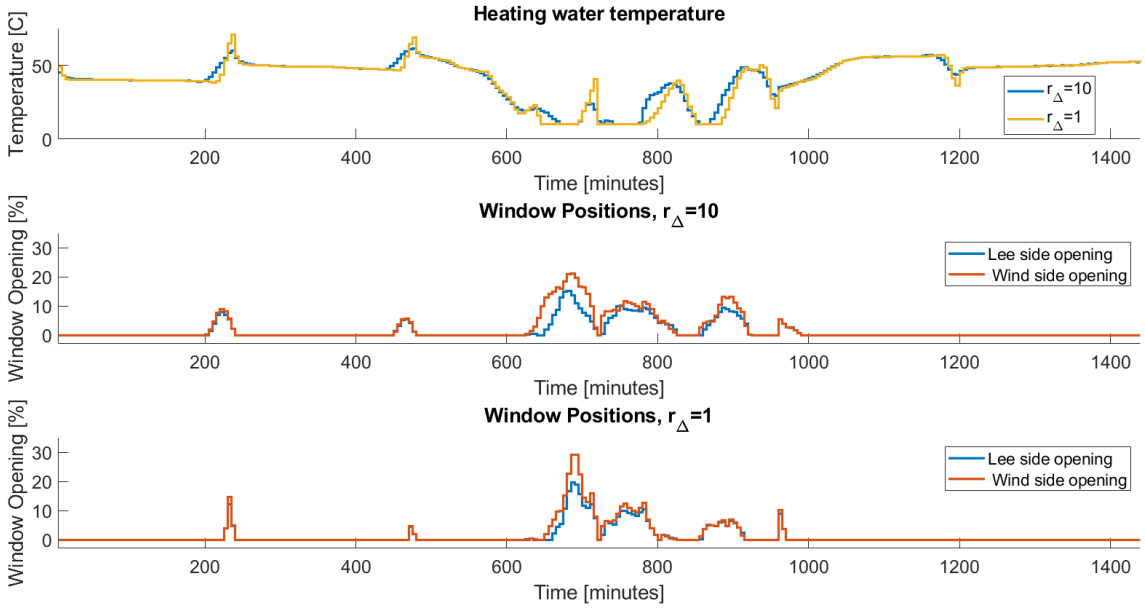

Figure 3-4: Control inputs computed by NMPC for different values of r_Δ. The top plot shows the heating water temperature for $r_\Delta = 10$ and $r_\Delta = 1$, respectively. The middle plot shows the lee and wind side window openings for $r_\Delta = 10$ and the bottom plot shows the lee and wind side opening for $r_\Delta = 1$.

3-3-3 Yield Maximization

Besides reference tracking to maintain an optimal temperature, another goal of NMPC might be to maximize the yield. Since we have access to both a climate and a crop model, these models might be used together to maintain an optimal growth climate while balancing the costs of resources and the gains of yield. Furthermore, the goal will be to keep the temperature within desired bounds. Hence, constraints will be placed such that the greenhouse air temperature state is constraint within those bounds. However, due to exogenous signals the system can be driven out of the desired range, which will result in a region where the NMPC problem is infeasible. Therefore, in order to avoid infeasibility problems, these constraints will be softened by using slack variables [32].

Therefore, the optimization problem becomes:

$$\begin{aligned} \min_{u} \quad & \sum_{k=0}^{N-1} -m_{\mathrm{F},k} + \|u_k\|_R^2 + \lambda_\epsilon \|\epsilon\|_1 \\ \text{subject to} \quad & x_{k+1} = f_{\mathrm{g}}(x_k, u_k), \quad \forall k \in \{0, ..., N-1\}, \\ & x_0 = \hat{x}(t), \\ & u_l \leq u_k \leq u_u, \quad \forall k \in \{0, ..., N-1\}, \\ & x_l - \epsilon \leq x_k^{\mathrm{Temp}} \leq x_u + \epsilon \quad \forall k \in \{0, ..., N-1\}, \\ & \epsilon \geq 0 \end{aligned} \tag{3-6}$$

Where $m_{\mathrm{F},k}$ denotes the fruit weight state at time index k. Furthermore, $\epsilon \in \mathbb{R}$ denotes a slack variable that represents the temperature constraint violation. Hence, if $\epsilon = 0$, the temperature constraints are satisfied. $\lambda_\epsilon \in \mathbb{R}$ denotes the penalty on the slack variable.

For this numerical result, the parameters are chosen as follows:
$N = 12$
$R = \mathrm{diag}(0.01, 0.01, 0.01, r_{\mathrm{CO}_2})$
$\lambda_\epsilon = 10^5$
$x_l = 15$
$x_u = 22$
$u_l = [10, 0, 0, 0]^T$
$u_u = [80, 100, 100, 2 * 10^{-3}]^T$

In the variables above, x_l and x_u are the lower and upper bounds on the states, respectively. Furthermore, r_{CO_2} denotes the penalty on the CO_2 injection control input. Two different numerical simulations are performed for both $r_{\mathrm{CO}_2} = 100$ and $r_{\mathrm{CO}_2} = 500$ in order to show the differences when the penalty on CO_2 injection is larger.

In Figure 3-5, Figure 3-6 and Figure 3-7 the results of the yield maximization simulation are shown. As can be seen from Figure 3-5, the temperature of the greenhouse air is kept between the desired bounds of 15 °C and 22°C. Since the CO_2 concentration and injection only effect the greenhouse crop and not the greenhouse temperature, the same greenhouse air temperature, heating pipe temperature, heating water temperature and window openings were realized for both cases of $r_{CO_2} = 100$ and $r_{CO_2} = 500$. Therefore, these are only shown once since they are similar for both cases of r_{CO_2}.

In Figure 3-6 can be seen that with higher CO_2 levels, more assimilates are produced, which is a logical cause since a higher CO_2 level enables a higher photosynthesis capacity. Furthermore, at the end of the day when the light levels decrease, the photosynthesis decreases as well. This causes in its turn a depletion of the assimilate buffer as all assimilates are used in order to from new biomass for the fruits, stems, leafs or for respiration. From the fruit weight plot can be seen that at the end of the day, the fruit weight is higher with a lower penalty on the CO_2 injection compared to the higher penalty.

Furthermore, from Figure 3-7 can be seen that CO_2 is dosed only in presence of solar radiation. Obviously, this makes sense since without solar radiation no photosynthesis is realized. However, from this figure can also bee seen that at the middle of the day when the light levels are high, the CO_2 dosage decreases. This can be clarified due to the fact that the CO_2 will go directly out of the window when the windows are open. The windows open in order to prevent too high temperatures when there is high radiation, therefore making it too costly to dose CO_2 when only a small part is used by the crop and a large part goes out of the window.

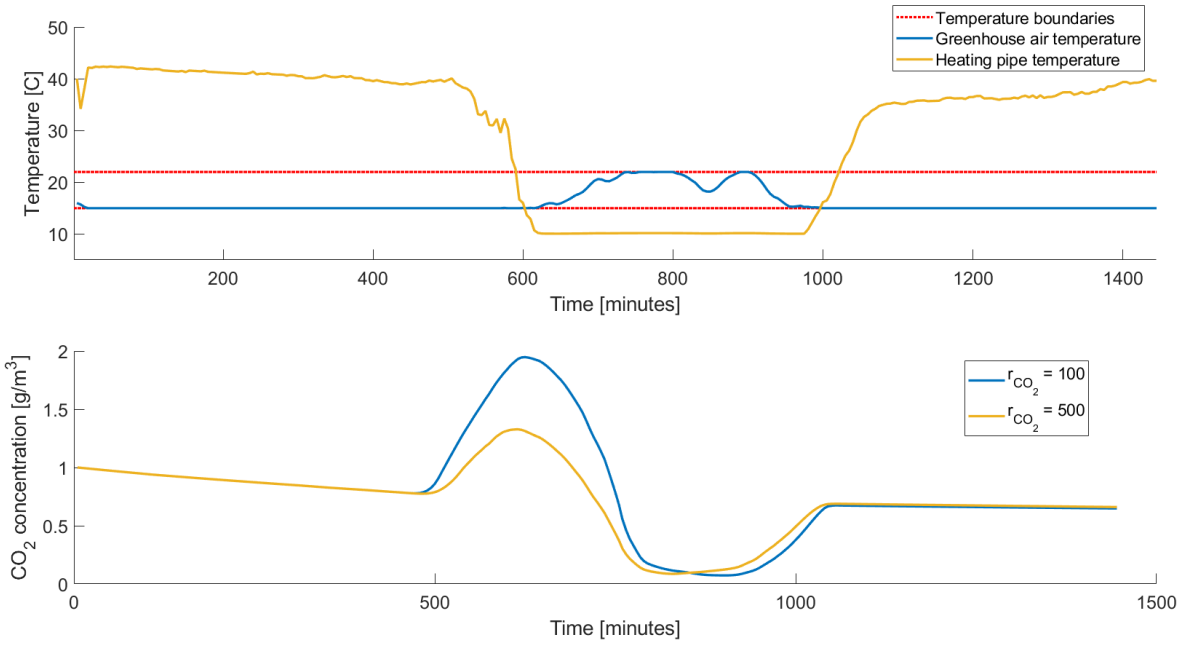

Figure 3-5: Greenhouse air and heating pipe temperatures (top) and greenhouse air CO_2 concentration (bottom) controlled by NMPC for $r_{CO_2} = 100$ and $r_{CO_2} = 500$.

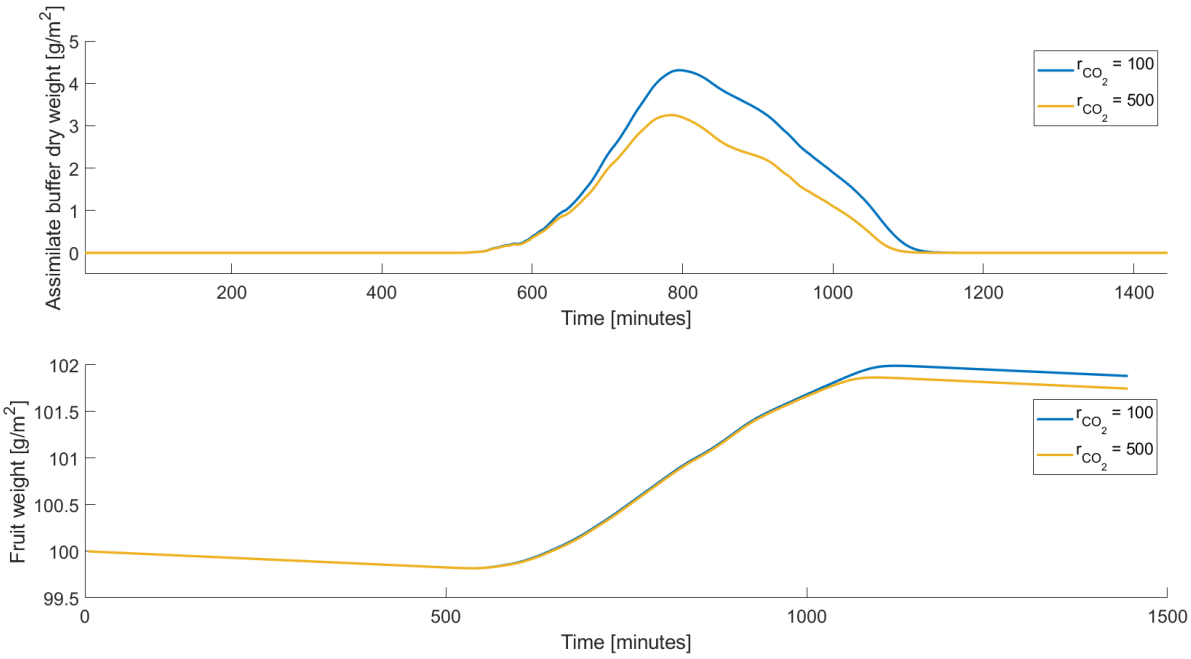

Figure 3-6: Assimilate buffer dry weight (top) and fruit dry weight (bottom) controlled by NMPC for $r_{CO_2} = 100$ and $r_{CO_2} = 500$.

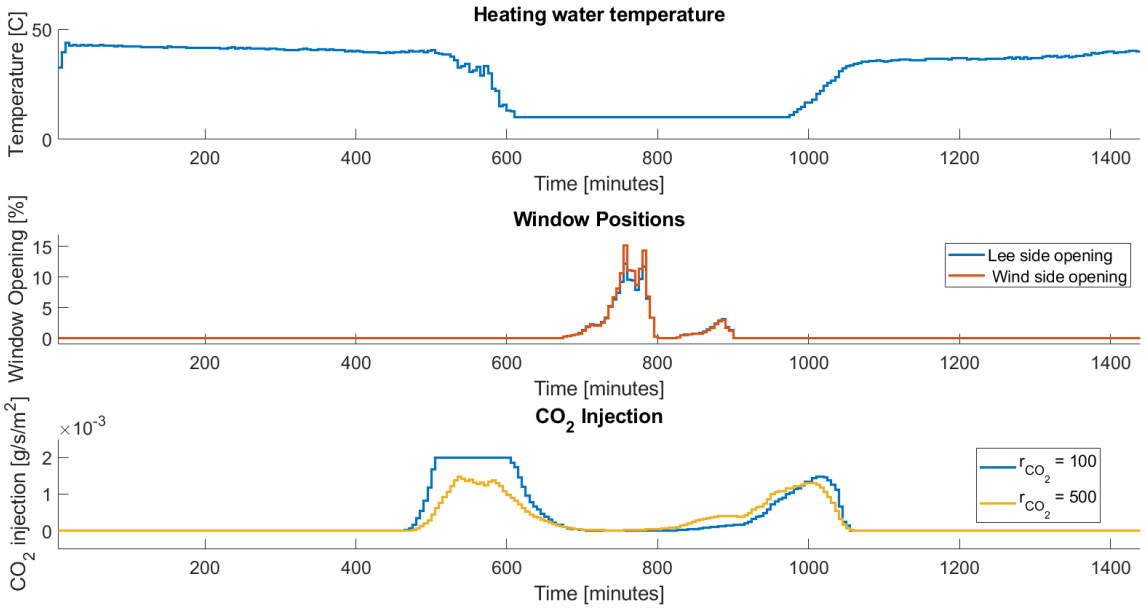

Figure 3-7: Control inputs computed by NMPC for different values of r_{CO_2}. The top plot shows the heating water temperature for both $r_{CO_2} = 100$ and $r_{CO_2} = 500$. The middle plot shows the window positions on lee and wind side for both $r_{CO_2} = 100$ and $r_{CO_2} = 500$. The bottom plot shows the CO_2 injection for both $r_{CO_2} = 100$ and $r_{CO_2} = 500$.

Chapter 4

Data-Enabled Predictive Control

4-1 Introduction

Data-Enabled Predictive Control (DeePC) is a predictive control algorithm that computes optimal control policies using real-time feedback driving the unknown system along a desired trajectory while satisfying system constraints [11]. The algorithm uses a finite number of data samples from the unknown system to learn a non-parametric system model that represents the dynamics of the system. This non-parametric system model is subsequently used to implicitly estimate the state and to predict future input/output trajectories of the unknown system.

Therefore, the DeePC algorithm replaces system identification, state estimation and trajectory prediction by one single optimization framework that determines the non-parametric model, implicitly estimates the state and optimizes the system trajectory over a future horizon.

The theory on exploiting a non-parametric model arises from behavioural systems theory. Therefore, the next section introduces a few preliminaries from behavioural systems theory on which the DeePC algorithm relies.

4-2 Behavioural Systems Theory

Within behavioural systems theory a dynamical system is described by its behaviour, i.e. the dynamical systems are defined by the subspace of the signal space in which trajectories of the system live. This is different from classical systems theory where a particular parametric system representation is used to describe the system. Often, system properties are defined on representation level and therefore these properties might be dependent on their particular representation. Hence, viewing a system from the behavioural systems theory perspective is a more general way in the sense that system properties can be defined in terms of the systems behaviour, independent of any representation. The following definitions describe a dynamical system and its properties from a behavioural systems theory perspective [11][33]:

Definition 4.1: A *dynamical system* is a 3-tuple $\Sigma = (\mathbb{Z}_+, \mathbb{W}, \mathscr{B})$ where $\mathbb{Z}_+$ is the *discrete-time axis*, $\mathbb{W}$ is the *signal space* and $\mathscr{B} \subseteq \mathbb{W}^{\mathbb{Z}_+}$ is the *behaviour*.

Definition 4.2: Let $\Sigma = (\mathbb{Z}_+, \mathbb{W}, \mathscr{B})$ be a dynamical system.

(i) A system $\Sigma = (\mathbb{Z}_+, \mathbb{W}, \mathscr{B})$ is said to be *linear* if the signal space $\mathbb{W}$ is a vector space and $\mathscr{B}$ is a linear subspace of $\mathbb{W}^{\mathbb{Z}_+}$.

(ii) A system $\Sigma = (\mathbb{Z}_+, \mathbb{W}, \mathscr{B})$ is said to be *time-invariant* if $\mathscr{B} \subseteq \sigma\mathscr{B}$ where σ is the forward shift operator: $(\sigma w)(t) := w(t+1)$ and $\sigma\mathscr{B} = \{\sigma w \mid w \in \mathscr{B}\}$.

(iii) A system $\Sigma = (\mathbb{Z}_+, \mathbb{W}, \mathscr{B})$ is said to be *complete* if:
$w\,|_{[t_0,t_1]} \in \mathscr{B}\,|_{[t_0,t_1]},\ \forall\ t_0, t_1 \in \mathbb{Z}_+,\ t_0 \leq t_1 \quad \Rightarrow \quad w \in \mathscr{B}$

The class of systems $(\mathbb{Z}_+,\mathbb{R}^{m+p},\mathscr{B})$ satisfying (i)-(iii) is denoted by $\mathcal{L}^{m+p}$, with $m, p \in \mathbb{Z}_+$. Any trajectory $w \in \mathscr{B}$ can be written as $w = \text{col}(u, y)$, where $\text{col}(u, y) := (u^T, y^T)^T$ [11].

Definition 4.3: A system $\mathscr{B} \in \mathcal{L}^{m+p}$ is *controllable* if for any two trajectories $w_1, w_2 \in \mathscr{B}$, there is a third trajectory $w \in \mathscr{B}$, such that $w_1(t) = w(t)$, $\forall\, t < 0$, and $w_2(t) = w(t)$, $\forall\, t \geq 0$.

Definition 4.4: Let $L, T \in \mathbb{Z}_{++}$ such that $T - L + 1 \geq mL$. The signal $u = \text{col}(u_1, ..., u_T) \in \mathbb{R}^{mT}$ is *persistently exciting of order L* if the Hankel matrix

$$\mathscr{H}_L(u) := \begin{pmatrix} u_1 & \cdots & u_{T-L+1} \\ \vdots & \ddots & \vdots \\ u_L & \cdots & u_T \end{pmatrix}$$

is full row rank.

The definitions above describe the non-parametric system representation. Multiple equivalent ways exist to represent a behavioural system in a parametric representation. One of them is the discrete time state space system (3-1) where $\mathscr{B} \in \mathcal{L}^{m+p}$ is represented by $\mathscr{B}(A, B, C, D) = \{\text{col}(u, y) \in (\mathbb{R}^{m+p})^{(\mathbb{Z}_+)} \mid \exists\ x \in (\mathbb{R}^n)^{(\mathbb{Z}_+)} \text{ s.t. } x(t+1) = Ax(t) + Bu(t),\ y(t) = Cx(t) + Du(t)\}$. The state space representation of smallest order is called a *minimal representation* of the system and the minimal order is denoted by $\mathbf{n}(\mathscr{B})$. Furthermore, denote the *lag* of a system $\mathscr{B} \in \mathcal{L}^{m+p}$ by $l(\mathscr{B})$. The lag is defined as the smallest integer $l \in \mathbb{Z}_{++}$ such that the observability matrix $\mathcal{O}_l(A, C) := \text{col}(C, CA, ..., CA^{l-1})$ has rank $\mathbf{n}(\mathscr{B})$. Finally, define the lower triangular Toeplitz matrix, denoted by $\mathscr{T}$, as follows:

$$\mathscr{T}_N(A, B, C, D) := \begin{pmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ CA^{N-2}B & \cdots & CB & D \end{pmatrix} \tag{4-1}$$

Now, two useful lemmas can be presented:

Lemma 4.1 ([34], Lemma 1): Let $\mathscr{B}(A,B,C,D)$ be a minimal state space representation of $\mathscr{B} \in \mathcal{L}^{m+p}$ and let $T_{\text{ini}}, T_f \in \mathbb{Z}_{++}$ with $T_{\text{ini}} \geq l(\mathscr{B})$ and $\text{col}(u_{\text{ini}}, u, y_{\text{ini}}, y) \in \mathscr{B}_{T_{\text{ini}}+T_f}$. Then there exists a unique $x_{\text{ini}} \in \mathbb{R}^{\mathbf{n}(\mathscr{B})}$ such that:

$$y_f = \mathcal{O}_{T_f}(A,C)x_{\text{ini}} + \mathscr{T}_{T_f}(A,B,C,D)u_f \tag{4-2}$$

Hence, if the window of initial system data $\text{col}(u_{\text{ini}}, y_{\text{ini}})$ is sufficiently long, the state x_{ini} to which the system is driven by the sequence of inputs u_{ini} is unique.

Lemma 4.2 ([35], Theorem 1): Consider a controllable system $\mathscr{B} \in \mathcal{L}^{m+p}$ and let $T, t \in \mathbb{Z}_{++}$ and $w = \text{col}(u,y) \in \mathscr{B}_T$. Furthermore, let u to be persistently exciting of order $t+\mathbf{n}(\mathscr{B})$. Then $\text{colspan}(\mathscr{H}_t(w)) = \mathscr{B}_t$.

Hence, the Hankel matrix, consisting of a finite amount of data samples, provides a way to replace the required model or the preliminary system identification procedure.

4-3 Data-Enabled Predictive Control

Data Collection DeePC is a data-driven control algorithm. Hence, the first step is to collect data. First there is assumed that the data is generated by an unknown controllable LTI system $\mathscr{B} \in \mathcal{L}^{m+p}$. Let $T, T_{\text{ini}}, T_f \in \mathbb{Z}_{++}$ such that $T \geq (m+1)(T_{\text{ini}} + T_f + \mathbf{n}(\mathscr{B})) - 1$. Then in an offline procedure a sequence of T inputs $u^d = \text{col}(u_1^d, ..., u_T^d) \in \mathbb{R}^{mT}$ is applied to the unknown system and the corresponding outputs $y^d = \text{col}(y_1^d, ..., y_T^d) \in \mathbb{R}^{pT}$ are collected. Here, the superscript d is used to denote the offline collected data. Next the data is separated into a past and future part:

$$\begin{pmatrix} U_p \\ U_f \end{pmatrix} := \mathscr{H}_{T_{\text{ini}}+T_f}(u^d), \quad \begin{pmatrix} Y_p \\ Y_f \end{pmatrix} := \mathscr{H}_{T_{\text{ini}}+T_f}(y^d) \tag{4-3}$$

where U_p consists of the first T_{ini} block rows of $\mathscr{H}_{T_{\text{ini}}+T_f}(u^d)$ and U_f consists of the last T_f block rows of $\mathscr{H}_{T_{\text{ini}}+T_f}(u^d)$ (similar for Y_p and Y_f). The past data matrices U_p and Y_p will be used to implicitly estimate the initial state whereas the future data matrices U_f and Y_f will be used to predict the future trajectories of the system. Below is shown in detail how the Hankel matrix $\mathscr{H}_{T_{\text{ini}}+T_f}(u^d)$ is divided in U_p and U_f (similar division holds for $\mathscr{H}_{T_{\text{ini}}+T_f}(y^d)$):

$$\left(\begin{array}{c} U_p \\ \hline U_f \end{array}\right) = \left(\begin{array}{cccc} u_1 & u_2 & \dots & u_{T-T_{ini}-T_f+1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{T_{\text{ini}}} & u_{T_{\text{ini}}+1} & \dots & u_{T-T_f} \\ \hline u_{T_{\text{ini}}+1} & u_{T_{\text{ini}}+2} & \dots & u_{T-T_f+1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{T_{\text{ini}}+T_f} & u_{T_{\text{ini}}+T_f+1} & \dots & u_T \end{array}\right) \tag{4-4}$$

Now using the result of Lemma 4.2: with the collected data, any trajectory of $\mathscr{B}_{T_{\text{ini}}+T_f}$ of length $T_{\text{ini}}+T_f$ could be constructed. It follows that a trajectory $\text{col}(u_{\text{ini}}, u_f, y_{\text{ini}}, y_f)$ belongs to $\mathscr{B}_{T_{\text{ini}}+T_f}$ if and only if there exists $g \in \mathbb{R}^{T-T_{\text{ini}}-T_f+1}$ such that:

$$\begin{pmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_f \\ y_f \end{pmatrix} \tag{4-5}$$

Now using the result of Lemma 4.1: if $T_{\text{ini}} \geq l(\mathscr{B})$, Lemma 4.1 implies that there exists a unique $x_{\text{ini}} \in \mathbb{R}^{\mathbf{n}(\mathscr{B})}$ such that y_f is uniquely determined by (4-2). Hence, when solving the first three block rows of (4-5) for g, a unique output y_f can be computed based on inputs u_f and the initial trajectory $\text{col}(u_{\text{ini}}, y_{\text{ini}})$.

Data-Enabled Predictive Control Next, the DeePC algorithm will be formulated. Consider the following optimal control problem:

$$\begin{aligned} \min_{g} \quad & \sum_{k=0}^{T_f-1} \left(\|y_{f,k} - r_{t+k}\|_Q^2 + \|u_{f,k}\|_R^2\right) \\ \text{subject to} \quad & \begin{pmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_f \\ y_f \end{pmatrix}, \\ & u_k \in \mathcal{U}, \quad \forall k \in \{0, ..., T_f - 1\}, \\ & y_k \in \mathcal{Y}, \quad \forall k \in \{0, ..., T_f - 1\}. \end{aligned} \tag{4-6}$$

Where $T_f \in \mathbb{Z}_{++}$ is the time horizon, $r = (r_0, r_1, ...) \in \mathbb{R}^{pT_f}$ is the output reference trajectory, $\text{col}(u_{\text{ini}}, y_{\text{ini}}) \in \mathscr{B}_{T_{\text{ini}}}$ is the past input and output data, $\mathcal{U} \subseteq \mathbb{R}^m$ is the input constraint set, $\mathcal{Y} \subseteq \mathbb{R}^p$ is the output constraint set. Furthermore, $\|u_k\|_R^2$ denotes $u_k^T R u_k$ (similar for $\|.\|_Q$), where $R \in \mathbb{R}^{m \times m}$ denotes the positive semi-definite control cost matrix and $Q \in \mathbb{R}^{p \times p}$ denotes the positive semi-definite output cost matrix.

The optimization problem in (4-6) is solved at every time step $t \in \mathbb{Z}_+$. Furthermore, in Algorithm 3 the DeePC algorithm is shown.

Algorithm 3: DeePC

Input: $\text{col}(u^d, y^d) \in \mathscr{B}_T$, reference trajectory $r \in \mathbb{R}^{pT_f}$, past input/output data $\text{col}(u_{\text{ini}}, y_{\text{ini}}) \in \mathscr{B}_{T_{\text{ini}}}$, constraint sets $\mathcal{U}$ and $\mathcal{Y}$ and cost matrices Q and R;

1) Solve (4-6) for $g^\star$.
2) Compute the optimal input sequence $u^\star = U_f g^\star$.
3) Apply input $u(t), ..., u(t+s) = (u_0^\star, ..., u_s^\star)$ for some $s \leq N-1$.
4) Set t to $t+s$ and update u_{ini} and y_{ini} to the T_{ini} most recent input/output measurements.
5) Return to 1.

In Figure 4-1, a graphical representation is shown on how the data of the different input/output trajectories is used. First, the data which captures the system dynamics is collected during a trajectory of length T. Then, a trajectory of length T_{ini} is measured to implicitly estimate the initial state of the system. This is done by solving the equality constraints $U_p g = u_{\text{ini}}$ and $Y_p g = y_{\text{ini}}$. Finally, an input/output trajectory of length T_f is predicted by $u_f = U_f g$ and $y_f = Y_f g$, respectively. Hence, the vector g contains the optimization variables which are used to fix the system dynamics, the state estimation and trajectory prediction.

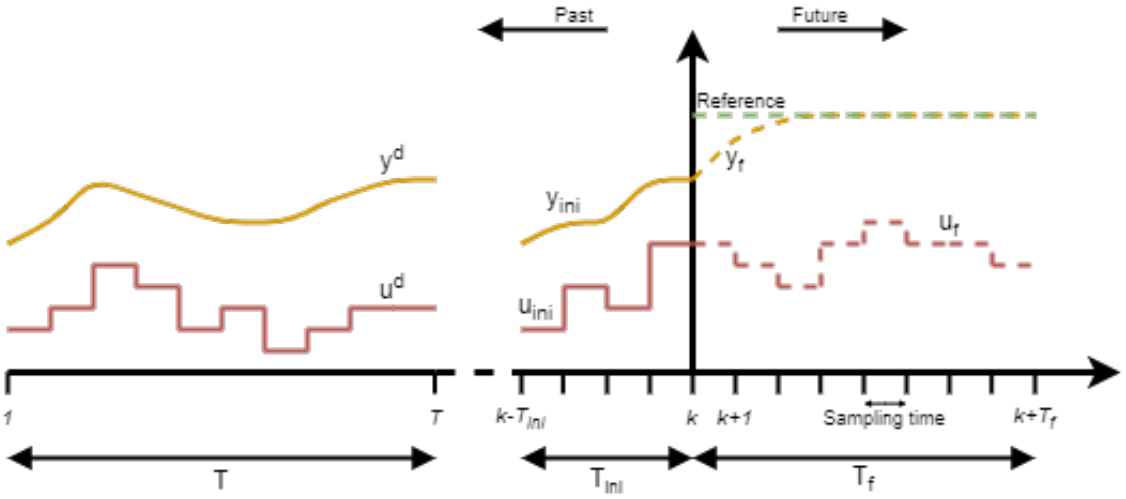

Figure 4-1: DeePC scheme

4-4 Greenhouse Controlled by DeePC

4-4-1 Including Exogenous Signals

In this section, the DeePC algorithm is used to control the greenhouse system. Since the exogenous inputs such as solar radiation and outside temperature have large effects on the states of the greenhouse, theses exogenous input signals need to be taken into account as well besides only the control input and output signals. Hence, the DeePC algorithm in (4-6) needs to be extended in order to include these exogenous signals. In case of the greenhouse system, the exogenous inputs are the outside weather conditions. Hence, these are measured in the past and forecast are available for the future. Therefore, a Hankel matrix is constructed in the same way as is done in (4-3) but it is build from the exogenous signal data. Hence, given the recorded data $v^d = \text{col}(v_1, ..., v_T) \in \mathbb{R}^{qT}$ a Hankel matrix is constructed such that:

$$\mathscr{H}_{T_{\text{ini}}+T_f}(v) := \begin{pmatrix} v_1 & \cdots & v_{T-T_{\text{ini}}-T_f+1} \\ \vdots & \ddots & \vdots \\ v_{T_{\text{ini}}+T_f} & \cdots & v_T \end{pmatrix} \tag{4-7}$$

Subsequently, this matrix is separated into a past and a future part similar to (4-3):

$$\begin{pmatrix} V_p \\ V_f \end{pmatrix} := \mathscr{H}_{T_{\text{ini}}+T_f}(v^d) \tag{4-8}$$

where V_p consists of the first T_{ini} block rows of $\mathscr{H}_{T_{\text{ini}}+T_f}(v^d)$ and V_f consists of the last T_f block rows of $\mathscr{H}_{T_{\text{ini}}+T_f}(v^d)$. In the same way as is done in (4-5), the past T_{ini} measurements of the external signals are stored in $\text{col}(v_{\text{ini}}) \in \mathbb{R}^{qT_{\text{ini}}}$ and the future T_f forecasts of the external signals are stored in $v_f \in \mathbb{R}^{qT_f}$. The equality constraints in (4-5) are then augmented with the aforementioned data matrices and vectors such that:

$$\begin{pmatrix} U_p \\ V_p \\ Y_p \\ U_f \\ V_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{\text{ini}} \\ v_{\text{ini}} \\ y_{\text{ini}} \\ u_f \\ v_f \\ y_f \end{pmatrix} \tag{4-9}$$

4-4-2 Extension to Non-Linear Systems

Furthermore, since the dynamics of the greenhouse system are non-linear, the equation $Y_p g = y_{\text{ini}}$ might become infeasible in non-linear regions of the system dynamics. Therefore, the constraint is softened by using a slack variable to allow constraint violation. Besides, due to the stochastic nature of the exogenous signals, the equations $V_P g = v_{\text{ini}}$ and $V_f g = v_f$ might become infeasible as well. Therefore, these equations are relaxed with a slack variable in the same way in order to avoid possible infeasibilities. Hence, the constraints in (4-9) are extended with auxiliary slack variables $\sigma_y \in \mathbb{R}^{pT_{\text{ini}}}$, $\sigma_{v_1} \in \mathbb{R}^{qT_{\text{ini}}}$ and $\sigma_{v_2} \in \mathbb{R}^{qT_f}$:

$$\begin{pmatrix} U_p \\ V_p \\ Y_p \\ U_f \\ V_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{\text{ini}} \\ v_{\text{ini}} \\ y_{\text{ini}} \\ u_f \\ v_f \\ y_f \end{pmatrix} + \begin{pmatrix} 0 \\ \sigma_{v_1} \\ \sigma_y \\ 0 \\ \sigma_{v_2} \\ 0 \end{pmatrix} \tag{4-10}$$

4-4-3 Window Constraint

When cooling the greenhouse, the wind side window has a larger effect on the ventilation rate compared to the lee side window. Hence, it might be desirable to keep the wind side window opening at a relative lower opening compared to the lee side window. Therefore, an inequality constraint will be added in order to preserve the difference between the lee side and wind side window openings:

$$U_{f,\text{lee}} g \geq 2 U_{f,\text{wind}} g \tag{4-11}$$

Here, $U_{f,\text{lee}}$ and $U_{f,\text{wind}}$ denote the rows of the U_f Hankel matrix that correspond to the lee side and wind side window openings, respectively. Furthermore, this additional constraint also decreases the computation time since it limits the search space when optimizing over g.

4-4-4 Regularized DeePC

Now, including the constraints formulated in (4-10) and (4-11), we arrive at the following optimization problem:

$$\begin{aligned}
\min_{g} \quad & \sum_{k=0}^{T_f-1} (\|y_{f,k} - r_{t+k}\|_Q^2 + \|u_{f,k}\|_R^2) + \lambda_y \|\sigma_y\|_1 + \lambda_v \|\sigma_v\|_1 \\
\text{subject to} \quad & \begin{pmatrix} U_p \\ V_p \\ Y_p \\ U_f \\ V_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{\text{ini}} \\ v_{\text{ini}} \\ y_{\text{ini}} \\ u \\ v \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ \sigma_{v_1} \\ \sigma_y \\ 0 \\ \sigma_{v_2} \\ 0 \end{pmatrix}, \\
& U_{f,\text{lee}} g \geq 2U_{f,\text{wind}} g, \\
& \sigma_{v_1} \geq 0, \\
& \sigma_{v_2} \geq 0, \\
& \sigma_y \geq 0, \\
& u_k \in \mathcal{U}, \quad \forall k \in \{0, ..., T_f - 1\}.
\end{aligned} \tag{4-12}$$

Here, $\lambda_y \in \mathbb{R}_+$ and $\lambda_v \in \mathbb{R}_+$ are regularization parameters on the constraint violations and $\sigma_v = [\sigma_{v_1}^T, \sigma_{v_2}^T]^T$.

The optimization problem in (4-12) is subsequently solved at each time step $t \in \mathbb{Z}_+$. Furthermore, in Algorithm 4 the extended regularized DeePC algorithm is shown.

Algorithm 4: Extended regularized DeePC

Input: $\text{col}(u^d, v^d, y^d) \in \mathscr{B}_T$, reference trajectory $r \in \mathbb{R}^{pT_f}$, past input/output data $\text{col}(u_{\text{ini}}, v_{\text{ini}}, y_{\text{ini}}) \in \mathscr{B}_{T_{\text{ini}}}$, constraint sets $\mathcal{U}$ and $\mathcal{Y}$ and output cost matrix Q and control cost matrix R;

1) Solve (4-12) for $g^\star$.
2) Compute the optimal input sequence $u^\star = U_f g^\star$.
3) Apply input $u(t), ..., u(t+s) = (u_0^\star, ..., u_s^\star)$ for some $s \leq T_f - 1$.
4) Set t to $t+s$ and update u_{ini}, v_{ini} and y_{ini} to the T_{ini} most recent input/output measurements.
5) Return to 1.

4-4-5 Reference Tracking with DeePC

Here, a numerical simulation is shown where the DeePC algorithm is used to control the greenhouse system (2-43).

As described above, first data needs to be collected in order to construct the Hankel matrices. In order to ensure the control input data to be persistently exciting the system, the control inputs will be sampled as random variables from a uniform distribution, see Figure 4-2. The heating water temperature and the window openings are drawn from a uniform distributions

with a minimum and maximum values of [20,80] and [0,20], respectively. The input/output trajectory that was collected is of length $T = 144$, i.e., 12 hours of data in this case.

Furthermore, $T_{\text{ini}} = 5$ and $T_f = 12$, i.e., the past 5 data samples are used to implicitly estimate the state and a prediction horizon of 12 samples (1 hour) is used. Within Figure 4-2 is indicated which part of the data is used in the Hankel matrix U_p and which data is used in the Hankel matrix U_f. Furthermore, R is fixed at $\text{diag}(0.1, 5, 5)$ and λ_v is fixed at $= 10^4$.

In Figure 4-3 the results of the simulations are shown for different values of λ_y and Q. Within this plots, the temperatures predicted by the DeePC algorithm and the realized temperatures by the model are compared together with the reference that needs to be tracked. Here, the predicted temperature is the first step ahead prediction computed by the DeePC algorithm, i.e., the first entry of the vector $Y_f g^\star \in \mathbb{R}^{pT_f}$. The simulated temperature is the output from the model after applying the first predicted control input computed by the DeePC algorithm, i.e., the first m entries of the vector $U_f g^\star \in \mathbb{R}^{mT_f}$.

From this figures can be seen that a high value for Q (bottom) results in good reference tracking for the model but a large mismatch in the model and prediction outputs. For a high value of λ_{ini} is shown that there is a low mismatch between the model and prediction outputs but the reference tracking accuracy is low. Hence, a trade-off needs to be made between a reliable match of the predicted and model outputs and desirable reference tracking by varying λ_y and Q.

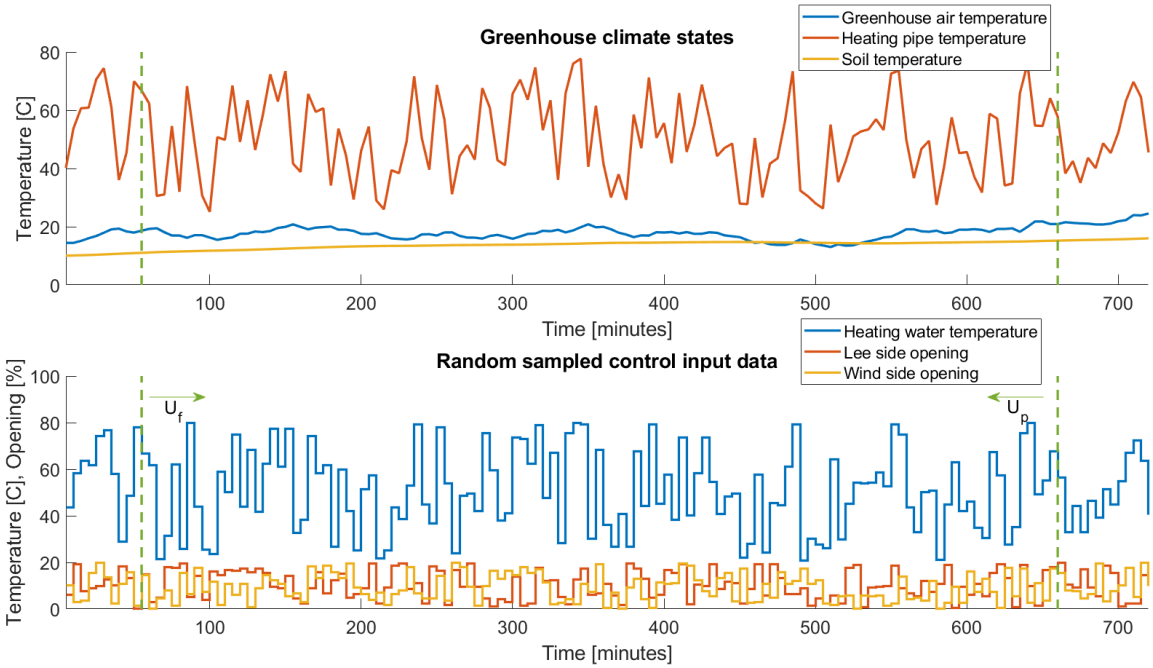

Figure 4-2: Random generated control inputs (bottom) and corresponding outputs (top). The green dashed lines and arrows indicate which part of the data is stored in U_p and U_f (similar for Y_p and Y_f).

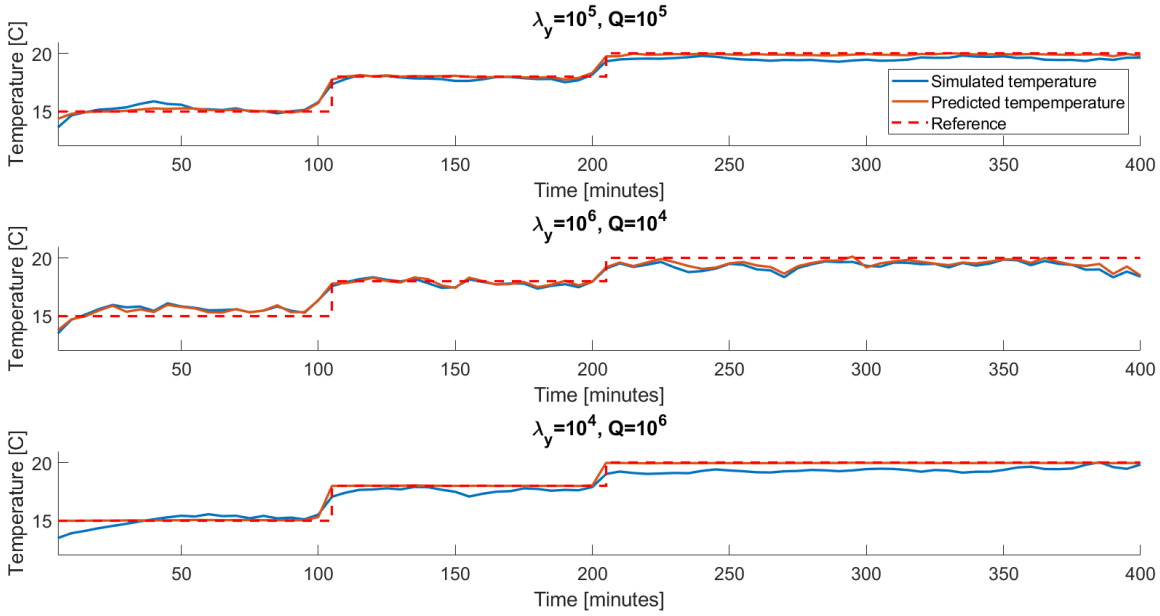

Figure 4-3: Model temperature and predicted temperature for different values of λ_y and Q.

Chapter 5

Case Study: NMPC vs DeePC

5-1 Introduction Case-Study

In this chapter a case study will be performed in order to compare the Nonlinear Model Predictive Control (NMPC) and Data-Enabled Predictive Control (DeePC) algorithms. The objective for both algorithms will be to track a temperature reference with the greenhouse air temperature on a similar day. From this day, the recorded weather conditions will be used as exogenous inputs. Afterwards, both algorithms will be compared on tracking accuracy and computational time.

In the previous chapter, the input/output data that is used in the Hankel matrices (4-5) in the DeePC algorithm is randomly generated. However, in practice it is often not desired or even impossible to apply a random generated control input signal to a system. In case of the greenhouse system, the greenhouse crop is sensitive to rapid changes in greenhouse air temperature which occur by applying a random control inputs, see Figure 4-2. Therefore, a control input sequence will be generated using NMPC such that a 'safe' input can be applied to the greenhouse in order to capture its dynamics.

Furthermore, the importance of the representativeness of this data will be shown in this chapter. In another example, the DeePC algorithm will be leveraged twice in order to control the greenhouse system. In the first case, the Hankel matrices will be constructed from input/output data where the crop is still young. In the second case, the control input and exogenous signals will be the same, however, the system will be initialized such that the output represents the dynamics of the greenhouse with a full grown crop. For both cases, a reference tracking example will be given. Afterwards, the tracking accuracy and computational time of both these simulations will be shown as well.

The next two subsections show the recorded weather conditions and NMPC control input data that is used within this case-study.

5-1-1 Weather Profile

Similar recorded weather conditions will be used in order to compare both algorithms. These weather conditions are shown in Figure 5-1. This data will be used to construct the matrices V_p and V_f in (4-5). Besides, in this figure is indicated which part of the data is used to construct the data matrices.

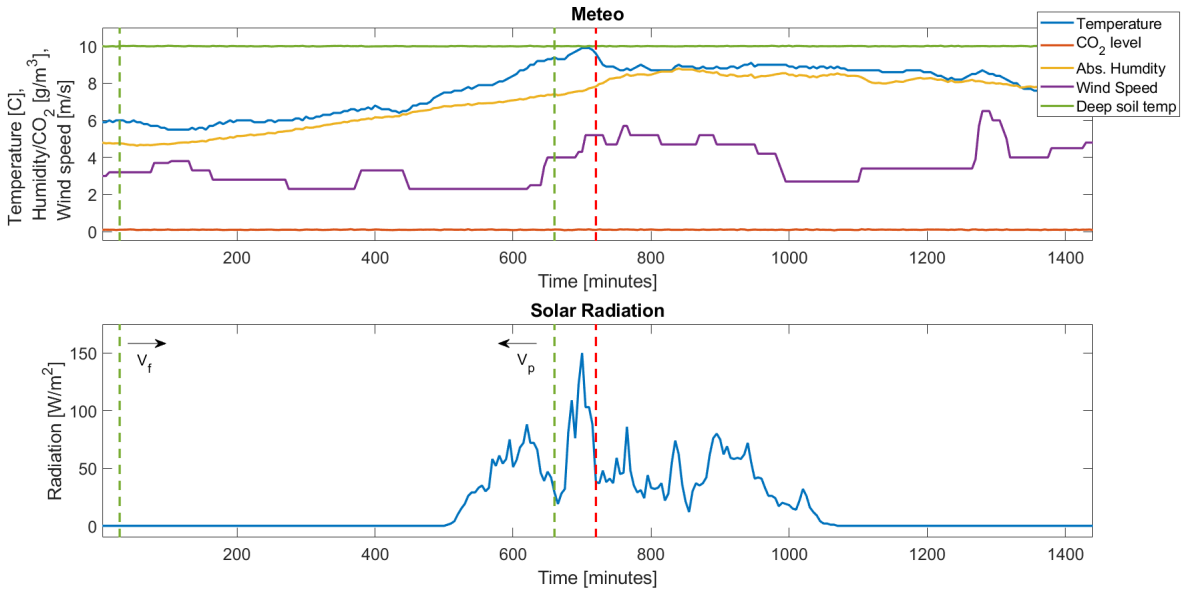

Figure 5-1: Measured weather conditions used in the DeePC algorithm.

5-1-2 Control Input Signal

In Figure 5-2 the NMPC control input signal is shown that is used to construct the Hankel matrices U_p and U_f. In this figure is indicated which part is used in the U_p matrix (used for capturing the dynamics of the system) and which part is used in the U_f matrix (used for predictions of the future trajectory). This control input signal is constructed while using the recorded weather data shown in Figure 3-2. Besides, in this figure is indicated which part of the data is used to construct the data matrices. A necessary condition for (4-12) to be feasible is that U_p has full rank. After checking it's rank using Matlab it turned out to be full rank. Hence, the control input data in Figure 5-2 seems to be appropriate as control input data for the DeePC algorithm.

Now with both the recorded weather data shown in Figure 5-1 and the control input signal shown in Figure 5-2, the data is available to generate the output data signals (not shown here) to construct the final two Hankel matrices Y_p and Y_f.

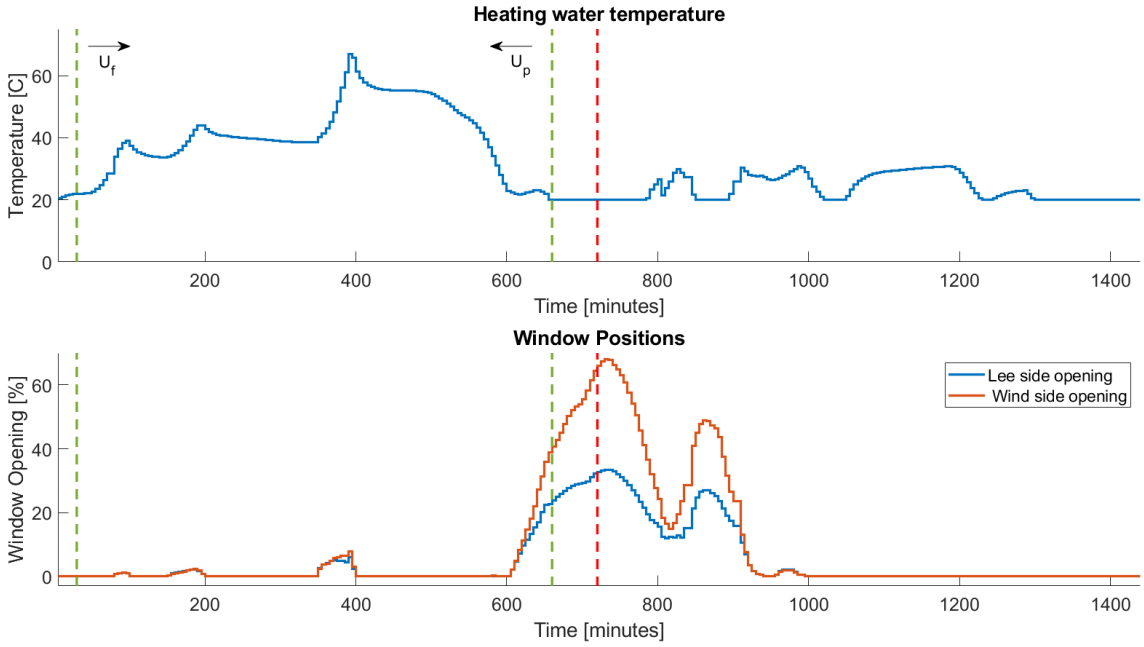

Figure 5-2: Generated NMPC input signals used in the DeePC algorithm.

5-2 Controlling the Greenhouse

Now all the data necessary is available, both algorithms can be used to track a temperature reference. The reference is constructed in such a way that the 24 hour average temperature is approximately 21 °C, which is an average also occurring in commercial practice. Furthermore, the minimum and maximum temperatures are chosen such that they are safe for the crop and the energy of the sun can be exploited by using a higher temperature in the middle of the day and lower temperature at the start and end of the day.

Since a half day of measurements with 5 minute sampling is used to construct the Hankel matrices $T = 144$. Furthermore, $T_{\text{ini}} = 5$, i.e., the last 5 samples are used to implicitly estimate the system state and $T_f = 12$, i.e., the prediction horizon over which the future trajectory of the system is predicted is 12 samples (1 hour). Other parameters that are used in the DeePC algorithm are: $Q = \text{diag}(10^5, 0, ..., 0)$, $R = \text{diag}(0.1, 5, 5)$, $\lambda_v = 10^4$, $\lambda_\epsilon = 10^5$, $u_l = [10, 0, 0]^T$ and $u_u = [80, 100, 100]^T$.

For the NMPC algorithm, the following parameters are used: $N = 12$, $Q = \text{diag}(500, 0, ...0,)$, $R = 0.1I_3$, $\Delta_R = I_3$, $u_l = [10, 0, 0]^T$ and $u_u = [80, 100, 100]^T$.

In Figure 5-3 and Figure 5-4, the states and control inputs controlled and computed by both NMPC and DeePC are shown. From this figures can be seen that both algorithms are able to accurately track the desired reference. Furthermore, a noticeable difference is that the NMPC algorithm returned smoother control inputs signals compared to the DeePC algorithm.

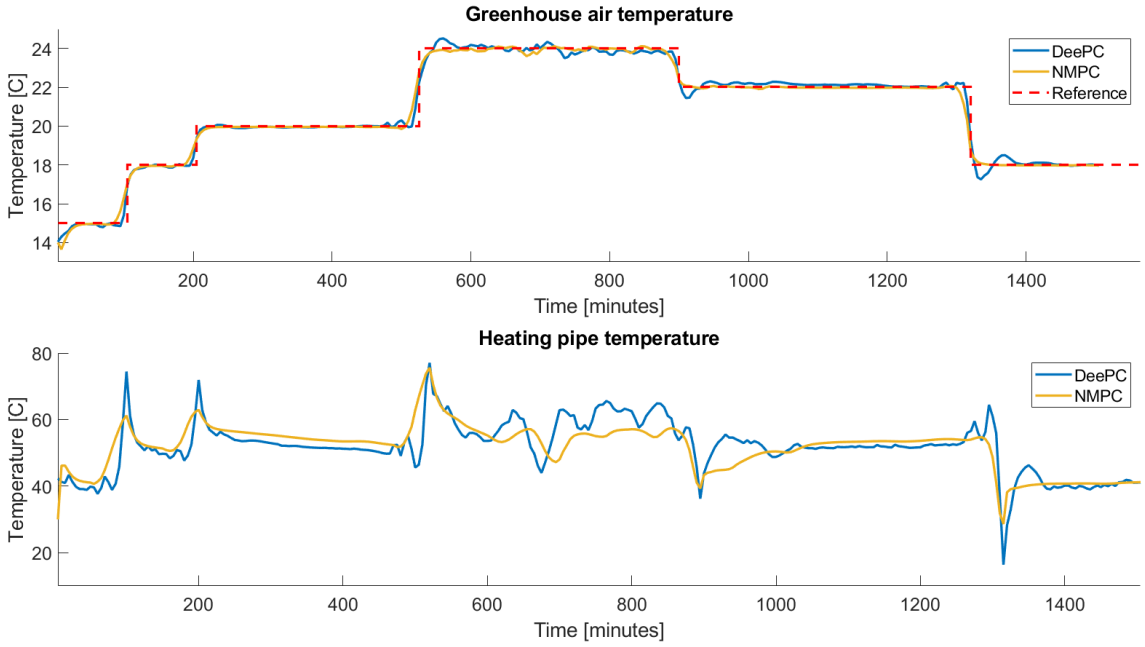

Figure 5-3: Greenhouse air and heating pipe temperatures controlled by both DeePC and NMPC.

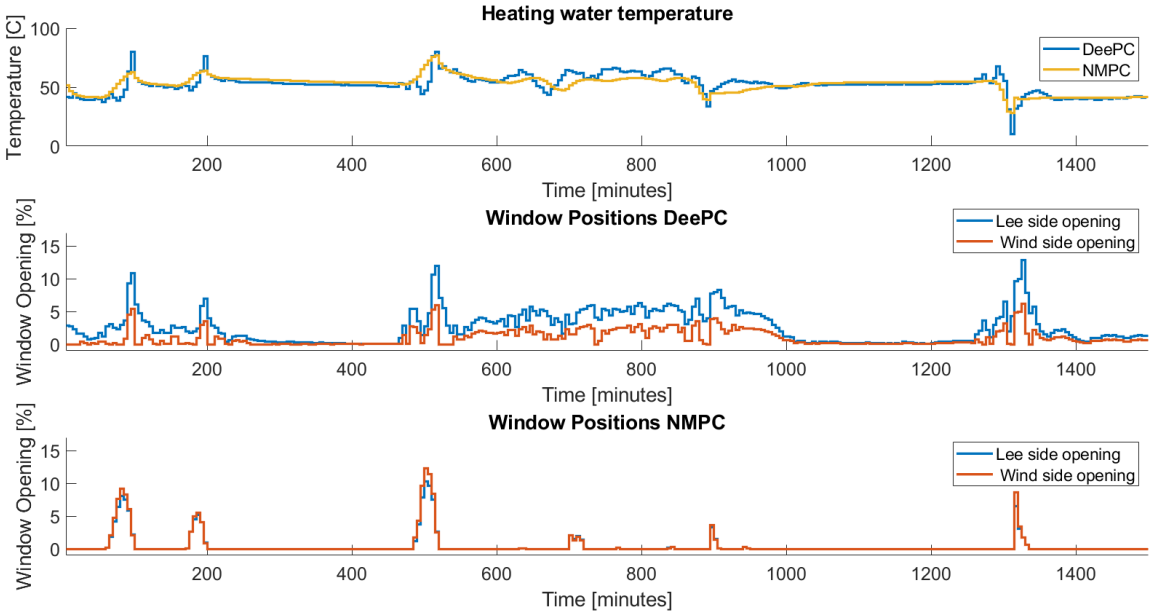

Figure 5-4: Heating water temperature and window positions computed by both DeePC and NMPC.

5-3 Comparison of Algorithm Performance

In Table 5-1 the tracking accuracy, average computation time and total computation time are shown for both the NMPC and DeePC algorithms. The Root Mean Squared Error (RMSE) is used in order to make a comparison on tracking accuracy. From this table can be seen that the NMPC algorithm has a higher reference tracking accuracy compared to the DeePC algorithm. Furthermore, the average computation time for NMPC is approximately 30 times smaller. However, since the sampling time is 5 minutes in this case, the DeePC algorithm would still be able to control the greenhouse in real-time.

Table 5-1: Comparison of DeePC and NMPC on tracking accuracy and computation time.

	NMPC	DeePC
Tracking accuracy [RMSE]	0.0461	0.2572
Average computation time [seconds]	0.5347	15.425
Total computation time [minutes]	160.42	4642.9

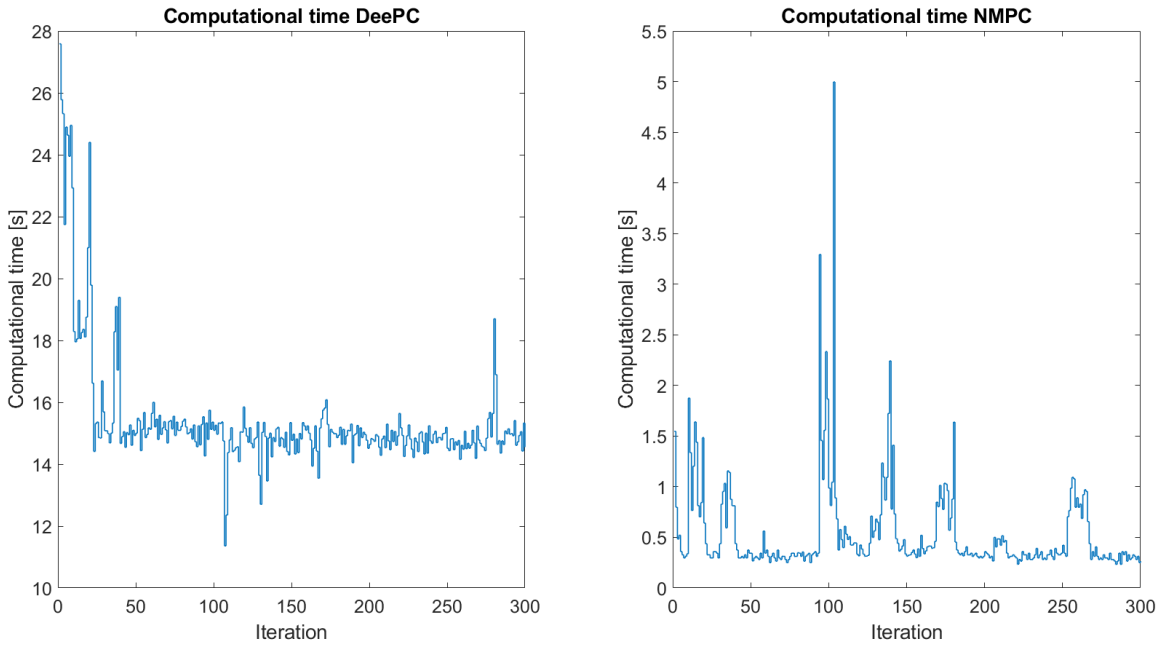

Figure 5-5: Computation time per iteration for both DeePC and NMPC.

5-4 Adjusted Greenhouse Dynamics

Here, another example will be given where two numerical simulations will be performed. In the first simulation the input/output data in the Hankel matrices will be the same as in the previous example. This data was obtained in a period where the crop was still young. However, within this simulation the initial state of the crop will be adapted in such a way that it represents a full grown crop; similar to the simulation example in Chapter 1. Hence, this will influence the dynamics of the greenhouse climate while these effects are not present in the data stored in the Hankel matrices. This will be done in order to show the sensitivity of the DeePC algorithm when data is used that is less representative for the current dynamics of the system. Subsequently, in the second simulation the Hankel matrices will be updated with data from the greenhouse where the crop is fully developed. With these updated data matrices the algorithm is used again to control the greenhouse climate in order to show that the DeePC algorithm is again able to control the system while the dynamics have changed. Within these examples, the same values as in Section 5-2 are used.

In Table 5-2, the tracking accuracy, average and total computation time of both simulations are shown. From this table can be seen that the tracking accuracy of the simulation with the new data is comparable with the tracking accuracy from the simulation in the previous example. However, the RMSE in case of the simulation with the old data has almost doubled compared to the case with the simulation with the new data. Hence, this shows the effect of using the old data matrices instead of the updated data matrices. The old data does not accurately represent the dynamics of the system anymore whereas the updated data does. The computation time of both numerical simulations here is roughly equal.

In Figure 5-6 and Figure 5-7 the results from these numerical simulations are shown. As can be seen from this figures the DeePC algorithm is sensitive to time-varying systems when the data in the Hankel matrices becomes less representative for the current greenhouse dynamics. However, when the data is updated with more recent input/output data, the algorithm is able to recover from this and control the system accordingly. In these figures the simulation of the DeePC algorithm with the data of the young crop is indicated as 'DeePC; Old data' and the simulation with the updated data is indicated as 'DeePC; New data'.

Table 5-2: Comparison of the two DeePC simulations on tracking accuracy and computation time.

	DeePC; Old data	DeePC; New data
Tracking accuracy [RMSE]	0.8045	0.3800
Average computation time [seconds]	9.2296	9.7268
Total computation time [minutes]	46.1479	48.6340

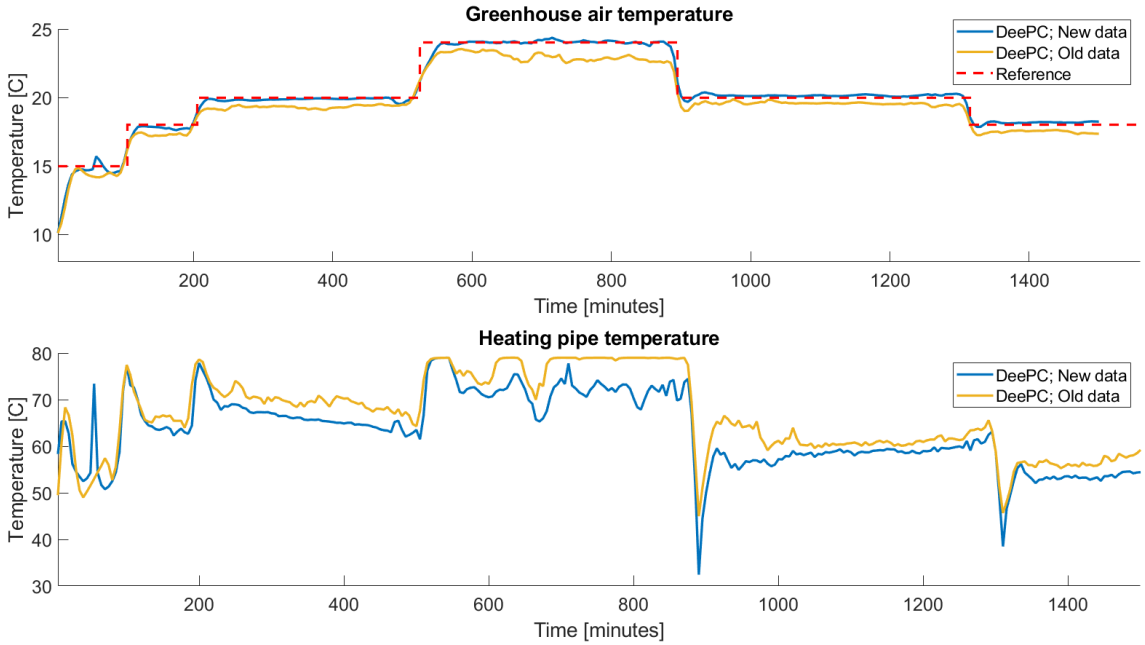

Figure 5-6: Greenhouse air and heating pipe temperatures controlled by DeePC based on both the old and updated data.

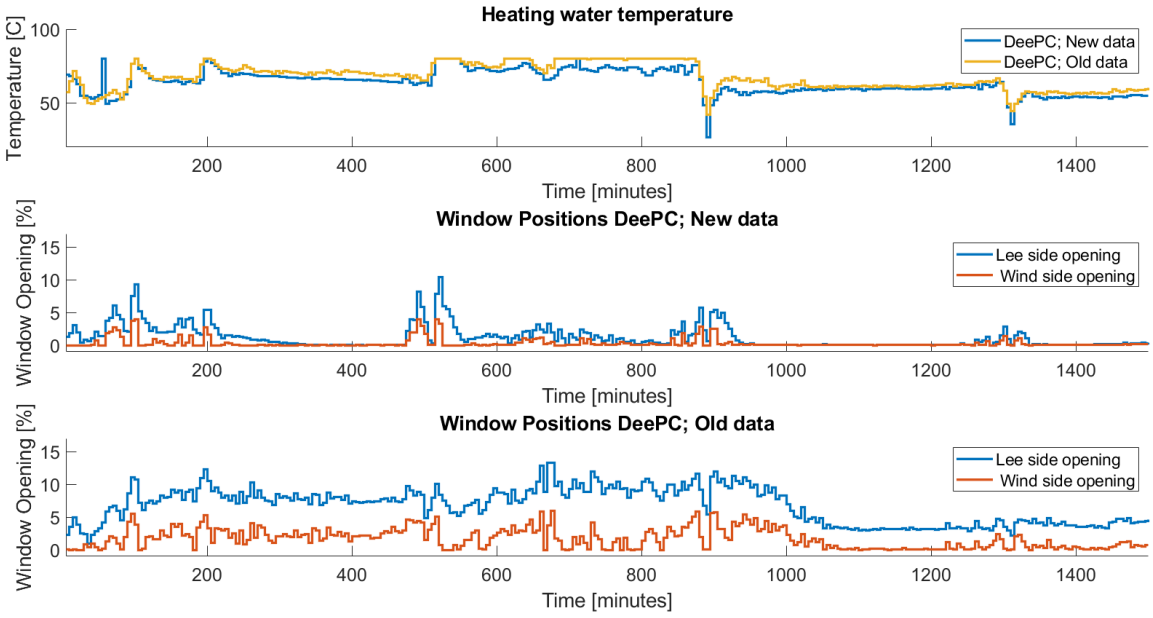

Figure 5-7: Heating water temperature and window positions computed by DeePC based on both the old and updated data.

Chapter 6

Irrigation Control

6-1 Introduction

The greenhouse model used in this thesis assumes that the crop is irrigated properly. This is a common assumption in many crop models. In commercial practice, irrigation if often done by an experienced grower who irrigates the crop based on their own strategy and experience.

Nowadays, greenhouses are often equipped with sensors which measure characteristics in the substrate slab such as the Water Content (WC), Electrical Conductivity (EC) and temperature of the slab. Hence, this data contains a lot of information about the water uptake of the crop, e.g., when the WC decreases. Therefore, in this chapter an irrigation control algorithm is proposed. This algorithm relies on these WC measurements and on the total solar radiation sum received on a daily base. Since solar radiation has a major influence on the water uptake of the crop, solar radiation was chosen as the quantity to relate with the WC measurements. Based on these measurements, a relation is derived between the total radiation sum and the gradient of the water content graph after the last irrigation.

6-2 Strategy

In this section, an irrigation strategy that is applied in commercial growing practice will be explained. First, at sunrise, irrigation is provided until the slab reaches its saturation point, i.e., when drain is realized. After this point this saturation level is maintained by applying a constant interval time such that the WC level remains constant at the saturation level. Then, the goal is to dry-down the slab a certain percentage. This is done since during the night the crop does not use the water. Therefore, drying down the slab prevents rotting in the root zone and it allows the slab to refill with oxygen again. However, growers determine when to stop the irrigation based on their experience. In order to automatize the irrigation, it is necessary to predict what the dry-down is going to be during the remaining part of the day and during the night, which will be explained in the next section.

6-3 Control Algorithm

Now, let Δ_L denote the dry-down rate during the remaining part of the day when it is light and let Δ_D denote the dry-down rate when it is dark, see Figure 6-2. In order to make predictions for these dry-down rates, historical data is collected from which dry-down rates of previous days are derived. Furthermore, for each day the corresponding total radiation sum was collected as well. Now, with both the dry-down rates and radiation sum available, a correlation can be made between dry-down rate and amount of radiation received. The correlation with radiation is made since radiation has a major influence on the water uptake of the crop.

Now, with the necessary collected data, predictions for the dry-down rates Δ_L and Δ_D can be made. This is done by fitting a linear relationship between the radiation sums and the dry-back rates to the data such that with the forecast of radiation sum, the corresponding prediction can be computed, see Figure 6-1. When fitting the linear regression, only the data points of the latest τ days are taken into account. This is done in order to account for the growth stage of the crop. Since a young crop takes up far less water compared to a full grown crop, the water uptake rate is much lower for young crops. Hence, if only the latest e.g., 14 days are taken into account, the data will implicitly account for the growth stage of the crop.

Now let $\hat{\Delta}_L$ and $\hat{\Delta}_D$ denote the predictions of the dry-down rate during the light and dark period, respectively, and let ξ_{dd} denote the total desired dry-down [%] that needs to be achieved to maintain optimal substrate slab conditions. Furthermore, let t_L and t_D denote the time in hours of the remaining light part of the day and the dark part, respectively. The total dry-down of a day is then be computed by:

$$\text{Total dry-down} = \Delta_L t_L + \Delta_D t_D \tag{6-1}$$

The time during the dark period t_D is often known in advance since the sunrise/sunset times and the time the lights switch on/off are available. Hence, when the estimates for the dry-down are available, the desired dry-down can be calculated as follows:

$$\xi_{dd} = \hat{\Delta}_L t_L + \hat{\Delta}_D t_D \tag{6-2}$$

In (6-2), the only unknown quantity is now the time length of the light period. Hence, re-arranging terms yields:

$$t_L = \frac{1}{\hat{\Delta}_L}(\xi_{dd} - \hat{\Delta}_t t_D) \tag{6-3}$$

Now both t_L and t_D are available, the total time that is necessary to dry-down the desired amount of WC can be calculated as the sum of t_L and t_D. With this information, let T_{stop} denote the time to stop the irrigation. Then in algorithm 5, the irrigation algorithm is shown.

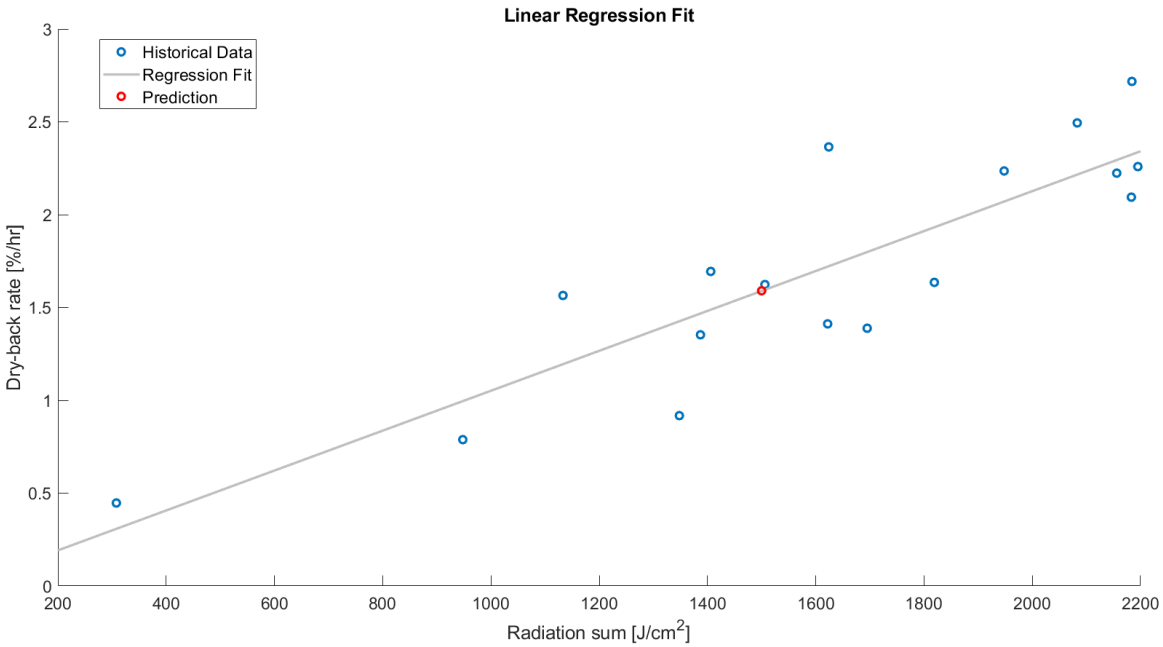

Figure 6-1: Linear regression fit between the radiation sum received and dry-down rate during the light period Δ_L with the dry-back rate prediction for 1500 J/cm^2 superimposed.

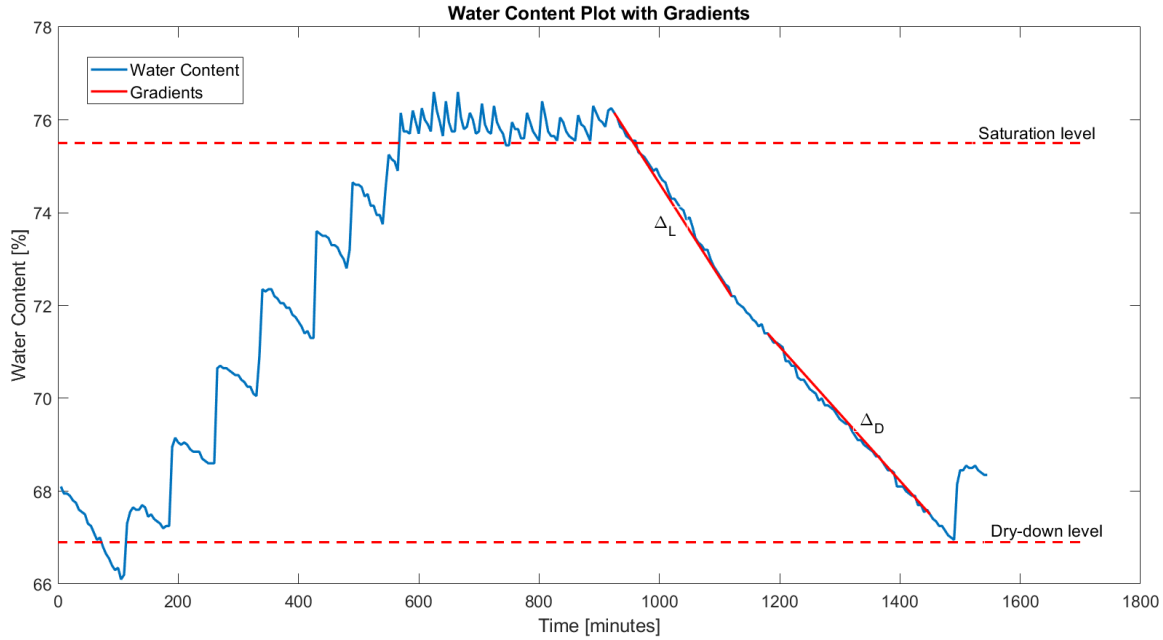

Figure 6-2: Water content measurements during a particular day with the gradients Δ_L and Δ_D superimposed.

Algorithm 5: Irrigation algorithm

Data: Dry-down target ξ_{dd}, τ data points of [Radiation sum, Δ_L] and [Radiation sum, Δ_D], drain and WC measurements.

```
while Drain ≤ 0% do
    Irrigate every hour;
    if Drain > 0% then
        Set: Setpoint = Saturation level;
        while Time < T_stop do
            Calculate T_stop using (6-3);
            if Measured WC < Setpoint then
                Supply irrigation shot
```

Chapter 7

Conclusions and Recommendations

7-1 Conclusion

7-1-1 Summary

In this thesis, a data-driven predictive control algorithm, Data-Enabled Predictive Control (DeePC), was employed in order to control a greenhouse system. The greenhouse system consisted of both a greenhouse climate and a greenhouse crop model, implemented in Matlab. The performance of the DeePC algorithm was compared to a model based control algorithm (Nonlinear Model Predictive Control (NMPC)), in a case-study where the greenhouse climate needed to be controlled. The NMPC algorithm had direct access to a greenhouse simulation model whereas the DeePC algorithm only relied on a non-parametric model which was built from data from this greenhouse simulation. The goal for both algorithms was to control the greenhouse air temperature by following a temperature reference.

In numerical simulation it was shown that the DeePC algorithm was able to follow the reference temperature. The performance in terms of tracking accuracy was lower for the DeePC algorithm, which is not surprising since the NMPC algorithm had direct access to the greenhouse simulation model. Hence, NMPC could directly see the effects of the control inputs over which it optimized whereas DeePC could only rely on the dynamics of the greenhouse captured in the data that was used to construct the Hankel matrices. Although the computation time of DeePC was on average 30 time larger compared to NMPC, it is in simulation still able to control the greenhouse system in real-time. Besides, DeePC replaces system identification, state estimation and trajectory prediction with one single optimization framework, making (usually time consuming) system identification and state observer design superfluous.

Furthermore, it has been shown that the DeePC algorithm would automatically adapt to the changing behaviour of the greenhouse climate through a growing cycle. This was done in numerical simulation by providing two simulations with the exact same control and exogenous input data but with different output data due to a different crop stage. The simulation which used the most recent data in the Hankel matrices showed that it accounted for the adapted behaviour and controlled the system accordingly.

7-1-2 Answers to Research Questions

The research questions as stated in Chapter 1 are answered in this subsection. The following sub-research questions are listed and answered below:

Which model should be used to include in the NMPC algorithm and to generate data for the DeePC algorithm?

- The model from [16] was used in this thesis. It concerns a 9-state time-varying non-linear greenhouse climate and crop model. The continuous time model was discretized and subsequently implemented in Matlab. This model was used as simulator in order to generate data for the DeePC algorithm and included as model in the NMPC algorithm.

Which of the available actuators are going to be controlled and which inputs could be retrieved from auxiliary controllers or forecasts?

- The control inputs that were used are the heating water temperature, the window openings on both the lee and wind side and the CO_2 injection. Furthermore, the exogenous signals were retrieved through LetsGrow.com and consisted of recorded weather data.

How can feedback of the crop status be included in the control loop?

- In the NMPC yield maximization case, the weight of the crop was directly taken into account in the cost function. Hence, the algorithm optimized the yield while taking into account the resource usage. In the case of DeePC the growth stage of the crop was implicitly taken into account since it was captured in the data used by this algorithm.

How is the performance of the DeePC algorithm going to be bench-marked against the NMPC algorithm?

- Both algorithms were compared on reference tracking accuracy in terms of Root Mean Squared Error (RMSE) and on computation time. This gave an indication on how accurate both algorithms were able to control the greenhouse climate with respect to each other and how computationally demanding both algorithms are.

Below, the main research question is answered:

How does Data-Enabled Predictive Control perform compared to Nonlinear Model Predictive Control when controlling an (autonomous) greenhouse?

- In numerical simulation has been shown that the DeePC algorithm is able to control the greenhouse climate by learning the dynamics of the greenhouse solely from input/output data. Furthermore, it was shown in numerical simulation that the DeePC algorithm was able to adapt to the changing behaviour through the growing cycle. The NMPC algorithm performed better in terms of reference tracking accuracy since it had direct access to the greenhouse model whereas the DeePC algorithm only relied on input/output data from this model. The DeePC algorithm has larger computation time compared to the NMPC algorithm but could still control the greenhouse simulation in real-time.

7-1-3 Autonomous Greenhouse Challenge

As mentioned in the introduction, the Autonomous Greenhouse Challenge, organized by Wageningen University and Research and Tencent, was a challenge where a greenhouse compartment needed to be remotely controlled for a growing period of 6 months. After a 24 hour hackathon in which 21 teams participated, 5 teams were selected to participate in the real Challenge. These 5 teams were given the opportunity to take part in the growing experiment to remotely control one of the greenhouse compartments for 6 months. Furthermore, a sixth reference team consisting of experienced growers got access to a compartment as well such that a comparison could be made between the remotely controlled greenhouses and 'commercial practice'. The goal of this Challenge was to produce a cherry tomato crop with a high level production and a high resource use efficiency. Furthermore, the quality of the tomatoes played an important role as well. Together with fellow team members from the Technical University of Delft, Hoogendoorn Growth Management, Van der Hoeven Horticultural Projects and Keygene we formed team Automatoes and participated during this international Challenge.

The participating teams in the Challenge were evaluated based on three factors: the obtained net profit (50%), sustainability (20%) and their Artificial Intelligence (AI) strategy (30%). The net profit factor was calculated by multiplying the weight of the harvested fruits with a price based on fruit quality and subtracting the costs of electricity, heating, CO_2 usage, labour and the plants itself. The sustainability factor was based on CO_2, water, nutrients, heat and electricity usage per kilogram of production. Finally, the AI strategy was presented for an international jury consisting of experts from both the horticultural and computer science domain. This jury reviewed the presentation on four components: novelty, robustness, scalability and functionality and combined those elements into one score for the AI strategy.

During this Challenge it was attempted to use the DeePC algorithm for control of the greenhouse climate. Unfortunately, due to the time-varying delays in input/output data and weather forecasts it was not feasible to test it for a long period during the real Challenge. Furthermore, the developed irrigation algorithm explained in Chapter 6 was used for several consecutive days in the Challenge. Also, its predictions were used as guideline on other days when the algorithm was not running and the stopping time of the irrigation needed to be determined. It proved that it was able to outperform a highly experienced world-class grower. Furthermore, in this Challenge I was responsible for drafting slides for the final presentation that accounted for 30% of the final score, participating in the weekly discussions on strategic choices and hacking our way through the 24 hour hackathon at the beginning of the Challenge.

Finally, with this multidisciplinary team we managed to maintain a high level of production of excellent quality tomatoes throughout the Challenge while having a high resource use efficiency. Furthermore, the Challenge was approached in a modular way, i.e., for different parts of the climate control, separate AI strategies were developed. Besides an innovative approach in control, some unprecedented choices concerning crop handling and stem density were made as well. Combining all these innovative algorithms and strategies with excellent teamwork, resulted in the first place for team Automatoes in the Autonomous Greenhouse Challenge, see Figure 7-1 and Figure 7-2. Our team managed to finish with the highest score possible in all three categories, which maximally displayed our great efforts and teamwork!

Points given by jury:

	Net profit	Sustainability	AI strategy	Total
Automatoes	25	10	15	50
AiCU	20	8	12	40
DIGILOG	15	6	3	24
IUA.CAAS	10	4	6	20
The Automators	5	2	9	16

Figure 7-1: Final results of the Autonomous Greenhouse Challenge based on net profit, sustainability and AI strategy. Team Automatoes ranked first in all categories [36].

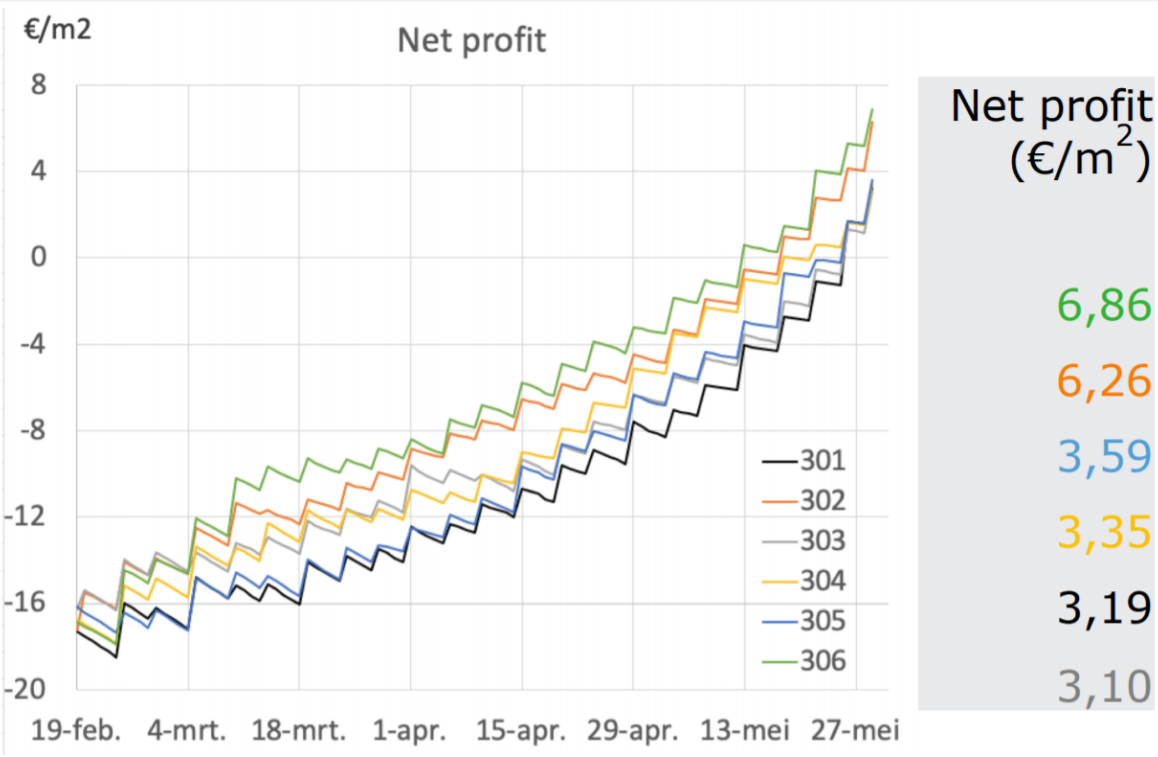

Figure 7-2: The net profit realized by all teams. Team Automatoes controlled compartment 306 (green). The reference team controlled compartment 303 (gray). The remaining colors are the other AI teams [36].

7-2 Recommendations for Future Work

Some points that are addressed for future work are listed below:

Time-Varying Systems The DeePC algorithm as used in this thesis exploits a fixed period of measured input/output data to represent the system dynamics. However, when a system exhibits time-varying dynamics, the recorded data that is used to construct the Hankel matrices should be updated regularly in order to ensure the representative of the current system dynamics. During this thesis, a method to account for time-varying dynamics, Online DeePC, was proposed in [37]. Hence, a topic for future work could be to use the method from [37] in order to account for time-varying system dynamics.

Persistence of Excitation The quality of the solution of the optimization problem (4-12) largely depends on the data that is used to construct the Hankel matrices. Hence, the input/output data collected during regular operation might not be persistently exciting the system or it might even be impossible to collect persistently exciting control input data during regular operation due to physical limitations. Therefore, a topic for future research would be to investigate how well a sequence of control input data excites the system and how well this captures the system dynamics, especially for non-linear systems.

Tuning of Parameters The DeePC algorithm has many different tuning and decision parameters. Within this thesis values for these parameters are used which proved to work for the these data sets and this simulation model. Due to long computational times it was not suitable to test a wide range of possible combinations of parameters. Hence, it might be recommended to develop some sort of automated tuning algorithm that is able to find a set of the most suitable tuning parameters according to a certain metric, e.g., RMSE.

Greenhouse Model The greenhouse model used in this thesis describes the main processes that happen in almost every modern greenhouse. However, even more state-of-the-art greenhouses nowadays are also equipped with more advanced tools such as a fogging system, active cooling and multiple screens. Hence, this increases the complexity of the greenhouse dynamics. Therefore, a topic for future work could be to improve the greenhouse model in such a way that more advanced equipment is modeled. Furthermore, a more advanced crop model that would for example have daily state updates based on e.g., daily radiation sums or daily average temperatures would be more suitable when controlling the crop states with DeePC.

Irrigation Now, the irrigation algorithm correlates the dry-down rates with the daily radiation sum. By taking only the most recent data points, the growth stage is implicitly taken into account as well. However, a point for future work could be to improve the algorithm by investigating the influence of other factors on the dry-down rate such as substrate slab temperature, greenhouse air temperature, radiation intensity, plant load or evaporation energy on the dry-down rate. In this way, more reliable predictions of the dry-down rates could be accomplished, leading to an improved performance of the irrigation control and a reduction in water usage.

Appendix A

Notation

This Appendix gives a description of the notation used in this thesis.

The following notation is used for the (sub-)sets of real numbers: $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}_+$ denotes the set of non-negative real numbers and $\mathbb{R}_{++}$ denotes the set of positive real numbers.

Similar, the following notation is used for the (sub-)sets of integers: $\mathbb{Z}$ denotes the set of integers, $\mathbb{Z}_+$ denotes the set of non-negative integers and $\mathbb{Z}_{++}$ denote the set of positive integers.

Furthermore, $\mathbb{R}^n$ denotes the real numbered n-dimensional vector (similar for $\mathbb{Z}$) and $\mathbb{R}^{n\times n}$ denotes the real numbered n-by-n matrix (similar for $\mathbb{Z}$). The transpose of a matrix $A \in \mathbb{R}^{m\times n}$ is denoted by $A^T \in \mathbb{R}^{n\times m}$.

Moreover, $\|x\|_2$ denotes the 2-norm on a vector $x \in \mathbb{R}^n$. $\|x\|_Q^2$ denotes the scalar x^TQx for a vector $x \in \mathbb{R}^n$ and a matrix $Q \in \mathbb{R}^{n\times n}$. The inequalities $Q \succeq 0$ and $Q \succ 0$ indicate that a matrix $Q \in \mathbb{R}^{n\times n}$ is positive semi-definite and positive definite, respectively.

Finally, I_n denotes the n-by-n identity matrix and $\text{diag}(1, 2, ...n) \in \mathbb{R}^{n\times n}$ denotes the following diagonal matrix:

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & n \end{bmatrix} \tag{A-1}$$

Appendix B

Greenhouse Simulation Model Constants

Table B-1: Greenhouse model parameters

Symbol	Description	Value	Unit
β	Heat absorption efficiency	0.01	[-]
γ	Apparent psychometric constant	0.067	kPa °C
ϵ	Cover heat resistance ration	3	[-]
ζ	Ventilation rate parameter	$2.7060*10^{-5}$	[-]
η	Radiation conversion factor	0.7	[-]
θ	Ventilation rate parameter	$4.02*10^{-5}$	[-]
κ	Ventilation rate parameter	$5.03*10^{-5}$	[-]
Λ	Pressure constant	0.46153	$Nm°C^{-1}g^{-1}$
μ	Molar weight fraction CO2 CH2O	1.4667	[-]
v	Ventilation rate parameter	$3.68*10^{-5}$	$\%^{-1}$
ξ	Ventilation rate parameter	$6.3233*10^{-5}$	$\%^{-1}$
ρ_w	Density of water	998	kg m^{-3}
ρ_a	Density mass of air	1.29	kg m^{-3}
σ	Ventilation rate parameter	$7.1708*10^{-5}$	$\%^{-1}$
τ	Pipe heat transfer coefficient	3	$°C^{-1/2}$
ν	Pipe heat transfer coefficient	0.74783	$Wm^{-2}°C^{-3/4}$
χ	Ventilation rate parameter	0.0156	$\%^{-1}$
ψ	Ventilation rate parameter	$7.4*10^{-5}$	m s^{-1}
ω	Humidity ratio parameter	0.622	[-]
a_1	Saturated vapour pressure parameter	0.611	kPa
a_2	Saturated vapour pressure parameter	17.27	[-]
a_3	Saturated vapour pressure parameter	239	°C
A_p	Heating pipe outer surface area	314.16	m^2

Table B-2: Greenhouse model parameters, continued

Symbol	Description	Value	Unit
b_1	Buffer switching function coefficient	2.7	m^2g^{-1}
C_g	Greenhouse heat capacity	$32*10^3$	$J°C^{-1}m^{-2}$
C_p	Specific heat of water at constant pressure	1010	$J°C^{-1}kg^{-1}$
C_s	Greenhouse soil heat capacity	$120*10^3$	$J°C^{-1}m^{-2}$
c_p	Air specific heat of water at constant pressure	1010	$J°C^{-1}kg^{-1}$
d_1	Plant development rate parameter	$2.1332*10^{-7}$	s^{-1}
d_2	Plant development rate parameter	$2.4664*10^{-1}$	s^{-1}
d_3	Plant development rate parameter	20	°C
d_4	Plant development rate parameter	$7.4966*10^{-11}$	[-]
f	Fruit assimilate requirement coefficient	1.2	[-]
f_1	Fruit growth rate coefficient	$8.1019*10^{-7}$	s^{-1}
f_2	Fruit growth rate coefficient	$4.6296*10^{-6}$	s^{-1}
g_1	Leaf conductance parameter	20.3	mm s^{-1}
g_2	Leaf conductance parameter	0.44	[-]
g_3	Leaf conductance parameter	$2.5*10^{-3}$	s $m^2\mu mol^{-1}$
g_4	Leaf conductance parameter	$3.1*10^{-4}$	m^3g^{-1}
g_b	Boundary layer conductance	10	mm s^{-1}
k_d	Soil to soil heat transfer coefficient	2.0	$W°C^{-1}m^{-2}$
k_r	Roof heat transfer coefficient	7.9	$W°C^{-1}m^{-2}$
k_s	Soil to air heat transfer coefficient	5.75	$W°C^{-1}m^{-2}$
l_1	Vaporisation energy coefficient	$2.501*10^6$	J g^{-1}
l_2	vaporisation energy coefficient	$2.381*10^3$	J $g^{-1}°C^{-1}$
m_1	Mass transfer parameter	$1.0183*10^{-3}$	g $s^{-1}m^{-2}$
m_2	Mass transfer parameter	0.33	[-]
M_{CO_2}	Molar mass CO_2	0.0044	kg mol^{-1}
M_F	Fruit maintenance respiration coefficient	$1.157*10^{-7}$	s^{-1}
M_L	Vegetative maintenance respiration coefficient	$2.894*10^{-7}$	s^{-1}
m_p	Watt to μmol conversion factor	4.57	μmol J^{-1}
p_1	Net-photosynthesis parameter	577	μmol $s^{-1}m^{-2}$
p_2	Net-photosynthesis parameter	221	ppm
p_{atm}	Atmospheric pressure	101.0	kPa
P_m	Maximum photosynthesis rate	$2.2538*10^{-3}$	g s^{-1} m^{-2}
Q_G	Fruit growth rate parameter Q_{10}-value	2.0	[-]
Q_R	Maintenance respiration Q_{10}-value	2.0	[-]
R_g	Gas constant	8.3144	J $mol^{-1}K_{-1}$

Table B-3: Greenhouse model parameters, continued

Symbol	Description	Value	Unit
s_1	Saturated water vapour pressure curve slope parameter	1.8407^{-4}	kPa°C^{-3}
s_2	Saturated water vapour pressure curve slope parameter	$9.7838*10^{-4}$	kPa°C^{-2}
s_3	Saturated water vapour pressure curve slope parameter	0.051492	kPa°C^{-1}
T_0	Conversion factor ° to K	273.15	°C
T_d	Deep soil temperature	10.0	°C
T_R	Maintenance respiration reference temperature	25.0	°C
v	Vegetative assimilate requirement coefficient	1.23	[-]
v_1	Vegetative fruit growth ratio parameter	1.3774	[-]
v_2	Vegetative fruit growth ratio parameter	-0.168	°C
v_3	Vegetative fruit growth ratio parameter	19.0	°C
V_p	Heating pipe volume	7.85	m^3
V_g/A_g	Average greenhouse height	10.0	m
w_R	LAI correction function parameter	32.23	g m^{-2}
y_F	Fruit harvest coefficient parameter	0.5983	[-]
y_L	Fruit harvest coefficient parameter	0.5983	[-]
z	Leaf fraction of vegetative dry weight	0.6081	[-]

Bibliography

[1] United Nations, "World urbanization prospects: The 2018 revision," tech. rep., United Nations, Department of Economic and Social Affairs, Population Division, New York, 2019.

[2] FAO, IFAD, UNICEF, WFP, and WHO, "The state of food security and nutrition in the world 2018," tech. rep., FAO, Rome, 2018.

[3] G. van Straten and E. J. van Henten, "Optimal greenhouse cultivation control: Survey and perspectives," *IFAC Proceedings Volumes*, vol. 43, no. 26, pp. 18 – 33, 2010.

[4] Eurostat, "Agri-environmental indicator - energy use." https://ec.europa.eu/eurostat/statistics-explained/index.php/Agri-environmental_indicator_-_energy_use, January 2018. [Online; accessed 19-Sep-2019].

[5] M. van Eerdt, "Doel co2-emissie meerjarenafspraak glastuinbouw (2020) waarschijnlijk niet binnen bereik." https://themasites.pbl.nl/balansvandeleefomgeving/jaargang-2018/themas/landbouw-en-voedsel/emissie-broeikasgassen-glastuinbouw-2020, January 2018. [Online; accessed 13-Nov-2019].

[6] D. Brian, "What is the current state of labor in the greenhouse industry?." https://www.greenhousegrower.com/management/what-is-the-current-state-of-labor-in-the-greenhouse-industry/, November 2018. [Online; accessed 17-Sep-2019].

[7] C. Van Rijswijk, "World vegetable map 2018. raboresearch food agribusiness." https://research.rabobank.com/far/en/sectors/regional-food-agri/world_vegetable_map_2018.html, January 2018. [Online; accessed 17-Sep-2019].

[8] R. Shamshiri, F. Kalantari, K. Ting, K. Thorp, I. Hameed, C. Weltzien, *et al.*, "Advances in greenhouse automation and controlled environment agriculture: A transition to plant factories and urban agriculture," *International Journal of Agricultural and Biological Engineering*, vol. 11, no. 1, pp. 1–22, 2018.

[9] S. Hemming, F. Zwart, A. Elings, I. Righini, and A. S. Petropoulou, "Remote control of greenhouse vegetable production with artificial intelligence - greenhouse climate, irrigation, and crop production," *Sensors*, vol. 19, no. 8, pp. 1–22, 2019.

[10] S. Zeng, H. Hu, L. Xu, and G. Li, "Nonlinear adaptive PID control for greenhouse environment based on rbf network," *Sensors*, vol. 12, no. 5, pp. 5328–5348, 2012.

[11] J. Coulson, J. Lygeros, and F. Dörfler, "Data-Enabled Predictive Control: In the Shallows of the DeePC," *arXiv e-prints*, p. arXiv:1811.05890, Nov 2018.

[12] M. Waldrop, "Autonomous vehicles: No drivers required," *Nature*, vol. 518, pp. 20–23, Feb 2015.

[13] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, and Y. Wang, "Artificial intelligence in healthcare: past, present and future," *Stroke and Vascular Neurology*, vol. 2, no. 4, pp. 230–243, 2017.

[14] K. Goldberg, "Robots and the return to collaborative intelligence," *Nature Machine Intelligence*, pp. 2–4, Jan 2019.

[15] K. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, pp. 2674–2703, 2018.

[16] F. Tap, *Economics-based Optimal Control of Greenhouse Tomato Crop Production*. PhD thesis, Wageningen Agricultural University, 2000.

[17] B. Vanthoor, C. Stanghellini, E. Van Henten, and P. Visser, "A methodology for model-based greenhouse design: Part 1, a greenhouse climate model for a broad range of designs and climates," *Biosystems Engineering*, vol. 110, no. 4, pp. 363–377, 2012.

[18] T. R. Sinclair, T. W. Rufty, and R. S. Lewis, "Increasing photosynthesis: Unlikely solution for world food problem," *Trends in Plant Science*, vol. 24, no. 11, pp. 1032–1039, 2019.

[19] Quistnix, "Westland_kassen.jpg." https://commons.wikimedia.org/wiki/File:Westland_kassen.jpg, May 2005. [Online; accessed 07-Oct-2019].

[20] G. van Straten, "Optimal greenhouse cultivation control: Quo vadis?," *IFAC Proceedings Volumes*, vol. 46, no. 4, pp. 11 – 16, 2013.

[21] A. Ramírez-Arias, F. Rodríguez, J. Guzmán, and M. Berenguel, "Multiobjective hierarchical control architecture for greenhouse crop growth," *Automatica*, vol. 48, no. 3, pp. 490 – 498, 2012.

[22] Z. Gao, "On the centrality of disturbance rejection in automatic control," *ISA transactions*, vol. 53, no. 4, pp. 850–857, 2013.

[23] I. Lopez-Cruz, E. Fitz-Rodríguez, S. Raquel, A. Rojano-Aguilar, and M. Kacira, "Development and analysis of dynamical mathematical models of greenhouse climate: A review," *European Journal of Horticultural Science*, vol. 83, no. 5, pp. 269–279, 2018.

[24] W. J. Kuijpers, M. J. van de Molengraft, S. van Mourik, A. van 't Ooster, S. Hemming, and E. J. van Henten, "Model selection with a common structure: Tomato crop growth models," *Biosystems Engineering*, vol. 187, pp. 247 – 257, Nov 2019.

[25] A. de Koning, *Development and dry matter distribution in glasshouse tomato : a quantitative approach*. PhD thesis, Landbouwuniversiteit Wageningen, 1994.

[26] S. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733 – 764, 2003.

[27] R. Amrit, J. B. Rawlings, and D. Angeli, "Economic optimization using model predictive control with a terminal cost," *Annual Reviews in Control*, vol. 35, no. 2, pp. 178 – 186, 2011.

[28] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control* (A. Garulli and A. Tesi, eds.), pp. 207–226, Springer London, 1999.

[29] M. Behrendt, "A basic working principle of model predictive control." https://commons.wikimedia.org/wiki/File:MPC_scheme_basic.svg, October 2009. [Online; accessed 24-Sep-2019].

[30] J. Rawlings, E. Meadows, and K. Muske, "Nonlinear model predictive control: A tutorial and survey," *IFAC Proceedings Volumes*, vol. 27, no. 2, pp. 185 – 197, 1994.

[31] TomLab, "Tomlab optimization environment." https://tomopt.com/tomlab/.

[32] E. C. Kerrigan and J. M. Maciejowski, "Soft constraints and exact penalty functions in model predictive control," in *Proceedings UKACC International Conference on Control*, Sep 2000.

[33] I. Markovsky, J. C. Willems, S. van Huffel, and B. de Moor, *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach*. Society for Industrial & Applied Mathematics (SIAM), 2006.

[34] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *International Journal of Control*, vol. 81, no. 12, pp. 1946–1959, 2008.

[35] J. C. Willems, P. Rapisarda, I. Markovsky, and B. de Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325 – 329, 2005.

[36] S. Hemming, F. Zwart, A. Elings, I. Righini, and A. S. Petropoulou, "Analysis of results." Final Presentation Autonomous Greenhouse Challenge; Unpublished, 2020.

[37] S. Baros, C.-Y. Chang, G. E. Colon-Reyes, and A. Bernstein, "Online data-enabled predictive control," *arXiv e-prints*, p. arXiv:2003.03866, Mar 2020.

Glossary

List of Acronyms

MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
WC	Water Content
EC	Electrical Conductivity
DeePC	Data-Enabled Predictive Control
AI	Artificial Intelligence
RMSE	Root Mean Squared Error

List of Symbols

Δ_D	Dry-Down Rate Dark Period
Δ_L	Dry-Down Rate Light Period
$\hat{\Delta}_D$	Prediction Dry-Down Rate Dark Period
$\hat{\Delta}_L$	Prediction Dry-Down Rate Light Period
$\mathbb{W}$	Signal Space
$\mathbb{Z}_+$	Discrete Time Axis
$\mathcal{U}$	Control Input Constraint Set
$\mathcal{X}$	State Constraint Set
$\mathscr{B}$	System Behaviour
$\mathscr{H}_t$	Hankel Matrix of Order t
ξ_{dd}	Total Dry-Down
C_{i}	Greenhouse Air CO_2 Concentration

D	Crop Development Stage
j_{c_o}	Flux between Crop and Outside Environment
j_{e_g}	Flux between Equipment and Greenhouse
j_{g_c}	Flux between Greenhouse and Crop
j_{g_o}	Flux between Greenhouse and Outside Environment
m_{B}	Dry Weight Assimilate Buffer
m_{F}	Fruit Weight
m_{L}	Leaf Weight
N	Prediction Horizon
R_{Δ}	Change of Control Cost Matrix
S_{c}	Greenhouse Crop Subsystem
S_{g}	Greenhouse Climate Subsystem
T	Trajectory Length to Capture System Dynamics
T_{g}	Greenhouse Air Temperature
T_{p}	Heating Pipe Temperature
T_{s}	Greenhouse Soil Temperature
t_D	Length Dark Period
T_f	Trajectory Length for Future Trajectory Prediction
t_L	Length Light Period
T_{ini}	Trajectory Length for Implicit State Estimation
T_{stop}	Stop Time Irrigation
U_f	Input Data Hankel Matrix
u_l	Lower Bound Control Input
U_p	Input Data Hankel Matrix
u_u	Upper Bound Control Inputs
V_{i}	Greenhouse Air Absolute Humidity
V_f	Exogenous Data Hankel Matrix
V_p	Exogenous Data Hankel Matrix
x_l	Lower Bound System States
x_u	Upper Bound System States
x_{c}	Greenhouse Crop States
x_{g}	Greenhouse Climate States
Y_f	Output Data Hankel Matrix
Y_p	Output Data Hankel Matrix