Optimal Control of Autonomous Greenhouses

A Literature Survey

L. Kerkhof

Literature Survey

TU Delft Delft University of Technology

Delft Center for Systems and Control

Optimal Control of Autonomous Greenhouses

A Literature Survey

Literature Survey

L. Kerkhof

December 16, 2019

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

Abstract

The world population is growing and the demand for healthy food grows with it. Greenhouse cultivation provides a way to grow crops in a protected environment of which the climate can be controlled. 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 nonlinear dynamics of greenhouses, these models might not be applicable to control greenhouses other than the one which has been used to identify the parameters. 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 model but solely on input-output data of the system, hereby establishing a way to control systems with unknown dynamics.

This literature survey aims to provide an overview of the most widely used greenhouse and crop models and their shortcomings. Furthermore, an overview of recently developed greenhouse control algorithms is given. After that, an overview of advanced model based control algorithms is given which serves the purpose of introducing the need for a control algorithm that does not rely on a parametric system representation. Finally, such an algorithm is described in detail. This literature survey is concluded with the research question how the performance of model based control and data-driven control compare when controlling a greenhouse.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The world population is increasing rapidly. According to the United Nations, the world population will reach 9.77 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, crop quality and the crop cultivation period [3]. Furthermore, the greenhouse provides a protection against pests and diseases.

An important concern in 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 sustainability concern, one of the biggest issues within the greenhouse industry nowadays is finding sufficient experienced labor to manage crop production [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 growing [7] and the greenhouse equipment is becoming more advanced [8], the need for advanced greenhouse control algorithms increases.

Furthermore, recent advancements in the field of Artificial Intelligence (AI) has led to major breakthroughs in fields such as autonomous driving [9], healthcare [10] and robotics [11]. Within the greenhouse industry, AI has been applied to yield prediction, disease detection, weed detection, crop quality classification and species recognition [12]. However, AI is still scarcely used in greenhouse climate and irrigation control and in making crop management decisions. Hence, AI could contribute to a fully autonomous greenhouse cultivation period with

yield levels compared to commercial practice [13]. For this reason, Wageningen University and Research has organised the Autonomous Greenhouse Challenge. During this Challenge, 5 teams get the opportunity to grow cherry tomatoes in a remotely controlled greenhouse department. Since the greenhouse compartments are being remotely controlled, the teams 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 the tomatoes at a high level of production and a high resource use efficiency.

In order to increase the level of automation in the greenhouse industry, various model based control algorithms have been developed [13]. 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 [14]. Therefore, data-driven control algorithms, e.g. Data-Enabled Predictive Control (DeePC) [15], 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. Hence, in this literature study the aim is to provide an overview of the most widely used greenhouse models and their shortcomings in Chapter 2. Then, a short overview of tomato crop models is given in Chapter 3. Thereafter, an overview of recently developed greenhouse control algorithms is given in Chapter 4. In order to overcome the difficulties found in modelling the greenhouse dynamics, Chapter 5 reviews several advanced model based control algorithms and introduces the need for a data-driven control algorithm: DeePC, after which this algorithm is described in detail. Finally, in Chapter 6 the problem formulation for the subsequent thesis work is formulated based on the results found in this literature survey.

Chapter 2

Greenhouse Climate Models

2-1 The Physical Greenhouse

Throughout the world, many varieties of greenhouses exist. Essential differences are seen in the greenhouse shape, cover material (e.g., glass or polyethylene) and the principles of operation [16]. For this survey, 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 [17]. 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 e.g., leaves and fruits. 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.

Figure 2-1: Typical multi-span Venlo greenhouse; the type of greenhouse considered in this literature survey, [18].

Besides variations in greenhouse designs, there is a large diversity of crops which can be grown in a greenhouse as well; ranging from bulk crops such as tomato and cucumber to ornamental and medicinal plants [19]. Since different crops have different behaviour in terms of growing processes, there exists a large variety of control inputs and objectives when it comes to controlling a greenhouse. Therefore, in the beginning of this chapter a general greenhouse model is illustrated. After that, several widely used greenhouse climate models from literature are reviewed. Thereafter, the most commonly used greenhouse actuators and corresponding control inputs are described and with this information one of the greenhouse climate models is described in more detail. Finally, the shortcomings of the described greenhouse models are described.

2-2 Basic Greenhouse Principles

Regardless of all the possible variations in greenhouse designs and crops, there are some fundamentals which are present in all greenhouse/crop combinations. A model of the fundamental mass and energy fluxes within a common greenhouse is shown in Figure 2-2. In this model, the greenhouse and crop are considered as two separate subsystems: S_g and S_c , with states x_g and x_c , respectively. These systems influence each other and are influenced by the control inputs u and exogenous signals d. 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 state, the crop state and the disturbances acting on the greenhouse and will be called 'information flows' for conciseness as is done in [3].

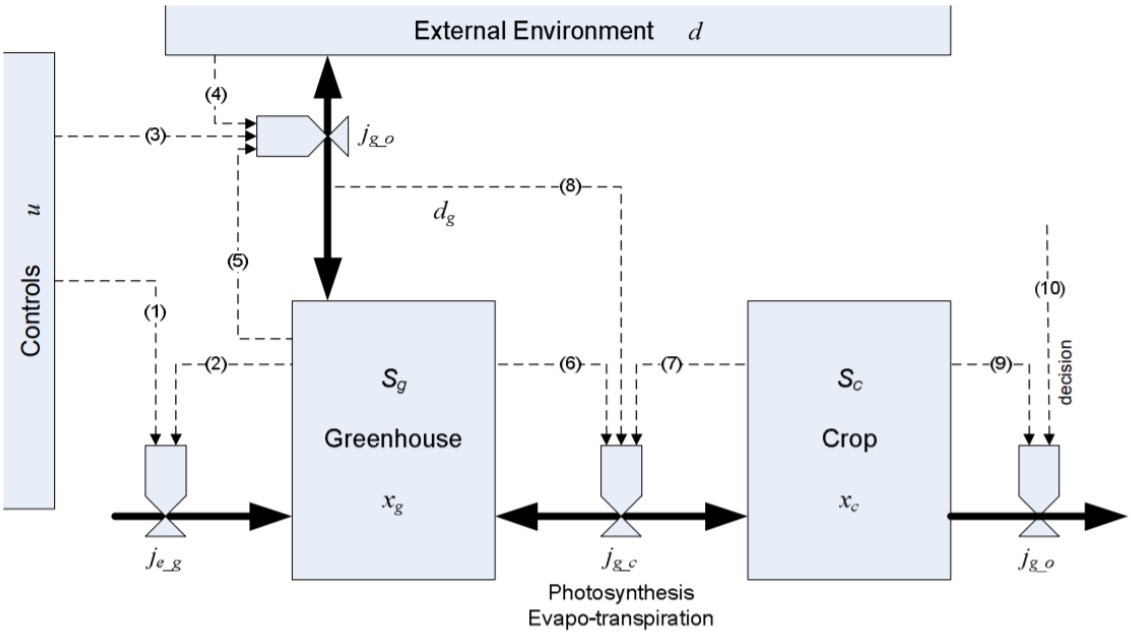

Figure 2-2: General model of a greenhouse with mass and energy fluxes (solid arrows) and information flows (dashed arrows) [3].

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) are the control inputs to the greenhouse such as the heating, irrigation or CO_2 supply.
- (2), (5), (6) denote the states of the greenhouse such as the inside air temperature, CO_2 concentration and relative humidity of the inside air.
- (3) are the control inputs to the windows of the greenhouse.
- (4) are the exogenous signals such as outside air temperature, wind speed and solar radiation.
- (7), (9) denote the states of the crop such as the growth stage of the crop or the amount of leafs and fruits on the crop.
- (8) denotes the solar radiation influence on the j_{g_c} fluxes, e.g., a higher amount of solar radiation could result in a higher CO_2 uptake.
- (10) are the discrete decision actions such as picking leafs, pruning and harvesting fruits which are usually determined by the grower.

Hence, the greenhouse and crop subsystems interact in a complex process with each other and with the control and exogenous 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 several greenhouse climate models which are described in literature.

2-3 Greenhouse Control Inputs

Modern greenhouses contain a broad range of equipment in order to stimulate crop growth. Hence, the amount of control inputs is extensive as well. In this section, commonly used greenhouse control inputs are described.

Heating Heat in the greenhouse is typically provided by a heating pipe system on the floor of the greenhouse. This system consists of steel pipes filled with hot water of which the temperature can be controlled. The water temperature is steered by controlling the mixing valve which allows hot water from a buffer tank to enter the heating system. Hence, with the temperature of the hot water, the temperature of the steel pipes is influenced and with the pipe temperature the greenhouse air temperature is influenced. The desired setpoints are

given for the greenhouse air temperature and these usually vary throughout the day. Hence, the control variables for the heating process are the water temperature in the buffer tank u_H [°C] and the relative opening of the mixing valve u_r [%].

Ventilation Ventilation of the greenhouse is done by opening the windows when the greenhouse air temperature rises above the desired setpoint. Typically, the ventilation strategy is defined as an offset with respect to the heating setpoint, e.g., when the heating setpoint is 20 °C with a ventilation offset of 2 °C, the windows will start to open when the temperature rises above 22 °C. For greenhouse air temperatures above 22 °C, the windows will open according to a proportional relationship with the temperature excess. This relationship is defined by the P-band. The P-band determines what the temperature excess should be in order to open the windows for 100 %, e.g., when the heating setpoint with ventilation offset is 22 °C and the P-band is 5 °C, the windows are opened for 0 % at 22 °C and for 100 % at 27 °C with linear interpolation in between. Hence, the control variables for the ventilation process are the minimum and maximum window opening, the ventilation offset and the P-band denoted by $u_{V,\min}$ [%], $u_{V,\max}$ [%], $u_{V,\text{vent}}$ [°C] and $u_{V,\text{Pband}}$ [°C], respectively.

Another important climate factor which plays a role in the ventilation process is humidity. Since for high levels of relative humidity (typically $\geq$90%) the risk on crop diseases increases, such levels should be avoided. Assuming that the outside level of humidity is always lower than the humidity level inside the greenhouse (which is typically the case in the Netherlands), the inside humidity level can be decreased by opening the windows. The window control for relative humidity is similar to the window control based on temperature; hence, when the relative humidity exceeds a threshold $u_{V,RH}$ [%], the windows will start to open proportional to a given humidity band $u_{V,\text{Hband}}$ [%] until it reaches the maximum opening of 100 %.

CO_2 The CO_2 usually originates from flue gasses of the boiler or from industrial waste. Here is assumed that the CO_2 can simply be supplied from a storage when desired. Hence, the control input will be the CO_2 supply rate $u_{C,\text{sup}}$ [$gm^{-2}hr^{-1}$]. The supply rate is limited to the maximum supply rate from the storage $u_{C,\max}$ [$gm^{-2}hr^{-1}$].

Artificial illumination Artificial illumination is provided by either LED or HPS lights attached to the roof of the greenhouse. The control inputs for artificial lighting are described in this paragraph. The first input is the intensity, i.e. the supplied amount of Photosynthetically Active Radiation (PAR) light, $u_{L,\text{int}}$ [µmol $m^{-2}s^{-1}$]. Furthermore, since most crops need a certain amount of darkness in order to synchronize their internal circadian rhythm, the illumination needs to be switched of during a part of the night. Hence, a maximum amount of hours light can be specified $u_{L,\text{hrL}}$ [hr]. Therefore, also a time at which the lights are switched off should be specified $u_{L,\text{hrT}}$ [hr].

Furthermore, during bright days it does not have to be necessary to have the illumination turned on because the solar radiation provides sufficient radiation. Hence, when the solar radiation exceeds the threshold $u_{L,\text{glob}}$ [Wm^{-2}] the lights will switch off. When the radiation provided in the morning by the artificial illumination and during a bright day by the solar radiation it might not be necessary to turn on the illumination in the afternoon again. Hence, when the maximum light sum $u_{L,\text{maxPAR}}$ [mol $m^{-2}day^{-1}$] is reached the illumination does not turn on again.

Screens Within the greenhouse there are different types of screens with different purposes. The most common types of screens used are the energy saving, shading and blackout screens. As the names suggest, the first one is used to save energy during nights and cold days, the second one is used to provide protection against excess solar radiation and the latter one is to block outgoing artificial light during the night. Hence, the energy saving screen will be applied when the outside temperature is below a given minimal temperature $u_{S,\text{minT}}$ [°C] or a minimum amount of solar radiation $u_{S,\text{minR}}$ [Wm^{-2}]. The shading screen will be applied when the solar radiation exceeds the threshold $u_{S,\text{maxR}}$ [Wm^{-2}]. The blackout screen will be deployed during the hours which are prescribed by legislation. The screens have a control input based on the relative opening, which are denoted by $u_{S,\text{energy}}$ [%], $u_{S,\text{shading}}$ [%] and $u_{S,\text{blackout}}$ [%], respectively.

Irrigation and fertilization Irrigation is supplied through a drip irrigation system which regulates that the irrigation is provided in shots of water at certain intervals. These intervals are mainly determined by the measured amount of PAR light. Hence, the control inputs here are the size of the shot $u_{I,\text{shot}}$ [cc shot^{-1}] and the amount of moles of PAR light after which a shot is given $u_{I,\text{PAR}}$ [mol]. However, on days with low PAR the plant still needs some irrigation and therefore the input $u_{I,\text{pause}}$ [hr] determines the maximum time interval between each shot.

Besides irrigation the plant needs fertilization as well. Fertilization is applied through the irrigation system by enriching the water with nutrients such as different types of salts. Due to the dissolved salts, the solvent is capable to conduct electricity. Therefore, the amount of nutrients in the water can be measured by measuring the Electrical Conductivity (EC) [mS cm^{-1}]. As a rule of thumb, the amount of salts per cubic meter of water [kgm^{-3}] corresponds roughly to the EC. Hence, the input for fertilization is $u_{I,\text{fert}}$ [mS cm^{-1}].

A summary of all common greenhouse control inputs discussed here is given in Table 2-1.

Table 2-1: List of common greenhouse control inputs

Control Input	Symbol
Heating water temperature	u_H
Relative opening of the mixing valve	u_r
Minimum window opening	$u_{V,\text{min}}$
Maximum window opening	$u_{V,\text{max}}$
Ventilation offset	$u_{V,\text{vent}}$
P-band	$u_{V,\text{Pband}}$
Maximum relative humidity	$u_{V,RH}$
Humidity band	$u_{V,\text{Hband}}$
CO_2 supply rate	$u_{C,\text{sup}}$
Maximum CO_2 supply rate	$u_{C,\text{max}}$
Light intensity	$u_{L,\text{int}}$
Maximum hours light	$u_{L,\text{hrL}}$
Time when lights are switched off	$u_{L,\text{hrT}}$
Solar radiation threshold	$u_{L,\text{glob}}$
PAR threshold	$u_{L,\text{maxPAR}}$
Minimum temperature for energy screen	$u_{S,\text{minT}}$
Minimum radiation for energy screen	$u_{S,\text{minR}}$
Maximum radiation for shading screen	$u_{S,\text{maxR}}$
Relative opening of energy screen	$u_{S,\text{energy}}$
Relative opening of shading screen	$u_{S,\text{shading}}$
Relative opening of blackout screen	$u_{S,\text{blackout}}$
Irrigation shot size	$u_{I,\text{shot}}$
PAR light per shot	$u_{I,\text{PAR}}$
Maximum time interval between shots	$u_{I,\text{pause}}$
EC of irrigation water	$u_{I,\text{fert}}$

2-4 Overview of Greenhouse Climate Models

In the past, various attempts to develop a dynamical model of the greenhouse climate system have been made [20]. In general, a distinction can be made between models based on physical (thermodynamic) properties such as laws conservation of energy and mass and models which are based on input-output data of the system, i.e. black box models [21]. These black box models could either be linear (e.g., auto-regressive models) or nonlinear (e.g., neural networks). Physical models consist of differential equations which could either be linear or nonlinear. Differences in the developed physical models are mainly the number of states and variations in (control) inputs. Furthermore, a distinction could be made between models which are used for simulation and models that are used for control. Usually, the simulation models are used to gain more insight in the greenhouse systems and are in general more complex whereas the models that are used for control are less complex in the sense that these models contain less states and input variables. Furthermore, a distinction can be made between Continuous Time (CT) and Discrete Time (DT) models.

Obviously, different types of greenhouses will result in different dynamical models. Hence,

only dynamical models of Venlo type greenhouses will be taken into account in this section. For dynamical models of different types of greenhouses the reader is referred to [20] and the references therein. Table 4-1 and Table 4-2 provide a list of widely used dynamical greenhouse models which are reported in literature. For each model in these tables there is indicated what type of model it concerns, what the inputs for the model are and a short summary of what characteristics are simulated in the model and possibly a remark if important elements are excluded.

Table 2-2: Dynamic greenhouse climate models.

Ref.	Model type	Model inputs	Remark(s)
[22]	Physical	Outside temperature, outside relative humidity, global solar radiation, wind speed	A CT model with nine states is derived for simulation purposes. The states that are modelled are the temperature of the greenhouse cover, temperature of greenhouse air, temperature of the crop canopy, temperature in four soil layers, inside air humidity, soil humidity. The model does not include artificial lighting.
[23]	Physical	Outside temperature, sky temperature, temperatures above and below the thermal screen, soil temperature, outside vapor pressure, outside CO_2 concentration, wind speed	A DT model with sixteen states is derived for simulation purposes. The states modeled here are the air temperature at the greenhouse cover, above and below the thermal screen, on the crop canopy, on the floor and in six soil layers, the CO_2 concentration above and below a thermal screen and vapor pressure above and below a thermal screen.
[24]	Physical	Outside temperature, sky temperature, soil temperature, global solar radiation, outside relative humidity, wind speed	A CT model with eight states is derived for simulation purposes. The states modeled here are the air temperature at the greenhouse cover, inside the greenhouse, at the crop canopy and in four soil layers and the vapor concentration of the greenhouse air. The model does not include the CO_2 balance.
[25]	Physical	Heating, CO_2 injection, window opening, solar radiation, wind speed, outside temperature, absolute humidity, outside CO_2 concentration	A CT three state model is derived for control purposes. The states modeled here are the greenhouse air temperature, inside CO_2 concentration and inside relative humidity. The model does not include artificial lighting.
[26]	Physical	Heating, CO_2 injection, window opening, solar radiation, wind speed, outside temperature, absolute humidity, outside CO_2 concentration	A CT model with five states is derived for control purposes. The states modeled here are the greenhouse air temperature, soil temperature, temperature of the heating pipe, inside CO_2 concentration and inside relative humidity. The model does not include artificial lighting.

Table 2-3: Dynamic greenhouse climate models, continued.

Ref.	Model type	Model inputs	Remark(s)
[27]	Black box (auto-regressive model)	Outside air temperature, outside relative humidity, global solar radiation, sky cloudiness	A linear auto regressive model with external input and an auto regressive moving average model with external input are identified using a data set with Belgian climate data and simulated greenhouse data from the model in [24]. The identified model is used to model the inside air temperature.
[28]	Black box (neural network)	Outside air temperature, outside humidity, global solar radiation, sky cloudiness	A neural network is trained using a data set with Belgian climate data and simulated greenhouse data from the model in [24]. The network is used to model the inside air temperature.
[29]	Black box (neural network)	Outside temperature, outside relative humidity, global solar radiation, window opening, screen position, heating, artificial light	A neural network is trained using data from a greenhouse with red pepper crops. The network is used to model the inside temperature and inside relative humidity.

2-5 A Greenhouse Climate Model in Detail

In this section, the greenhouse climate model from [26] is described in more detail to illustrate how such a model looks like. Hence, a five state climate model is described by the differential equations in (2-1), (2-2), (2-3), (2-4) and (2-6). This model is e.g., used in [30], [31] and [32] for the design of optimal greenhouse control algorithms. The five states represent the greenhouse air temperature T_g [°C], the temperature of the top soil layer T_s [°C], the heating pipe temperature T_p [°C], the greenhouse CO_2 concentration C_i [kg m^{-3}] and the greenhouse absolute humidity V_i [%]. The soil is modeled as two layer system with a top and a bottom layer where the temperature of the bottom layer is assumed to be constant. The states and their corresponding differential equations will be described in the next paragraphs. For completeness, similar notation as in [26] is used here.

Greenhouse air temperature

$$\frac{dT_g}{dt} = \frac{1}{C_g}\left(k_v(T_o - T_g) + \alpha(T_p - T_g) + k_r(T_o - T_g) + k_s(T_s - T_g) + \eta G - \lambda E + \frac{\lambda}{\epsilon + 1} M_c\right) \quad (2\text{-}1)$$

In (2-1), C_g is the specific heat capacity of the greenhouse air [J°C^{-1}m^{-2}], k_v is the ventilation heat transfer coefficient [W°C^{-1}m^{-2}], T_o is the outside temperature [°C], α is the heat transfer coefficient between the heating pipe and the greenhouse air [W°C^{-1}m^{-2}], T_p is the heating pipe temperature [°C], k_r is the heat transfer coefficient of the greenhouse cover [W°C^{-1}m^{-2}], k_s

is the heat transfer coefficient between the top soil layer and the greenhouse air [W°C^{-1}m^{-2}], η is the radiation conversion factor [-], G is the incoming short-wave solar radiation [Wm^{-2}], λ is the vaporisation energy of water [Jg^{-1}], E is the rate of transpiration of the crop [g s^{-1}m^{-2}] and finally $\frac{\lambda}{\epsilon+1}M_c$ represents the energy released to the greenhouse air by condensation of water vapour at the greenhouse cover, where M_c is the water mass flow resulting from condensation at the greenhouse cover [g s^{-1}m^{-2}] and $\frac{1}{\epsilon+1}$ is the fraction of the condensation heat transported to the greenhouse air. Furthermore, λ is a linear function of temperature T [°C] and vaporisation energy coefficients l_1 [Jg^{-1}] and l_2 [Jg^{-1}°C^{-1}], defined as: $\lambda = l_1 - l_2 T$.

Soil temperature

$$\frac{dT_s}{dt} = \frac{1}{C_s}(k_d(T_d - T_s) - k_s(T_s - T_g)) \tag{2-2}$$

In (2-2), C_s is the greenhouse soil heat capacity [J°C^{-1}m^{-2}], k_d is the heat transfer coefficient from the bottom soil layer to the top soil layer [W°C^{-1}m^{-2}], k_s is the heat transfer coefficient from the top soil layer to the greenhouse air [W°C^{-1}m^{-2}] and T_d is the temperature of the bottom soil layer [°C], which is assumed to be constant.

Heating pipe temperature

$$\rho C_p V_p \frac{dT_p}{dt} = \rho C_p \varphi(u_r T_{pi} - (1 - u_r)T_{po}) + A_p \beta G - A_p \alpha(T_p - T_g) \tag{2-3}$$

In (2-3), ρ is the density of water [kg m^{-3}], C_p is the specific heat of water at a constant pressure [J°C^{-1}kg^{-1}], V_p is the volume of the heating pipe [m^{-3}], φ is the water flow through the heating pipes [m^{3}s^{-1}], T_{pi} is the temperature of the water flowing into the heating pipe [°C], T_{po} is the temperature of the water flowing out of the heating pipes [°C], u_r is the relative opening of the mixing valve [%], A_p is the area of the outer surface of the heating pipes [m^2], β is the heat absorption efficiency [-], G is the incoming short-wave solar radiation [Wm^{-2}] and α is the pipe air heat transfer coefficient [W°C^{-1}m^{-2}].

CO_2 concentration

$$\frac{dC_i}{dt} = \frac{A_g}{V_g}(\Phi_v(C_o - C_i) + \varphi_{inj} + R - \mu P) \tag{2-4}$$

In (2-4), $\frac{V_g}{A_g}$ is the average greenhouse height [m], Φ_v is the natural ventilation flux [ms^{-1}], C_o is the outside CO_2 concentration [g m^{-3}], φ_{inj} is the CO_2 injection flux [g s^{-1}m^{-2}], R is the respiration of the crop [g s^{-1}m^{-2}], P is the crop photosynthesis rate [g s^{-1}m^{-2}] and μ is the fraction of molar weight of CO_2 and CH_2O [-].

Furthermore, Φ_v is a function of the wind speed w [ms^{-1}], the relative opening of the windows on both wind side r_{ww} [%] and lee side r_{wl} [%] and several ventilation rate parameters σ [-], χ [-], ζ [-], ξ [-] and ψ [ms^{-1}] and is defined as follows:

$$\Phi_v = \left(\frac{\sigma r_{wl}}{1 + \chi r_{wl}} + \zeta + \xi r_{ww}\right) w + \psi \tag{2-5}$$

Humidity

$$\frac{dV_i}{dt} = \frac{A_g}{V_g}(E - \Phi_v(V_i - V_o) - M_c) \tag{2-6}$$

In (2-6), V_o is the outside absolute humidity concentration [%] and M_c is a condensation term which only considers condensation at the cover of the greenhouse [g $s^{-1}m^{-2}$], hereby assuming that condensation on the plant not occurs. Furthermore, there is assumed that the ground area is covered with plastic and the re-evaporation of condensate from the greenhouse cover is neglected. Finally, E denotes the rate of transpiration of the crop [g $s^{-1}m^{-2}$].

Hence, the model shown here includes a lot of parameters which can not be measured and therefore should be identified by using input-output data. In this case this is done by minimizing the weighted sum of squared errors between the simulated data and measured data.

2-6 Conclusion

In this chapter several greenhouse climate models from literature are reviewed. However, these models might not be sufficiently reliable to be used for control purposes. This is, among others, due to the simplifications to which these models are subject. Examples of these simplifications are listed below:

- The ventilation flow is only based on the relative window opening and wind speed and not on factors such as the type of window or the geometric properties of the windows [25], [26].
- Leakage of CO_2 and heat loss through imperfections of the greenhouse is not taken into account [26].
- The heat transfer to the soil is not taken into account in [25], only heat transfer through ventilation and the greenhouse cover.
- The CO_2 balance is not taken into account in [24] or the model does not include artificial lighting [22], [25], [26].
- The temperature, CO_2 concentration and relative humidity inside the greenhouse are considered to be the same throughout the whole greenhouse. Hence, the greenhouse is considered as a perfectly stirred tank [33]. In practice, however, variations can be seen in different regions in the greenhouse.
- The identified models in [27] and [28] are determined by relying on both measured and simulation data. Hence the fitted models are partly dependent on an underlying model which might reduce the reliability of these fitted models.

Another important aspect here is that these models contain a lot of estimated parameters which should be identified by using experimental data. However, in practice greenhouses vary e.g., in size, volume, geometry, material, cultivated crops, equipment, window geometry and type of windows. Hence, all these variations result in different dynamics for each greenhouse. For example, take two greenhouses with the same equipment and area but with different

height. The greenhouse with the smallest height will have the lowest volume of air inside and hence the inside temperature will rise faster when similar heating is applied compared to the greenhouse with the larger height. Therefore, the parameters of the models which are calibrated with data from a specific greenhouse do not necessarily apply to other greenhouses. Examples of these parameters are the ventilation parameters in (2-5). These parameters have been estimated by minimizing the weighted sum of squared errors between the simulated data and measured data from a particular greenhouse compartment in Wageningen. However, greenhouses with different window sizes or a different window type will result in a different ventilation rate and hence the ventilation parameters should be identified again when using this model in a different greenhouse.

Another example is that different greenhouses are supplied with different equipment. For example the parameters in [22] have been estimated in a greenhouse which was only equipped with a thermal screen. However, greenhouses nowadays also are equipped with e.g., blackout screens which need to be deployed during the night due to legislation. Hence, these additional screens also effect the dynamics of the greenhouse when both screens are deployed. Therefore, this model could not be necessarily applied to greenhouses with these additional screens.

Hence, the need for data-driven models arises. Nowadays, greenhouses are equipped with a large variety of sensors which provide large amounts of measurement data. This allows the possibility to generate a model of the greenhouse purely from data. In this way, the complex dynamics of the greenhouse are implicitly taken into account and are not subject to a predefined parametric model. In Chapter 5 algorithms which combine data-driven models and control algorithms are described. Furthermore, an algorithm is described which does not rely on a parametric system representation but solely on the input-output data of the system. In Chapter 4, several state-of-the-art control algorithms of which many do rely on a parametric model are described.

Chapter 3

Crop models

3-1 Basic Crop Principles

The total greenhouse system includes both the greenhouse climate and crop subsystems which influence each other, see Figure 2-2. Hence, in order to obtain an optimal control strategy, besides a greenhouse climate model a model of the crop growth should be taken into account as well. Combining a greenhouse climate model with a crop model could result in an an optimal control strategy which should include crop decisions such as picking leafs, pruning or harvesting fruits. In the past, several crop growth modeling approaches have been proposed [34]. In this Chapter, a short overview is given of some well known crop models. For a more extensive overview of crop modeling approaches the reader is referred to [34] and the references therein.

Numerous processes could be taken into account when it comes to crop growth modeling. Most crop growth models describe the connection between the development of a crop, e.g., development of fruits, leafs and stem segments, and the most influential climate factors, e.g., temperature, radiation and CO_2 concentration. Processes which have less been focused on are e.g., the water balance of plants, the uptake of minerals, the interaction with pests, diseases and genetics, the inter-plant variability and the formation of product quality [34]. Furthermore, the diversity of species cultivated in greenhouses makes it more difficult to include crop growth models in a control loop since different species require different models. Despite the fact that different species share some basic features such as biomass production driven by photosynthesis, some processes are inherently crop specific, e.g., development of dry matter [34]. However, attempts have been done to develop a generic crop model, e.g., the CROPGRO model [35]. For this survey only tomato crop growth models will be taken into account, since only tomatoes will be grown during the Autonomous Greenhouse Challenge.

For tomato crops, distinctions can be made between some of the main processes. A lot of the models in the literature are using these processes as building blocks for their models. In Figure 3-1, a scheme of the main processes in tomato growth is shown.

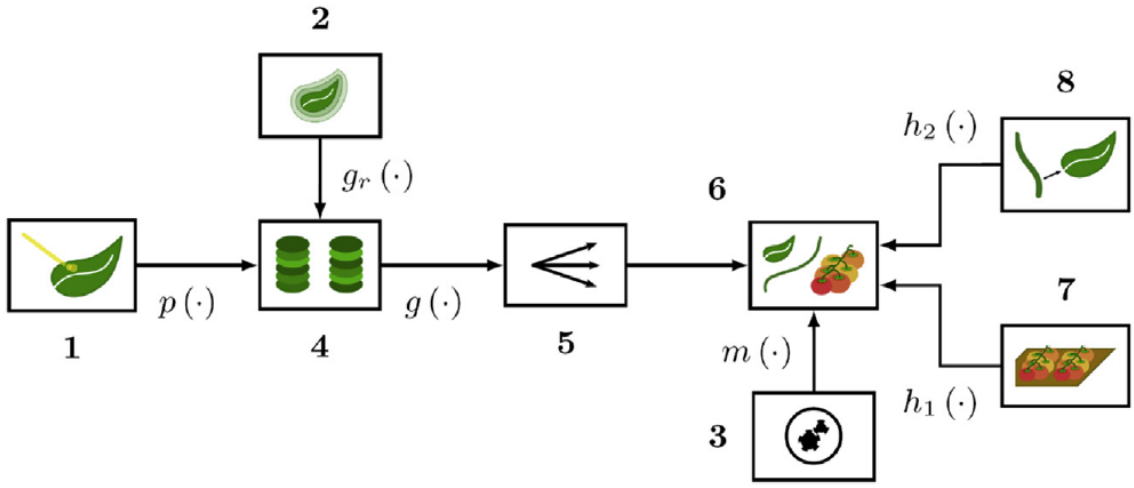

Figure 3-1: Scheme of the main processes in tomato growth. The numbers and symbols represent the following processes: 1: photosynthesis, 2: growth respiration, 3: maintenance respiration, 4: accumulating assimilates, 5: assimilate partitioning, 6: accumulating biomass, 7: tomato harvest, 8: leaf harvest, p: assimilates generated through photosynthesis, g_r: the assimilates used for growth, g: the amount of assimilates converted to biomass, m: part of the biomass used for maintenance respiration, h_1: harvest of fruits, h_2: harvest of leafs [36].

In [37], a dynamic tomato crop growth and yield model (TOMGRO) was developed. This model represents the changes in numbers and weight of leaves, fruits and stem segments under influence of temperature, CO_2 concentration and radiation with a set of differential equations. It included the source/sink relationship and an age structure for the state variable to account for the growth stage of the plant. The resulting model has seven state variables: the number and weight of fruits, leafs and stem segments and the leaf area expansion. Furthermore, each state has 20 age classes. In the model was assumed that water and nutrients were supplied adequately and pests and diseases were absent.

A model similar to TOMGRO was developed in [38]. In this work, a model (TOMSIM) was derived which describes the development of tomato crop growth and yield as a function of temperature, CO_2 concentration and radiation. Instead of using age classes for the fruits, leaves and stem segments, TOMSIM uses the physiological age of these parts. Hence, for initializing a simulation with TOMGRO, the fruits, leafs and stem segments should be divided into the age classes whereas for TOMSIM the physiological age for each of those parts should be determined. Another difference between TOMGRO and TOMSIM is the different modeling of the dry matter distribution in the crop (process 5 in Figure 3-1).

In [39], a dynamic tomato crop growth model was developed. The states of this model are truss position, fruit position within a truss, fruit development stage, fruit dry weight, truss development stage and truss dry weight. These state variables are used to simulate the assimilate demand and dry matter distribution in tomato crop. In this research, mainly the influence of temperature was analysed.

Hence, often models contain different descriptions of the same fundamental crop processes. In [36], a modeling approach is proposed where components from different models are combined to obtain a new crop model. For example, the photosynthesis modeling from one model is

combined with the biomass accumulation modeling from another model. In this work, a total of 27 combined models was derived from 4 existing models. The combined models were in some cases able to outperform the currently existing models when the simulated results were compared to measured data.

3-2 Conclusion

In this Chapter, a short overview is given of the general framework of tomato crop modeling and some modeling approaches are described. Disadvantages of using these models is that such models assume that crops grow under non-limiting biophysical conditions, i.e. there is assumed that water and nutrients are supplied adequately and pests and diseases are absent. Furthermore, these models are calibrated using empirical data which is obtained by growing the crops in certain temperature, CO_2 and radiation regions. The effects of extreme high and low temperatures are often not considered in these models and hence the model could only be used in the regions in which it is calibrated. Another drawback of using crop models is that it is difficult to account for the variation between plants of the same crop. Hence, it is difficult to take the stochastic behaviour of plants into account.

Furthermore, a lot of crop states can normally not be measured or only be measured by a destructive method. Therefore, data which could be measured without harming the crops should be used in a fully autonomous growing cycle. Examples of such measurements are thickness of the head of the crop, growth rate and fruit load. Hence, the question arises on how to take crop feedback into account in the control loop without having to take crop destructive measurements.

Chapter 4

Greenhouse Control

4-1 State-of-the-art

In general, each type of crop grows best in their own ideal climate. Hence, in order to reach the desired climate for a certain crop to grow, the greenhouse control inputs need to be steered in such a way that this desired climate is reached. Within the literature, many different approaches have been proposed in order to control the greenhouse climate. Many of the control algorithms encountered are model based, e.g., Model Predictive Control (MPC). Besides differences in the various control algorithms which are applied, there is a lot of variation in the choice of control inputs and objectives as well. Modern greenhouses have an extensive amount of potential control inputs (Table 2-1) and hence the possible control options to use in a greenhouse control algorithm are extensive as well. Therefore, some of the described control algorithms only take e.g., the heating as control input whereas other methods also include CO_2 injection, cooling, fogging, artificial lighting or ventilation as control inputs. Furthermore, the control objectives vary from tracking a temperature profile to minimizing the heating energy usage or maximizing the profit over a growing season. In the latter case, a crop growth model should be taken into account as well in order to make predictions of the profit.

In Table 4-1 and Table 4-2 an overview of greenhouse control algorithms described in literature is given. In these tables, for each reference there is specified what type of control algorithm was used, what the control objective was, which control inputs were used, what the result of the designed control algorithm was and some remarks on what models were used in the control algorithms or where the experimental greenhouse was located. In this way, an overview of the state-of-the-art in greenhouse control is given.

As can be seen from this tables, a vast amount of advanced control algorithms for greenhouse control is already developed in literature. However, in common practice barely any of the control algorithms published is used in commercial practice. According to [40], reasons for the lack of use are:

- A lack of reliable crop production models for the large variety of crops grown in greenhouses.
- A lack of trust from growers and doubts regarding the quality of greenhouse and crop models and due to the lack of experimentally demonstrated advantages.
- The need to leave part of the decision making freedom in the hands of the grower.
- The best approach to any model-based control strategy will require feedback from the crop state. Therefore, additional online plant measurements are necessary and these are often lacking in practice.
- There is a lack of accurate predictions of market prices.

Hence, in conventional horticultural practice most decisions are still in hands of the growers as is shown in the scheme in Figure 4-1. In this scheme, the grower provides setpoints for the greenhouse climate such as temperature, relative humidity or CO_2 concentration based on their experience. The climate computer then at its turn generates the control input for the actuators in a rule based way, based on the outside weather and the settings of the grower. Therefore, the conventional control boils down to simple P- or PI-controllers which control the indoor climate in a rule based way, based on the knowledge and experience of the grower and without explicitly including the control costs [26].

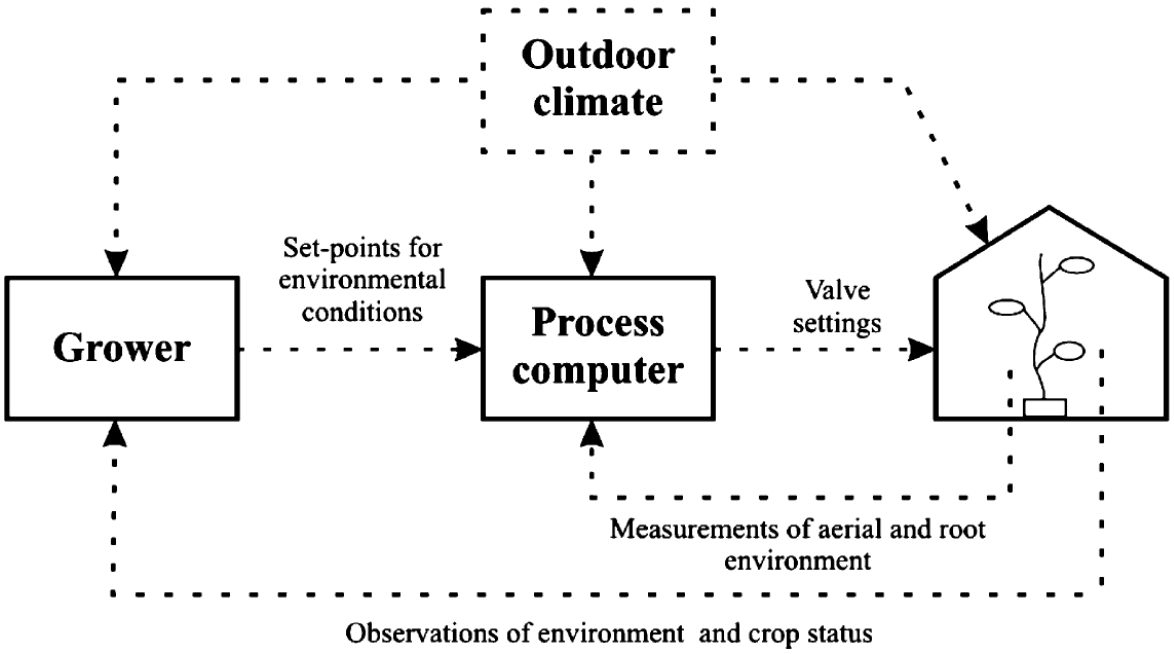

Figure 4-1: Scheme of greenhouse climate control in conventional horticultural practice [41].

Table 4-1: Greenhouse climate control methods.

Ref.	Control Algorithm	Objective	Control inputs	Result	Remark(s)
[25]	Receding Horizon Optimal Control	Maximize profit	Heating, ventilation, CO_2 injection	The control algorithm was able to achieve a higher profit compared to an experienced grower by reducing the heating while obtaining similar or higher fruit production.	The experimental greenhouse was located in the Netherlands. The climate model that was used is derived in the same work. A two state lettuce growth model was used as crop model.
[26]	Two Timescale Receding Horizon Optimal Control	Maximize profit	Heating, ventilation, CO_2 injection	The control algorithm showed a 8% decrease in energy consumption compared to conventional (PI) control in a real greenhouse experiment and showed an theoretical 60% increase in profit compared to conventional control.	The experimental greenhouse was located in the Netherlands. The climate model that was used is derived in the same work. The crop model that was used was a simplified version of [39].
[42]	Hierarchical control. Upper layer: MPC. Lower layer: Linear Quadratic Optimal Control	Minimize costs	Heating	Tracking of an optimal temperature reference specified by the controller is realised and the controller outperformed a classic PID controller in terms of control input efficiency.	The algorithm was only tested in simulation and the climate models from [25] and [26] were used.
[43]	Multi-objective Optimal Control	Maximize profit, fruit quality and efficiency of water usage	Heating, EC of irrigation water	A Pareto front was obtained from which trade-off solutions between the three objectives could be determined.	The experimental greenhouse was located in the Southeast of Spain.
[40]	Dynamic Optimal Control	Minimize total energy usage	Heating, cooling, ventilation, CO_2 injection	A theoretical reduction of 47% heating, 15% cooling and 10% CO_2 injection was realised compared to a common grower's strategy.	The algorithm was only tested in simulation and the climate models from [23] and [25] were used.
[44]	Robust Nonlinear MPC	Temperature regulation	Heating, ventilation	Tracking of temperature reference was realised in the presence of disturbances and simulations showed that the algorithm outperformed conventional MPC control in terms of robustness.	The algorithm was only tested in simulation and the model that was used was largely based on the climate model from [25].
[45]	Nonlinear MPC	Temperature regulation	Ventilation	Tracking of temperature reference was realized in both simulated and experimental cases.	The algorithm used a second-order Volterra series model which was identified from experimental input/output data. The experimental greenhouse was located in the Southeast of Spain.

Table 4-2: Greenhouse climate control methods, continued.

Ref.	Control Algorithm	Objective	Control inputs	Result	Remark(s)
[31]	Nonlinear MPC	Temperature regulation	Heating, ventilation, CO_2 injection	Tracking of temperature reference was realised and the controller outperformed conventional PI controllers in terms of control input efficiency.	The effects of plants on the greenhouse climate was neglected. The experimental greenhouse was located in Germany. The climate models from [25] and [26] were used.
[46]	H∞ Loop Shaping	Temperature and humidity regulation	Ventilation, fogging	Tracking of temperature and humidity references under relatively large (50%) parametric uncertainties is realized within predefined bounds	The experimental greenhouse was located in Israel. A simplified climate model from [47] was used.
[48]	Fuzzy Control	Temperature regulation	Heating	Temperature is maintained within a desired interval.	The algorithm was only tested in simulation. A simple energy balance was used as model.
[49]	Bayesian Network Control	Temperature regulation	Ventilation	Tracking of a temperature reference was realized by using the control input with the highest probability value generated by the Bayesian network.	The algorithm was only tested in simulation and the climate model from [50] was used.
[32]	Two Timescale Receding Horizon Optimal Control	Maximize profit	Heating, ventilation, CO_2 injection, LED lighting	A theoretical increase of 2.98% (without LED lighting) and 24.5% (with LED lighting) was realised compared to a lettuce cultivation experiment with LED lights.	The algorithm was only tested in simulation and the climate models that were used were largely based on [25] and [26]. A lettuce growth model was used as crop model.
[51]	Reinforcement Learning	Minimize daily heating costs while maintaining desired crop growth and development rate	Heating	A policy was learned under uncertainty, which reached desired temperature intervals during day and night. In some cases the learned policy outperformed the policy of growers based on a model of greenhouse roses.	The algorithm was only tested in simulation. A combined rose crop model with greenhouse temperature control from [52] was used here.

4-2 Conclusion

In this Chapter several state-of-the-art greenhouse control algorithms are reviewed. An important remark here is that almost all control algorithms rely on a fixed greenhouse climate or crop model. However, due to the complex and highly nonlinear dynamics of greenhouses, the models might lack accuracy for certain regions in which the greenhouses are controlled or the models might not be applicable to every greenhouse design. As shown in this Chapter, this is also one of the reasons why advanced control algorithms are barely used in practice. Furthermore, many of the control algorithms only focus on the climate and do not take a crop model into account. Hence, no crop feedback is taken into account.

Therefore, it is promising to use a data-driven approach. In this way, a general control algorithm could be designed that could be widely applied to different greenhouse designs and locations, as long as there is sufficient data available from the concerning greenhouse. Furthermore, if there is sufficient data of the crop available, it would be possible to take feedback of the crop into account within the control loop. Therefore, the next Chapter describes an algorithm which replaces the need for a fixed model by a purely data-driven control approach.

Chapter 5

Control Algorithms

In this chapter, several advanced control algorithms will be reviewed. First, a description of the linear Model Predictive Control (MPC) framework is given. Thereafter, the extensions to the nonlinear and the stochastic case are described. Next, in order to overcome issues with obtaining an explicit system description, techniques such as subspace predictive control and combining neural network models with MPC are discussed. Finally, an algorithm which does not rely on a parametric system, Data-Enabled Predictive Control (DeePC), is described in detail.

For this chapter the assumption is done that the optimal greenhouse control problem will boil down to optimal trajectory tracking. Optimal trajectory tracking is the guidance of state or output trajectories of a dynamical system along a desired reference trajectory [53]. Hence, in the description of the following control algorithms optimal trajectory tracking will be the objective.

Now, consider the following linear discrete-time state space system:

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{5-1}$$

Here $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ are respectively the system state, control input and system output at time $t \in \mathbb{Z}_{\geq 0}$. $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$ and $D \in \mathbb{R}^{p \times m}$ are the system matrices.

5-1 Model Predictive Control

5-1-1 Linear MPC

MPC is a widely used receding-horizon control technique which optimizes a cost function while satisfying system dynamics, input constraints and state constraints. This technique computes the optimal control input over a prediction horizon at each time step in order to

track a desired reference, see Figure 5-1. Hence, when an explicit parametric model is available such as (5-1), an optimization problem can be formulated where a cost function is minimized while the problem is subject to the system dynamics and the input and state constraints [15]. First, in (5-2) a general MPC framework with an arbitrary cost function is shown [54]. Then, in Algorithm 1, the outline of the MPC algorithm is shown [55].

$$\begin{aligned} \min_{u} \quad & V_N(x,u) = \sum_{k=0}^{N-1} \ell(x_k, u_k) + V_f(x_N) \\ \text{subject to} \quad & x_{k+1} = Ax_k + Bu_k, \quad \forall k \in \{0, ..., N-1\}, \\ & y_k = Cx_k + Du_k, \quad \forall k \in \{0, ..., N-1\}, \\ & x_0 = \hat{x}(t), \\ & u_k \in \mathcal{U}, \quad \forall k \in \{0, ..., N-1\}, \\ & y_k \in \mathcal{X}, \quad \forall k \in \{0, ..., N-1\}. \end{aligned} \tag{5-2}$$

Here, $V_N(x,u)$ is the cost function which is minimized. In this cost function, $\ell(x_k, u_k)$ is the stage cost computed at each prediction time step k and $V_f(x_N)$ is the terminal cost where $V_f : \mathbb{R}^n \to \mathbb{R}$ is the penalty function on the terminal state computed at the final prediction time step $k = N$. Furthermore, $N \in \mathbb{Z}_{>0}$ is the time horizon, $u = (u_0, ..., u_{N-1})$ are the optimization variables, the predicted states and outputs are denoted by $x = (x_0, ..., x_N)$ and $y = (y_0, ..., y_{N-1})$, respectively. $\hat{x}(t)$ denotes the state estimation at time t where $t \in \mathbb{Z}_{\geq}0$ is the time step at which the optimization problem is solved. Hence, if the full state is measured, $\hat{x}(t) = x(t)$. If the full state measurement is not available and the system is observable, an observer is typically used to estimate the state. $\mathcal{U} \subseteq \mathbb{R}^m$ 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), constraint sets $\mathcal{U}$ and $\mathcal{X}$;
1) Get the state estimate $\hat{x}(t)$.
2) Solve (5-2) 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.

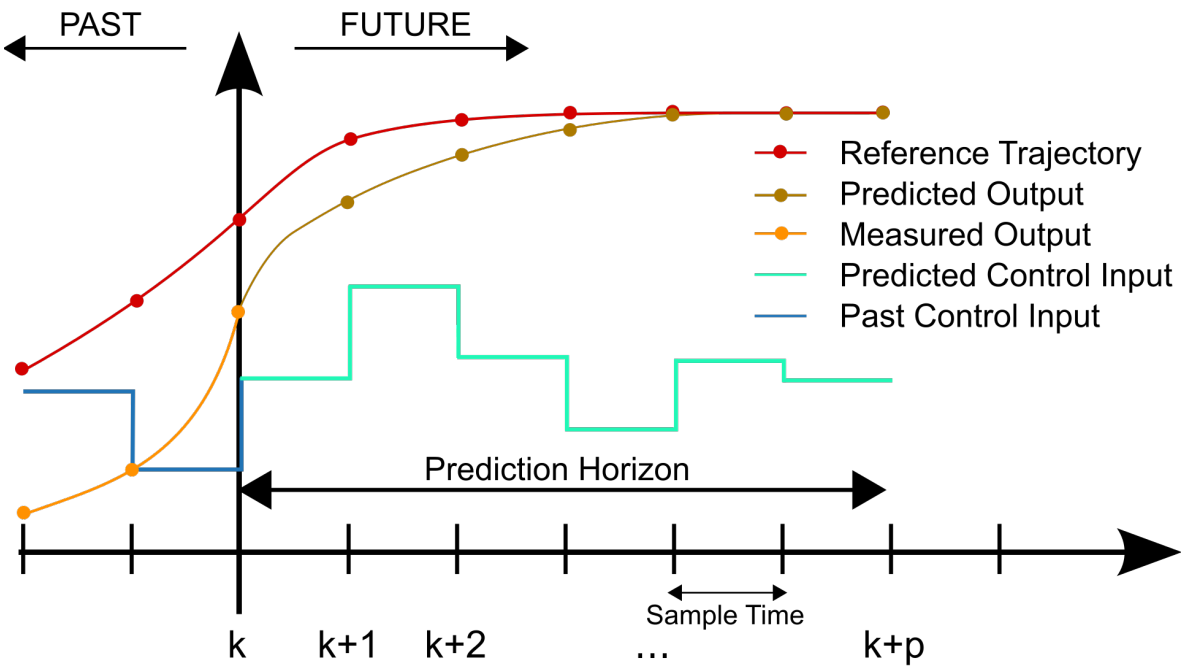

Figure 5-1: MPC scheme [56].

5-1-2 Nonlinear Model Predictive Control

Since in practice all systems are nonlinear by nature, a system description as in (5-1) might not be appropriate to model the system dynamics. Therefore, a nonlinear system description (5-3) could be used [57] in the MPC algorithm:

$$\begin{cases} x(t+1) = f(x(t), u(t)) \\ y(t) = h(x(t), u(t)) \end{cases} \tag{5-3}$$

Here, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ are nonlinear functions which map the current state and input to the next state and current output, respectively. With this nonlinear system representation, we arrive at the following Nonlinear Model Predictive Control (NMPC) framework [58]:

$$\begin{aligned} \min_{u} \quad & V_N(x,u) = \sum_{k=0}^{N-1} \ell(x_k, u_k) + V_f(x_N) \\ \text{subject to} \quad & x_{k+1} = f(x_k, u_k), \quad \forall k \in \{0, ..., N-1\}, \\ & y_k = h(x_k, u_k), \quad \forall k \in \{0, ..., N-1\}, \\ & x_0 = \hat{x}(t), \\ & u_k \in \mathcal{U}, \quad \forall k \in \{0, ..., N-1\}, \\ & y_k \in \mathcal{X}, \quad \forall k \in \{0, ..., N-1\}. \end{aligned} \tag{5-4}$$

which is solved at every $t \in \mathbb{Z}_{\geq} 0$. The algorithm for NMPC is similar to Algorithm 1, only (5-2) is replaced by (5-4).

5-1-3 Stochastic Nonlinear Model Predictive Control

Until now we assumed all systems to be deterministic, i.e. there are no disturbances in the system or errors in the output measurement. However, in practice all systems behave in a non-deterministic way since the system dynamics and output measurements are always subjected to uncertainties. Hence, the model in (5-3) could be extended as follows [59]:

$$\begin{cases} x(t+1) = f(x(t), u(t), w(t)) \\ y(t) = h(x(t), u(t), v(t)) \end{cases} \tag{5-5}$$

Here $w(t) \in \mathbb{R}^{n_w}$ and $v(t) \in \mathbb{R}^{n_v}$ are process noise and measurement noise respectively. $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{n_w} \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{n_v} \to \mathbb{R}^p$ are nonlinear functions which map the map the current state and input to the next state and current input under the influence of the process noise and measurement noise, respectively. The process noise and measurement noise are typically described as random variables, which results in a stochastic nonlinear model (5-5). Hence the cost function $V_N(x, u)$ becomes a random variable. In order to obtain a deterministic objective, the expectation of the cost function is considered: $\mathbb{E}[V_N(x, u)]$ [60]. Now, besides the set of feasible inputs $\mathcal{U}$ and feasible states $\mathcal{X}$, consider the set of feasible outputs denoted by $\mathcal{Y}$. Due to the stochastic nature of the states and outputs, the deterministic constraints are replaced with *change constraints* which are defined as follows [59]: $\mathbb{P}[y_k \in \mathcal{Y}] \geq 1 - \beta_y$, and $\mathbb{P}[x_k \in \mathcal{X}] \geq 1 - \beta_x$, where $\mathbb{P}[y_k \in \mathcal{Y}]$ denotes the probability that y_k is contained in the output constraint set $\mathcal{Y}$ (similar for u). Furthermore, $\beta_y \in (0, 1]$ and $\beta_x \in (0, 1]$ are the permitted constraint violation probabilities. However, since an optimization problem requires all variables to be deterministic, in practice the change constraints need to be reformulated into deterministic constraints [61].

When the expectation of the stochastic cost function and the change constraints are combined, we arrive at the following Stochastic Nonlinear Model Predictive Control (SNMPC) framework: [59]:

$$\begin{aligned} \min_{u} \quad & \mathbb{E}[V_N(x, u)] \\ \text{subject to} \quad & \mathbb{P}[x_k \in \mathcal{X}] \geq 1 - \beta_x, \quad \forall k \in \{0, ..., N-1\}, \\ & \mathbb{P}[y_k \in \mathcal{Y}] \geq 1 - \beta_y, \quad \forall k \in \{0, ..., N-1\}, \\ & x_0 = \hat{x}(t), \\ & u_k \in \mathcal{U}, \quad \forall k \in \{0, ..., N-1\}. \end{aligned} \tag{5-6}$$

which is solved at every $t \in \mathbb{Z}_{\geq}0$. For more details and some extensions on SNMPC the reader is referred to [59] and the references therein.

5-2 Subspace Predictive Control

However, in practice an accurate parametric system is not always known and hence an appropriate model should be identified [62]. A well established way to identify a model from input and output data of a dynamical system is subspace identification [63]. Subspace identification algorithms derive the state space matrices of a system (5-1) by using input and output measurements only [64]. Well known subspace identification algorithms are Multivariate Output-Error State Space (MOESP) [65] and Numerical Algorithms for Subspace State Space System Identification (N4SID) [66]. Hence, a two stage problem arises of first identifying an appropriate system and thereafter using this system to compute the optimal control policy. In [67] and [68], the authors propose a method to sequentially perform system identification and controller synthesis for systems with unknown dynamics.

However, instead of sequentially performing the system identification and controller design there are also algorithms to simultaneously perform both methods, called Subspace Predictive Control (SPC) [69]. In this algorithm both system identification and model predictive control design are combined in an elegant way and solved simultaneously.

5-2-1 Introduction to Subspace Identification

Consider the following LTI system

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) + Ke(t) \\ y(t) = Cx(t) + Du(t) + e(t) \end{cases} \tag{5-7}$$

Which is similar to the system in (5-1) but with an additional noise sequence $e(t) \in \mathbb{R}^{n_e}$, which is assumed to be zero-mean Gaussian white noise with variance $E[e_p e_q^T] = S\delta_{pq}$ and noise system matrix $K \in \mathbb{R}^{n \times n_e}$. With the system in (5-7) the following input-ouput equations can be formulated:

$$\begin{cases} Y_f = \Gamma_N X_f + H_N U_f + H_N^s E_f \\ Y_p = \Gamma_M X_p + H_m U_p + H_M^s E_p \end{cases} \tag{5-8}$$

Where U_p and U_f are both Hankel matrices constructed from measured input data u_k for $k \in \{1, ..., j+M+N\}$:

$$U_p = \begin{pmatrix} u_1 & u_2 & \cdots & u_j \\ u_2 & u_3 & \cdots & u_{j+1} \\ \vdots & \vdots & \ddots & \vdots \\ u_M & u_{M+1} & \cdots & u_{j+M-1} \end{pmatrix} \tag{5-9}$$

$$U_f = \begin{pmatrix} u_{M+1} & u_{M+2} & \cdots & u_{M+j} \\ u_{M+2} & u_{M+3} & \cdots & u_{M+j+1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{M+N} & u_{M+N+1} & \cdots & u_{M+N+1} \end{pmatrix} \tag{5-10}$$

where the subscripts p and f denote the past and future, respectively. The matrices Y_f, Y_p, E_f and E_p are constructed in a similar fashion by using y_k and e_k, respectively. Furthermore,

X_p and X_f are past and future state sequences defined by: $X_p = (x_1, ..., x_j)$ and $X_f = (x_{M+1}, x_{M+2}, ..., x_{M+j})$. Furthermore, denote the following matrices:

$$W_p = \begin{pmatrix} Y_p \\ U_p \end{pmatrix} \tag{5-11}$$

$$\Gamma_q = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{q-1} \end{pmatrix} \tag{5-12}$$

$$H_q = \begin{pmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ CA^{q-2}B & CA^{q-3}B & \cdots & D \end{pmatrix} \tag{5-13}$$

$$H_q^s = \begin{pmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ CA^{q-2}B & CA^{q-3}B & \cdots & D \end{pmatrix} \tag{5-14}$$

Where Γ_q is the extended observability matrix and H_q and H_q^s are the block Toeplitz matrices containing the impulse response of the system to the deterministic input u_k and the stochastic input e_k, respectively.

Most subspace identification have two common steps: In the *first* step the future outputs are predicted based on the past inputs and outputs and future inputs:

$$\hat{Y}_f = L_w W_p + L_u + U_f \tag{5-15}$$

Then, the least squares prediction $\hat{Y}_f$ of Y_f can be found as the solution of the least squares problem:

$$\min_{Lw,Lu} \| Y_f - \begin{pmatrix} L_w & L_u \end{pmatrix} \begin{pmatrix} W_p \\ U_f \end{pmatrix} \|_F^2 \tag{5-16}$$

With the obtained L_w and L_u predictions of the future outputs could be made. However, since the data might be subject to measurement noise. Hence, these noisy measurements will propagate in the matrices L_w and L_u. Therefore we arrive at the *second* step of computing the Singular Value Decomposition (SVD) of L_w:

$$L_w = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} \tag{5-17}$$

In order to remove the effects of the noise, an approximation of L_w is done by only keeping the most dominant singular values S_1, and hence we obtain: $L_w \approx U_1 S_1 V_1^T$.

Now using the matrix $L_w \approx U_1 S_1 V_1^T$, it is possible to estimate the extended observability matrix $\hat{\Gamma}_N$ and the future state sequence matrix $\hat{X}_f$ as follows [70]:

$$\hat{\Gamma}_N = U_1 S_1^{1/2} \tag{5-18}$$

$$\hat{X}_f = S_1^{1/2} V_1^T W_P \tag{5-19}$$

The differences between several subspace identification algorithms can be found in the way how the system matrices are derived from $\hat{\Gamma}_N$ and $\hat{X}_f$ [69]. However, for the SPC algorithm obtaining the approximation $L_w \approx U_1 S_1 V_1^T$ is sufficient.

5-2-2 Subspace Identification Based MPC

Now consider the following cost function:

$$J = \sum_{k=1}^{N} (\hat{y}_k - r_k)^T Q_k (\hat{y}_k - r_k) + u_k^T R_k u_k \tag{5-20}$$

Where $\hat{y}_k$ is the k-step ahead predicted output, u_k is the future input and $Q_k \in \mathbb{R}^{p \times p}$ and $R_k \in \mathbb{R}^{m \times m}$ are the output and control cost matrices, respectively. Now define the following vectors and matrices: $u_f = (u_1, ..., u_N)^T$, $\hat{y}_f = (\hat{y}_1, ..., \hat{y}_N)$, $r_f = (r_1, ..., r_N)$, $Q = \text{diag}(Q1, ..., Q_N)$ and $R = \text{diag}(R_1, ..., R_N)$. Then we can rewrite (5-20) as follows:

$$J = (\hat{y}_f - r_f)^T Q (\hat{y}_f - r_f) + u_f^T R u_f \tag{5-21}$$

Now using the result from (5-16), $\hat{y}_f$ can be expressed as:

$$\hat{y}_f = L_w w_p + L_u u_f \tag{5-22}$$

Where $w_p = (y_p, u_p)^T$ with $y_p = (y_{-M+1}, ..., y_{-1}, y_o)^T$ and $u_p = (u_{-M+1}, ..., u_{-1}, u_0)$

Using this result, the cost function in (5-21) can then be rewritten as:

$$J = (L_w w_p + L_u u_f - r_f)^T Q (L_w w_p + L_u u_f - r_f) + u_f^T R u_f \tag{5-23}$$

Now taking the trace of the derivative of (5-23) with respect to u_f, equating it to zero and solve it for u_f results in the following control law:

$$u_f = (R + L_u^T Q L_u)^{-1} L_u^T Q (r_f - L_w w_p) \tag{5-24}$$

Finally, in Algorithm 2 the outline of the SPC algorithm is shown.

5-3 DeepMPC

As mentioned before, MPC needs a model in the computation of the optimal control input.

Combining a neural network with MPC is often referred to as DeepMPC in the literature [71], [72]. In DeepMPC, instead of a physical or identified model of the system, a neural network is used as system representation. This network is trained on input and output data of the system that one attempts to model. Neural networks are capable of learning complex and non-linear dynamics [73] and could therefore be used in the NMPC framework.

For example, in [71] the authors trained a deep neural network on measured data from a food cutting robot to learn the dynamics when cutting different types of food. In this way, the complex and time-consuming task of designing separate controllers for each material is replaced with learning features of material properties. The trained network is then combined with a NMPC algorithm to compute the optimal control inputs for the cutting robot hand. The combined algorithm was able to reach significant improvements in cutting rate for different hard-to-cut materials.

In [72] a recurrent neural network is used to model an unsteady fluid flow. Since the dynamics of an unsteady fluid flow are highly nonlinear and possess high dimensionality, real-time feedback control becomes infeasible. In this paper, a neural network is trained to capture the dominant, low-dimensional behaviour of the system. The trained network is then used to replace the physical model of the fluid flow in the NMPC framework. The algorithm then computes the optimal angular velocity of a cylinder in the fluid to achieve a desired fluid flow.

5-3-1 Neural Networks

A short introduction to neural networks is given here: In Figure 5-2 a layout of a simple neural network is shown. The inputs of the network are denoted by $u_1, ..., u_m$. Each input is multiplied by a weight $w^h_{m,n}$, where the superscript denotes the hidden layer and the subscript m corresponds to input m and subscript n corresponds to hidden layer neuron n. Each neuron in the hidden layer consist of a nonlinear function $f_h : \mathbb{R} \to \mathbb{R}$, called the activation function. Examples of functions which are often chosen as activation functions are the logistic function,

Algorithm 2: SPC

Input: Input and output data $\{u_k, y_k$ for $k \in \{1, ..., j + M + N\}$, reference trajectory $r \in \mathbb{R}^{N_p}$ and cost matrices Q and R;

1) Construct the data block Hankel matrices Y_f, U_f and W_p from the input-output data.
2) Solve the least square problem in (5-16) to obtain L_w and L_u.
3) Take the SVD of L_w (5-17) and take the low rank approximation: $L_w \approx U_1 S_1 V_1^T$.
4) Construct w_p with the last M inputs and outputs:
$w_p = (y^T_{-M+1}, ..., y^T_{-1}, y^T_0, u^T_{-M+1}, ..., u^T_{-1}, u^T_0)$
5) Solve (5-24) to obtain u_f.
6) Implement the first input u_1 of the control sequence u_f and measure the corresponding new output y_1.
7) Return to 4.

the hyperbolic tangent function and the rectified linear unit (ReLU) function [74]. This activation function takes the summation of the multiplied input and weights and output as input and gives the nonlinear mapping as output. Then the outputs of the neurons in the hidden layers are again multiplied by a weight $w^o_{n,p}$ which corresponds to the weight between hidden layer neuron n and output p and where o denotes the output layer. These multiplied hidden layer outputs and weights are then summed and again fed through another activation function f_o. After this final activation function is applied, the outputs $y_1, ..., y_p$ are obtained. This procedure from calculating the outputs from inputs is often referred to as a *forward pass* through the network. Furthermore, the network shown here has only one hidden layer for simplicity but a network can have an arbitrarily amount of hidden layers.

The goal of the network is to map the inputs $u_1, ..., u_m$ to the desired outputs $y_1, ..., y_p$. Hence, the weights $w^h_{1,1}, ..., w^h_{m,n}, w^o_{1,1}, ..., w^o_{n,p}$ should be obtained in such a way that the correct mapping is reached. The process of obtaining the correct values of the weights is often referred to as *training* the network. The training is done by optimizing the weights based on an output error function such as the mean squared error [75]. Hence when corresponding input and output data is available, the error between the network output y and the true output y_t could be calculated based on a cost function $J(y, y_t)$. Then the gradient of the cost with respect to the weights is calculated; hence the gradient of the cost function with respect to the weight $w^h_{m,n}$ is denoted by: $\partial J(y, y_t)/\partial w^h_{m,n}$. Thereafter, the weights are update by the update rule in (5-25).

$$w^h_{m,n} \leftarrow w^h_{m,n} - \alpha * \frac{\partial J(y, y_t)}{\partial w^h_{m,n}} \tag{5-25}$$

Here, α denotes the learning rate which determines how much influence the gradient has on the weights. Hence by repeatedly updating the weights the error between the network output and the true output is minimized and the network 'learns' the relation between the input and output data. The procedure of updating the weights based on the output of the network is often referred to as a *backward pass* or *back propagation* through the network. In [76] it is shown that a finite linear combination of fixed, univariate functions can approximate any function of n real variables under some mild conditions. Hence a neural network as shown in Figure 5-2 with at least one hidden layer can approximate any continuous nonlinear function $f : \mathbb{R}^n \to \mathbb{R}$ arbitrarily well.

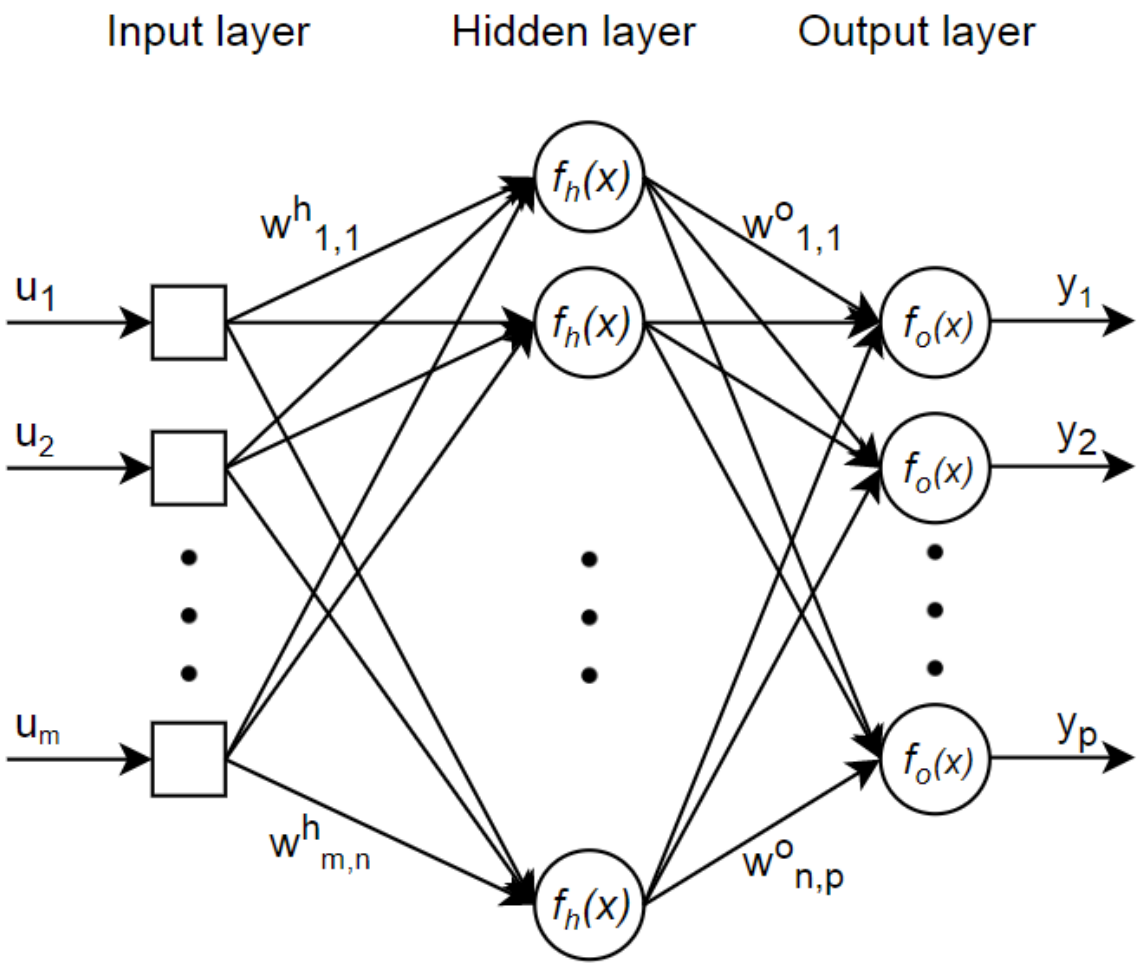

Figure 5-2: Graphical representation of a simple neural network.

Hence, when data from a nonlinear system $y(t) = f(x, u, t)$ is available, a neural network could be used to represent the system. The network should then be trained on data of the system. However, the neural network as presented so far is not able to capture time dependent relations which are often present in dynamical systems. Hence, different network architectures are necessary here. Architectures which are specially designed to capture time dependent relations are the so-called Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) [77]. Examples of applications of these architectures are adaptive control of non-linear systems [78], stock price prediction [79] and text classification [80]. For a more in depth review of neural networks and the RNN and LSTM architectures, the reader is referred to [74] and [81] and the references therein.

Hence, RNN and LSTM architectures are promising network architectures when it comes to modelling nonlinear dynamical systems. In order to apply these architectures to greenhouse control, sufficient input and output data of a greenhouse is necessary to train the network. When a neural network model of a greenhouse is developed, it could be combined with the NMPC algorithm in order to reach optimal greenhouse control.

5-4 Data-Enabled Predictive Control

DeePC is an algorithm that computes optimal control policies using real-time feedback driving the unknown system along a desired trajectory while satisfying system constraints [15]. The algorithm uses a finite number of data samples from the unknown system to learn a non-parametric system model. This non-parametric system model is used to predict future trajectories of the unknown system. The DeePC algorithm does not rely on a parametric system representation but approaches the problem from a behavioural systems theory perspective instead. Therefore, the next subsection introduces a few preliminaries from behavioural systems theory.

5-4-1 Preliminaries of Behavioural System 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 of the 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 [15][82]:

Definition 4.1: A *dynamical system* is a 3-tuple $\Sigma = (\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ where $\mathbb{Z}_{\geq 0}$ is the *discrete-time axis*, $\mathbb{W}$ is the *signal space* and $\mathscr{B} \subseteq \mathbb{W}^{\mathbb{Z}_{\geq 0}}$ is the *behaviour*.

Definition 4.2: Let $\Sigma = (\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ be a dynamical system.

(i) A system $\Sigma = (\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ is said to be *linear* if the signal space $\mathbb{W}$ is a vector space and $\mathscr{B}$ is a linear subspace of $\mathbb{W}^{\mathbb{Z}_{\geq 0}}$.

(ii) A system $\Sigma = (\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ 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 $\Sigma = (\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ is said to be *complete* if:
$w\,|_{[t_0,t_1]} \in \mathscr{B}\,|_{[t_0,t_1]},\ \forall\, t_0, t_1 \in \mathbb{T},\ t_0 \leq t_1 \quad \Rightarrow \quad w \in \mathscr{B}$

The class of systems $(\mathbb{Z}_{\geq 0}, \mathbb{R}^{m+p}, \mathscr{B})$ satisfying (i)-(iii) is denoted by $\mathcal{L}^{m+p}$, with $m, p \in \mathbb{Z}_{\geq 0}$. Any trajectory $w \in \mathscr{B}$ can be written as $w = \mathrm{col}(u, y)$, where $\mathrm{col}(u, y) := (u^T, y^T)^T$ [15].

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 $L, T \in \mathbb{Z}_{>0}$ such that $T \geq L$. The signal $u = \mathrm{col}(u_1, ..., u_T) \in \mathbb{R}^{Tm}$ 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 a 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 (5-1) where $\mathscr{B} \in \mathcal{L}^{m+p}$ is represented by $\mathscr{B}(A,B,C,D) = \text{col}(u,y) \in (\mathbb{R}^{m+p})^{(\mathbb{Z}_{\geq 0})} \quad | \quad \exists\, x \in (\mathbb{R}^n)^{(\mathbb{Z}_{\geq 0})}$ s.t. $x(k+1) = Ax(k) + Bu(k), y(k) = Cx(k) + Du(k)$. 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 $l \in \mathbb{Z}_{>0}$ 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 as follows:

$$\mathscr{T}_N(A,B,C,D) := \begin{pmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ CA^{n-2}B & \cdots & CB & D \end{pmatrix} \tag{5-26}$$

We can now present two useful lemmas:

Lemma 4.1 ([83], 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 $T_{ini}, N \in \mathbb{Z}_{>0}$ with $T_{ini} \geq l(\mathscr{B})$ and $\text{col}(u_{ini}, u, y_{ini}, y) \in \mathscr{B}_{T_{ini}+N}$. Then there exists a unique $x_{ini} \in \mathbb{R}^{\mathbf{n}(\mathscr{B})}$ such that:

$$y = \mathcal{O}_N(A,C)x_{ini} + \mathscr{T}_N(A,B,C,D)u \tag{5-27}$$

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

Lemma 4.2 ([84], Theorem 1): Consider a controllable system $\mathscr{B} \in \mathcal{L}^{m+p}$ and let $T, t \in \mathbb{Z}_{>0}$ 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 system identification procedure.

5-4-2 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 $T, T_{ini}, N \in \mathbb{Z}_{>0}$ such that $T \geq (m+1)(T_{ini} + N + \mathbf{n}(\mathscr{B})) - 1$. Then in an offline procedure a sequence of T inputs $u^d = \text{col}(u_1^d, ..., u_T^d) \in \mathbb{R}^{Tm}$ is applied to the unknown system and the corresponding outputs are $y_d = \text{col}(y_1^d, ..., y_T^d) \in \mathbb{R}^{Tp}$. Next the data is separated into a past and future part:

$$\begin{pmatrix} U_p \\ U_f \end{pmatrix} := \mathscr{H}_{T_{ini}+N}(u^d), \quad \begin{pmatrix} Y_p \\ Y_f \end{pmatrix} := \mathscr{H}_{T_{ini}+N}(y^d) \tag{5-28}$$

where U_p consists of the first T_{ini} block rows of $\mathscr{H}_{T_{ini}+N}(u^d)$ and U_f consists of the last N block rows of $\mathscr{H}_{T_{ini}+N}(u^d)$ (similar for Y_p and Y_f). The past data U_p and Y_p will be used to implicitly estimate the initial state whereas the future data U_f and Y_f will be used to predict the future trajectories of the system.

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

$$\begin{pmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{pmatrix} \tag{5-29}$$

Now using the result of Lemma 4.1: if $T_{ini} \geq l(\mathscr{B})$, Lemma 4.1 implies that there exists a unique $x_{ini} \in \mathbb{R}^{\mathbf{n}(\mathscr{B})}$ such that y is uniquely determined by (5-27). Hence, when solving the first three block rows of (5-29) for g, a unique output y can be computed based on inputs u and the initial trajectory $\text{col}(u_{ini}, y_{ini})$.

Next, the DeePC algorithm will be formulated. Consider the following optimal control problem:

$$\begin{aligned} \min_{g,u,y} \quad & \sum_{k=0}^{N-1} (\|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2) \\ \text{subject to} \quad & \begin{pmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{pmatrix}, \\ & u_k \in \mathcal{U}, \quad \forall k \in \{0, ..., N-1\}, \\ & y_k \in \mathcal{Y}, \quad \forall k \in \{0, ..., N-1\}. \end{aligned} \tag{5-30}$$

Where $N \in \mathbb{Z}_{>0}$ is the time horizon, $r = (r_0, r_1, ...) \in (\mathbb{R}^p)^{\mathbb{Z}_{\geq 0}}$ is a reference trajectory, $\text{col}(u_{ini}, y_{ini}) \in \mathscr{B}_{Tini}$ 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 $R \in \mathbb{R}^{m \times m}$ is the control cost matrix and $Q \in \mathbb{R}^{p \times p}$ is the output cost matrix.

In Algorithm 3, the DeePC algorithm is shown.

Algorithm 3: DeePC

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

1) Solve (5-30) 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.

5-4-3 Extension Beyond Deterministic LTI Systems

In the DeePC algorithm as stated in (5-30) there is assumed that the data originates from a deterministic LTI system. However, all systems are stochastic and non-linear by nature. Hence, it is desirable to extend the DeePC algorithm to this case. Now consider the non-linear discrete time system (5-5) with only measurement noise ($w(t) = 0$). In order to

apply the DeePC algorithm to this stochastic nonlinear system, the authors propose three regularizations to the original control problem (5-30), resulting in the following regularized control problem:

$$\begin{aligned} \min_{g,u,y} \quad & \sum_{k=0}^{N-1} (\|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2) + \lambda_g \|g\|_1 + \lambda_y \|\sigma_y\|_1 \\ \text{subject to} \quad & \begin{pmatrix} \hat{U}_p \\ \hat{Y}_p \\ \hat{U}_f \\ \hat{Y}_f \end{pmatrix} g = \begin{pmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ \sigma_y \\ 0 \\ 0 \end{pmatrix}, \\ & u_k \in \mathcal{U}, \quad \forall k \in \{0, ..., N-1\}, \\ & y_k \in \mathcal{Y}, \quad \forall k \in \{0, ..., N-1\}. \end{aligned} \tag{5-31}$$

where $\sigma_y \in \mathbb{R}^{T_{ini}p}$ is an auxiliary slack variable, $\lambda_y, \lambda_g \in \mathbb{R}_{>0}$ are regularization parameters and $\text{col}(\hat{U}_p, \hat{Y}_p, \hat{U}_f, \hat{Y}_f)$ is a low-rank approximation of $\text{col}(U_p, Y_p, U_f, Y_f)$.

The three regularizations are described below:

Slack variable σ_y: Due to the measurement noise, the equality constraint in (5-30) might become infeasible. Hence, the slack variable σ_y is included to make sure the equality constraint is always feasible. The slack variable is penalized with a one-norm penalty function weighted by λ_y. By choosing λ_y sufficiently large, the slack variable will only be non-zero if the equality constraint is infeasible.

One-norm regularization on g: The vector g is penalized with a one-norm weighted by λ_g. This regularization is applied in order to get a robust mapping of the equality constraints from (5-31) compared with the equality constraints from (5-30).

Low-rank approximation: By applying a low-rank approximation to the data Hankel matrices U_p, Y_p, U_f and Y_f (e.g., SVD) and truncation after the most significant components, only the most dominant behaviour is taken into account. The low-rank approximation is able to filter the noise and approximate an LTI model with only the most dominant behaviour of the nonlinear system. Furthermore, after the low-rank approximation the resulting data matrices $(\hat{U}_p, \hat{Y}_p, \hat{U}_f, \hat{Y}_f)$ do not have the Hankel structure anymore and the extended DeePC algorithm does not require this whereas in several subspace identification algorithms it is necessary to preserve the particular Hankel structure [85].

Stability and robustness: In [86] an extension of the DeePC algorithm is done by proving exponential stability of the optimal control problem (5-30) with additional terminal equality constraints in the case of no measurement noise. Furthermore, they propose a robust modification for bounded additive output measurement noise of which they prove exponential stability of the closed loop with respect to the noise level.

5-5 Conclusion

Model based control techniques are widely used in the field of control. However, as shown in previous chapters, difficulties arise with deriving the models on which these control algorithms rely. Techniques such as system identification provide a way to derive a model from input-output data of the system after which these identified models could be used in a model based control algorithm. Furthermore, SPC provides a way to simultaneously perform the identification and control steps of a model. However, these identified models still rely on an underlying parametric model. For complex and highly nonlinear (or unknown) systems the assumption on a particular parametric model might not be valid. Therefore, it might be desirable to control these systems based on their input-output data such that the complex dynamics could be implicitly taken into account. This is done in e.g., DeepMPC where first a NN is trained based on input-output data of the system. Thereafter the NN is included in an NMPC algorithm which computes the optimal control input. However, in this procedure still an NN should be trained in order to obtain a model of the system, which comes with its own difficulties [87].

DeePC provides a way to control an unknown system in a data-driven way. This algorithm does not need a parametric system description but solely relies on input-output data of the system. Hence, for systems with complex and highly nonlinear dynamics the algorithm provides a way to avoid the system identification step and opens the way to control a system purely based on measured data. In this way, difficulties arising in model based control for complex nonlinear systems could be overcome.

Particularly for greenhouses it turned out to be difficult to derive an explicit system description. Therefore, DeePC is a promising solution when it comes to greenhouse control in order to obtain an optimal greenhouse control policy based on output feedback of the greenhouse system. Hence, the final problem formulation in the next chapter will be based on these findings.

Chapter 6

Conclusion and Problem Formulation

6-1 Concluding Remarks

In this chapter, the research questions for the subsequent thesis will be formulated based on this literature study. Within this study it was found that it is a difficult task to derive an explicit model for the greenhouse climate due to its complex and highly nonlinear dynamics. The models currently developed often rely on physical laws such as conservation of mass and energy in which the unknown variables are identified by using input-output data of the system. However, greenhouses with e.g., different size, volume, geometry, material, cultivated crops, equipment, window geometry or window type exhibit different behaviour. Hence, models identified at a particular greenhouse might not be sufficiently reliable to use for control in another greenhouse. Besides, often the crop status is not included in the control loop. Hence, in order to reach optimal crop growth, feedback on the crop status should be taken into account.

Furthermore, many state-of-the-art control algorithms still rely on these models, e.g., Nonlinear Model Predictive Control (NMPC). Therefore, a solution to overcome the arising difficulties while applying model based control, is to replace the model based control with a data-driven control algorithm (e.g., Data-Enabled Predictive Control (DeePC)). This algorithm, introduced in Chapter 5, does not rely on a parametric system representation but only uses input-output data of the system. Therefore, this algorithm makes it possible to control a system when only a finite amount of input-output data is available. Hence, instead of separately identifying a model for each greenhouse, it suffices to collect enough input-output data of the system. This results in an algorithm which is simpler to implement compared to learning techniques or model based control techniques since no system identification is necessary here. However, the original DeePC algorithm is designed for deterministic LTI systems but by adding regularizations, a slack variable and a low-rank approximation extension to the stochastic nonlinear case is shown. Hence, for greenhouse control the DeePC algorithm provides a promising way to overcome the issues which arise in model based control. Therefore, we arrive at the following Problem Formulation.

6-2 Problem Formulation

6-2-1 Main Research Question

Based on the conclusions drawn in the previous section, the Main Research Question for the subsequent thesis will be:

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

6-2-2 Sub-Research Questions

In order to support the Main 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?*

6-2-3 Optional Research Questions

Furthermore, if time allows it, the following Optional Research Questions could be addressed as well:

- *How to extend the DeePC algorithm and its regularizations to the stochastic and non-linear system case?*
- *How to guarantee robustness when applying the DeePC algorithm to stochastic and non-linear systems?*

Future work recommendations from [86] which could perhaps be addressed as well are:

- *How to extend the DeePC scheme to output constraints?*
- *How to consider artificial setpoints in the DeePC framework?*

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

[10] 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.

[11] K. Goldberg, "Robots and the return to collaborative intelligence," *Nature Machine Intelligence*, vol. 1, no. 1, pp. 2–4, 2019.

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

[13] 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, p. 1807, 2019.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

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

[20] 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.

[21] J. B. Cunha, "Greenhouse climate models: An overview," *EFITA Conference*, pp. 5–9, 2003.

[22] G. Bot, *Greenhouse climate : from physical processes to a dynamic model*. PhD thesis, Landbouwhogeschool Wageningen, 1983.

[23] H. Zwart, *Analyzing energy-saving options in greenhouse cultivation using a simulation model*. PhD thesis, Landbouwuniversiteit Wageningen, 1996.

[24] J. Pieters, J. Deltour, and M. Debruyckere, "Condensation and dynamic heat transfer in greenhouses, part i: theoretical model," *Agricultural Engineering Journal*, vol. 5, pp. 119–133, 01 1996.

[25] E. van Henten, *Greenhouse Climate Management: An Optimal Control Approach*. PhD thesis, Wageningen Agricultural University, 1994.

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

[27] H. Frausto, J. Pieters, and J. Deltour, "Modelling greenhouse temperature by means of auto regressive models," *Biosystems Engineering*, vol. 84, no. 2, pp. 147 – 157, 2003.

[28] H. Frausto and J. Pieters, "Modelling greenhouse temperature using system identification by means of neural networks," *Neurocomputing*, vol. 56, pp. 423 – 428, 2004.

[29] A. Calderón and I. González, "Neural networks-based models for greenhouse climate control," pp. 875–879, 2018.

[30] M. El Ghoumari, H. Tantau, D. Megas, and J. Serrano, "Real time non linear constrained model predictive control of a greenhouse," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 61–65, 2002.

[31] M. E. Ghoumari, H.-J. Tantau, and J. Serrano, "Non-linear constrained mpc: Real-time implementation of greenhouse air temperature control," *Computers and Electronics in Agriculture*, vol. 49, no. 3, pp. 345 – 356, 2005.

[32] D. Xu, S. Du, and G. van Willigenburg, "Double closed-loop optimal control of greenhouse cultivation," *Control Engineering Practice*, vol. 85, pp. 90 – 99, 2019.

[33] J. Roy, T. Boulard, C. Kittas, and S. Wang, "PA-precision agriculture: Convective and ventilation transfers in greenhouses, part 1: the greenhouse considered as a perfectly stirred tank," *Biosystems Engineering*, vol. 83, no. 1, pp. 1 – 20, 2002.

[34] C. Gary, J. Jones, and M. Tchamitchian, "Crop modelling in horticulture: State of the art," *Scientia Horticulturae*, vol. 74, no. 1, pp. 3–20, 1998.

[35] K. J. Boote, J. W. Jones, G. Hoogenboom, and N. B. Pickering, *The CROPGRO model for grain legumes*, pp. 99–128. Springer Netherlands, 1998.

[36] 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, 2019.

[37] J. Jones, E. Dayan, L. Allen, Keulen, and H. Challa, "A dynamic tomato growth and yield model (TOMGRO)," *Transactions of the ASAE*, vol. 34, no. 2, pp. 663–672, 1991.

[38] E. Heuvelink, *Tomato growth and yield: quantitative analysis and synthesis*. PhD thesis, Landbouwuniversiteit Wageningen, 1996.

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

[40] P. van Beveren, J. Bontsema, G. van Straten, and E. van Henten, "Optimal control of greenhouse climate using minimal energy and grower defined bounds," *Applied Energy*, vol. 159, pp. 509 – 519, 2015.

[41] E. V. Henten and J. Bontsema, "Time-scale decomposition of an optimal control problem in greenhouse climate management," *Control Engineering Practice*, vol. 17, no. 1, pp. 88 – 96, 2009.

[42] C. Lijun, D. Shangfeng, H. Yaofeng, and L. Meihui, "Linear quadratic optimal control applied to the greenhouse temperature hierarchal system," *International Federation of Automatic Control*, vol. 51, no. 17, pp. 712 – 717, 2018.

[43] 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.

[44] L. Chen, S. Du, Y. He, M. Liang, and D. Xu, "Robust model predictive control for greenhouse temperature based on particle swarm optimization," *Information Processing in Agriculture*, vol. 5, no. 3, pp. 329 – 338, 2018.

[45] J. Gruber, J. Guzmán, F. Rodríguez, C. Bordons, M. Berenguel, and J. Sánchez, "Nonlinear mpc based on a volterra series model for greenhouse temperature control using natural ventilation," *Control Engineering Practice*, vol. 19, no. 4, pp. 354 – 366, 2011.

[46] W. I. R. Agmail, R. Linker, and A. Arbel, "Robust control of greenhouse temperature and humidity," *IFAC Proceedings Volumes*, vol. 42, no. 6, pp. 138 – 143, 2009.

[47] G. Pasgianos, K. Arvanitis, P. Polycarpou, and N. Sigrimis, "A nonlinear feedback technique for greenhouse environmental control," *Computers and Electronics in Agriculture*, vol. 40, no. 1, pp. 153 – 177, 2003.

[48] S. Revathi and N. Sivakumaran, "Fuzzy based temperature control of greenhouse," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 549 – 554, 2016.

[49] J. del Sagrado, J. Sánchez, F. Rodríguez, and M. Berenguel, "Bayesian networks for greenhouse temperature control," *Journal of Applied Logic*, vol. 17, pp. 25 – 35, 2016.

[50] F. Rodriguez, M. Berenguel, J. Guzmán, and A. Arias, *Modeling and Control of Greenhouse Crop Growth*. Springer International Publishing, 2015.

[51] M. Tchamitchian, C. Kittas, T. Bartzanas, and C. Lykas, "Daily temperature optimisation in greenhouse by reinforcement learning," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 131 – 136, 2005.

[52] M. Tchamitchian and C. Kittas, "An alternative rose crop model for greenhouse temperature control.," p. 13, 2004. 26th Annual Congress, Volos, GRC.

[53] J. Löber, *Optimal trajectory tracking*. PhD thesis, TU Berlin, 2015.

[54] 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.

[55] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in Identification and Control*, (London), pp. 207–226, Springer London, 1999.

[56] 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].

[57] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002.

[58] 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.

[59] T. A. N. Heirung, J. A. Paulson, J. O'Leary, and A. Mesbah, "Stochastic model predictive control - how does it work?," *Computers & Chemical Engineering*, vol. 114, pp. 158 – 170, 2018.

[60] V. Rostampour, P. M. Esfahani, and T. Keviczky, "Stochastic nonlinear model predictive control of an uncertain batch polymerization reactor," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 540 – 545, 2015.

[61] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari, "Use of model predictive control and weather forecasts for energy efficient building climate control," *Energy and Buildings*, vol. 45, pp. 15 – 27, 2012.

[62] F. Behrooz, N. Mariun, M. Marhaban, M. A. Mohd Radzi, and A. Ramli, "Review of control techniques for hvac systems-nonlinearity approaches based on fuzzy cognitive maps," *Energies*, vol. 11, no. 3, p. 495, 2018.

[63] S. J. Qin, "An overview of subspace identification," *Computers & Chemical Engineering*, vol. 30, no. 10, pp. 1502 – 1513, 2006.

[64] M. Moonen, B. De Moor, L. V, and J. Vandewalle, "On- and off-line identification of linear state space models," *International Journal of Control*, vol. 49, no. 1, pp. 219–232, 1998.

[65] M. Verhaegen and P. D. P. Dewilde, "Subspace model identification part 1. the output-error state-space model identification class of algorithms," *International Journal of Control*, vol. 56, no. 5, pp. 1187–1210, 1992.

[66] P. V. Overschee and B. D. Moor, "N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems," *Automatica*, vol. 30, no. 1, pp. 75 – 93, 1994. Special issue on statistical signal processing and control.

[67] R. Boczar, N. Matni, and B. Recht, "Finite-data performance guarantees for the output-feedback control of an unknown system," *2018 IEEE Conference on Decision and Control (CDC)*, Dec 2018.

[68] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Foundations of Computational Mathematics*, Aug 2019.

[69] W. Favoreel, B. D. Moor, and M. Gevers, "SPC: Subspace predictive control," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 4004 – 4009, 1999. 14th IFAC World Congress 1999, Beijing, China, 5-9 July.

[70] P. Van Overschee and B. de Moor, *Subspace Identification for Linear Systems: Theory - Implementation - Applications*. Springer US, 1 ed., 1996.

[71] I. Lenz, R. Knepper, and A. Saxena, "Deepmpc: Learning deep latent features for model predictive control," *Robotics: Science and Systems*, 2015.

[72] K. Bieker, S. Peitz, S. L. Brunton, J. N. Kutz, and M. Dellnitz, "Deep model predictive control with online learning for complex physical systems," *arXiv e-prints*, p. arXiv:1905.10094, 2019.

[73] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[74] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[75] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[76] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[77] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–80, 1997.

[78] H. Youlal and A. Kada, "A class of recurrent neural networks for adaptive control of nonlinear systems," *IFAC Proceedings Volumes*, vol. 27, no. 8, pp. 1393 – 1398, 1994.

[79] T. Gao and Y. Chai, "Improving stock closing price prediction using recurrent neural network and technical indicators," *Neural Computation*, vol. 30, pp. 1–22, 2018.

[80] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *arXiv e-prints*, p. arXiv:1605.05101, 2016.

[81] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *arXiv e-prints*, p. arXiv:1808.03314, 2018.

[82] I. Markovsky, J. Willems, S. van Huffel, and B. de Moor, *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach*. SIAM, 2006.

[83] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *Int. J. Control*, vol. 81, no. 12, pp. 1946–1959, 2008.

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

[85] I. Markovsky, *Low Rank Approximation: Algorithms, Implementation, Applications*. Springer, 2012.

[86] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *arXiv e-prints*, p. arXiv:1906.04679, 2019.

[87] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research - Proceedings Track*, vol. 9, pp. 249–256, 2010.

Glossary

List of Acronyms

MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
SNMPC	Stochastic Nonlinear Model Predictive Control
SPC	Subspace Predictive Control
PAR	Photosynthetically Active Radiation
EC	Electrical Conductivity
CT	Continuous Time
DT	Discrete Time
DeePC	Data-Enabled Predictive Control
SVD	Singular Value Decomposition
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
AI	Artificial Intelligence

List of Symbols

α	Heating Pipe Heat Transfer Coefficient
β	Heat Absorption Efficiency
χ	Ventilation Rate Parameter
η	Radiation Conversion Factor

$\frac{1}{\epsilon+1}$	Fraction of Condensation Heat Transported to Greenhouse Air
$\frac{V_g}{A_g}$	Greenhouse Height
λ	Vaporisation Energy of Water
μ	Fraction of Molar Weight of CO_2 and CH_2O
Φ_v	Natural Ventilation Flux
ψ	Ventilation Rate Parameter
σ	Ventilation Rate Parameter
φ	Water Flow Heating Pipe
φ_{inj}	CO_2 Injection Flux
ξ	Ventilation Rate Parameter
ζ	Ventilation Rate Parameter
A_p	Surface Area Heating Pipe
C_g	Specific Heat Capacity Greenhouse Air
C_i	CO_2 concentration
C_o	Outside CO_2 Concentration
C_p	Specific Heat Water
C_s	Greenhouse Soil Heat Capacity
d	Exogenous Signals
E	Rate of Crop Transpiration
G	Incoming Short-wave Solar Radiation
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
k_d	Deep Soil Heat Transfer Coefficient
k_r	Greenhouse Cover Heat Transfer Coefficient
k_s	Soil Heat Transfer Coefficient
k_v	Ventilation Heat Transfer Coefficient
l_1	Vaporisation Energy Coefficient
l_2	Vaporisation Energy Coefficient
M_c	Condensation Water Mass Flow
P	Photosynthesis Rate
R	Crop Respiration
r_{wl}	Relative Window Opening Lee Side
r_{ww}	Relative Window Opening Wind Side
S_c	Crop Subsystem
S_g	Greenhouse Subsystem
T_d	Temperature Deep Soil
T_g	Greenhouse Temperature
T_o	Outside Temperature

T_p	Heating Pipe Temperature
T_p	Heating Pipe Temperature
T_s	Soil Temperature
T_{pi}	Water Temperature Flowing in the Heating Pipe
T_{po}	Water Temperature Flowing out the Heating Pipe
u	Input Signals
u_H	Heating Water Temperature
u_r	Relative Opening Mixing Valve
$u_{C,\max}$	Maximum CO_2 Supply Rate
$u_{C,\sup}$	CO_2 Supply Rate
$u_{I,\text{fert}}$	EC of Irrigation Water
$u_{I,\text{PAR}}$	PAR Light per Shot
$u_{I,\text{pause}}$	Maximum Time Interval between Shots
$u_{I,\text{shot}}$	Irrigation Shot Size
$u_{L,\text{glob}}$	Solar Radiation Threshold
$u_{L,\text{hrL}}$	Maximum Hours of Light
$u_{L,\text{hrT}}$	Time to Switch Off Lights
$u_{L,\text{int}}$	Light Intensity
$u_{L,\text{maxPAR}}$	PAR threshold
$u_{S,\text{blackout}}$	Relative Opening Blackout Screen
$u_{S,\text{energy}}$	Relative Opening Energy Screen
$u_{S,\text{maxR}}$	Maximum Radiation Shading Screen
$u_{S,\text{minR}}$	Minimum Radiation Energy Screen
$u_{S,\text{minT}}$	Minimum Temperature Energy Screen
$u_{S,\text{shading}}$	Relative Opening Shading Screen
$u_{V,\text{Hband}}$	Humidity Band
$u_{V,\max}$	Maximum Window Opening
$u_{V,\min}$	Minimum Window Opening
$u_{V,\text{Pband}}$	Ventilation P-band
$u_{V,\text{vent}}$	Ventilation Offset
$u_{V,RH}$	Maximum Relative Humidity
V_i	Greenhouse Absolute Humidity
V_o	Outside Absolute Humidity
V_p	Volume Heating Pipe
w	Wind Speed
x_c	Crop States
x_g	Greenhouse States