Predictive Control of Autonomous Greenhouses
A Data-Driven Approach

Kerkhof, L.; Keviczky, T.

DOI
10.23919/ECC54610.2021.9655228

Publication date
2021

Document Version
Final published version

Published in
Proceedings of the European Control Conference (ECC 2021)

2021 European Control Conference (ECC)
June 29 - July 2, 2021. Rotterdam, Netherlands

Predictive Control of Autonomous Greenhouses: A Data-Driven Approach

L. Kerkhof[1] and T. Keviczky[2]

Abstract— **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 nonlinear dynamics of greenhouses, these models might not be applicable to control greenhouses other than the ones for which these models have been designed and identified. Hence, in current horticultural practice these control algorithms are scarcely used. Therefore, the need rises for a control algorithm which does not rely on a parametric system representation but rather on input/output data of the greenhouse system, hereby establishing a way to control the system with unknown or unmodeled dynamics. A recently proposed algorithm, Data-Enabled Predictive Control (DeePC), is able to replace system identification, state estimation and future trajectory prediction by one single optimization framework. The algorithm exploits a non-parametric model constructed solely from input/output data of the system. In this work, we apply this algorithm in order to control the greenhouse climate. It is shown that in numerical simulation the DeePC algorithm is able to control the greenhouse climate while only relying on past input/output data. The algorithm is bench-marked against the Nonlinear Model Predictive (NMPC) algorithm in order to show the differences between a predictive control algorithm that has direct access to the nonlinear greenhouse simulation model and a purely data-driven predictive control algorithm. Both algorithms are compared based on reference tracking accuracy and computational time. Furthermore, it is shown in numerical simulation that the DeePC algorithm is able to cope with changing dynamics within the greenhouse system throughout the crop cycle.**

I. INTRODUCTION

The world population is increasing rapidly. According to the United Nations, the world population will increase to nearly 10 billion people in 2050 [1]. Recent estimates report an amount of 821 million people 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 such as improving crop quality and decreasing the crop cultivation period [3]. Furthermore, the greenhouse provides protection against insects, pests and diseases.

An important concern within the greenhouse industry is sustainability. According to the European Union, the share of energy consumption of agriculture with respect to the total energy consumption was 8.2% in 2017 in the Netherlands [4]. In addition, the 2020 CO_2 emission target of 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 decrease the total energy and CO_2 consumption in the greenhouse industry, leading to a more sustainable horticultural sector.

Another important problem within the greenhouse industry is finding sufficient experienced labor to manage crop production because the amount of experienced growers worldwide is declining [6]. A solution to overcome the shortfall of experienced labor is to increase the level of autonomy in greenhouse crop production. As the worldwide greenhouse vegetable production is increasing [7] and the greenhouse equipment is becoming more advanced [8], the necessity for advanced greenhouse control algorithms grows.

To increase the level of autonomy in the greenhouse industry, various model based control algorithms, e.g., Nonlinear Model Predictive Control (NMPC), have been developed [9]. However, these algorithms usually exploit an explicit parametric model of the system dynamics. Difficulties in modelling the greenhouse dynamics arise in the fact that greenhouses exhibit complex and nonlinear behaviour [10]. Data-driven control algorithms such as Data-Enabled Predictive Control (DeePC) [11] could overcome these difficulties since such algorithms do not rely on an explicit parametric system representation but rather on input-output data of the system. Therefore, in this paper the purely data-driven DeePC algorithm is applied on a non-linear greenhouse simulator and its performance is bench-marked against the NMPC algorithm which has direct access to the greenhouse simulator.

II. GREENHOUSE MODEL

A. Greenhouse Climate Model

In this section, a description of the greenhouse climate system is given. Fig. 1 schematically shows how the greenhouse climate interacts with the control inputs, the external environment and the crop. In this scheme, the greenhouse climate and crop are considered as two separate subsystems: S_g and S_c, with states x_g and x_c, respectively.

[1]L. Kerkhof is Control Engineer at Hoogendoorn Growth Management, Vlaardingen, The Netherlands lk@hoogendoorn.nl

[2]T. Keviczky is with the Delft Center of Systems and Control, Delft University of Technology, Delft, The Netherlands. t.keviczky@tudelft.nl

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

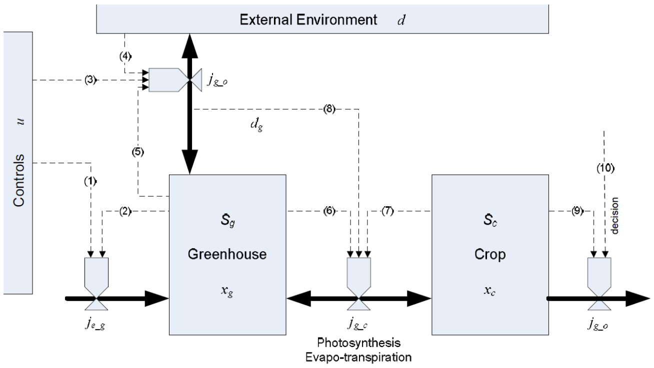

Fig. 1. Block scheme of the greenhouse climate system [3].

Mass and energy fluxes between four different parts are distinguished: between the control equipment and the greenhouse (j_{e_g}), between the greenhouse and the outside environment (j_{g_o}), between the greenhouse and the crop (j_{g_c}) and between the crop and output (j_{c_o}). Mass fluxes are e.g., the water and CO_2 fluxes from and to the crop and energy fluxes are e.g., the heating and solar energy fluxes from and to the greenhouse. The information flows, denoted by (1), ... , (10) are described below:

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

B. Greenhouse Crop Model

In Fig. 2, a block scheme is shown which shows the main processes involved in the growth of a tomato crop. The diagram starts with photosynthesis (1) where solar radiation, water and CO_2 are converted to assimilates. The assimilates produced by photosynthesis are transferred (p) to the assimilate buffer (4) where accumulation of assimilates takes place. From this buffer assimilates are again transferred (g_r, g) in order to be used for growth respiration (2) and distribution (5) among the fruits, stems, leafs and roots as the crop grows. Growth respiration is the process where assimilates are combined with oxygen and converted to energy required for crop growth. After distribution, the assimilates are converted into biomass (6) or used for maintenance respiration (3, m). Finally, the resulting biomass can be harvested in the form of fruits (7, h_1) or leafs (8, h_2).

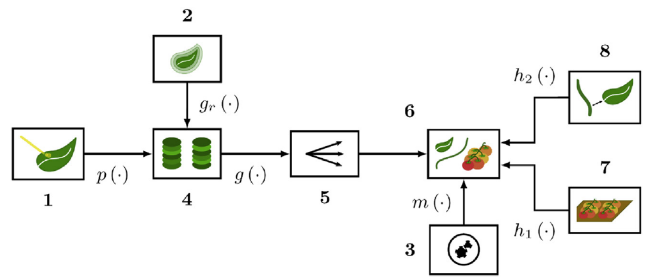

Fig. 2. Block scheme of the greenhouse tomato crop system [12].

C. Discrete Time Model

Models for both the greenhouse climate and crop as described before are obtained from [13]. The states of the greenhouse climate that are considered in this model are the greenhouse air temperature T_g [°C], temperature of the heating pipes T_p [°C], temperature of the soil T_s [°C], CO_2 concentration of the greenhouse air C_i [g/m^3] and the greenhouse air absolute humidity V_i [g/m^3]. The states for the greenhouse crop that are described in this model are the assimilate buffer dry weight per ground area m_{B} [g/m^2], the weight of the fruits per ground area m_{F} [g/m^2], the weight of the leafs per ground area m_{L} [g/m^2] and the crop development stage D [-]. The continuous time models are discretized in order to obtain the following discrete time systems:

$$x_g(t+1) = f_g(x_g(t), x_c(t), u(t), v(t)) \tag{1}$$

$$x_c(t+1) = f_c(x_g(t), x_c(t), v(t), t) \tag{2}$$

In (1), f_{g} denotes the set of equations that describe the dynamics of the greenhouse climate based on the greenhouse climate state x_{g}, greenhouse crop state x_{c}, control inputs u and outside conditions v. In (2), f_{c} denotes the set of equations that describe the dynamics of the greenhouse crop based on the greenhouse climate state, greenhouse crop state and outside conditions. Combining both greenhouse climate and crop states and systems into one system renders the following:

$$x(t+1) = f_{\mathrm{gh}}(x(t), u(t), v(t), t) \tag{3}$$

Where $x = \mathrm{col}(x_{\mathrm{g}}, x_{\mathrm{c}}) := [x_{\mathrm{g}}^T, x_{\mathrm{c}}^T]^T$ and f_{gh} combines the state transition equations from both f_{g} and f_{c}.

III. NONLINEAR MODEL PREDICTIVE CONTROL

In order to control the nonlinear discrete time system, the Nonlinear Model Predictive Control (NMPC) algorithm is applied. NMPC is a predictive control algorithm that makes use of an explicit model of the nonlinear system that it aims to control. At each time step, the algorithm computes the optimal control input signal over a prediction horizon in order to reach a certain objective [14].

In (4) the NMPC reference tracking optimization problem is shown.

$$\begin{aligned} \min_{u} \quad & \sum_{k=0}^{N-1} \|T_{\mathrm{g},k} - r_k\|_Q^2 + \|u_k\|_R^2 + \|u_k - u_{k-1}\|_{R_\Delta}^2 \\ \text{subject to} \quad & x_{k+1} = f_{\mathrm{gh}}(x_k, u_k, v_k), \quad \forall k \in \{0, ..., N-1\}, \\ & x_0 = \hat{x}(t), \\ & u_l \leq u_k \leq u_u, \quad \forall k \in \{0, ..., N-1\}. \end{aligned} \tag{4}$$

Here, $Q \succeq 0$, $R \succ 0$ and $R_\Delta \succeq 0$ are positive (semi-)definite matrices that represent the cost on reference deviation, control input and change of control input, respectively. Besides, $\|u_k\|_R^2$ denotes $u_k^T R u_k$ (similar for the other norms used here). Furthermore, $T_{\mathrm{g},k}$, r_k, x_k, u_k and v_k denote the temperature of the greenhouse air, the temperature reference, the state of the greenhouse climate and crop, the control input and the outside conditions at time index k, respectively. Furthermore, u_l and u_u are the lower and upper bounds of the control input, respectively.

IV. DATA-ENABLED PREDICTIVE CONTROL

A. Data-Enabled Predictive Control

DeePC is a predictive control algorithm that computes optimal control policies using real-time feedback driving the unknown system along a desired trajectory while satisfying system constraints [11]. The algorithm uses a finite number of data samples from the unknown system to create a non-parametric system model that represents the dynamics. This non-parametric system model is subsequently used to implicitly estimate the state and to predict future input/output trajectories of the unknown system. Hence, the DeePC algorithm replaces system identification, state estimation and trajectory prediction by one single optimization framework.

B. Data Collection

DeePC is a data-driven control algorithm. Hence, the first step is to collect data. First, it is assumed that the data is generated by an unknown controllable LTI system represented in the behavioral framework [15] by $\mathscr{B} \in \mathcal{L}^{m+p}$, where $\mathcal{L}^{m+p}$ denotes a *linear*, *time-invariant* and *complete* system with *m* inputs and *p* outputs and $\mathscr{B}$ denotes the behaviour of this system. Let $T, T_{\text{ini}}, T_f \in \mathbb{Z}_{++}$ such that $T \geq (m+1)(T_{\text{ini}} + T_f + \mathbf{n}(\mathscr{B})) - 1$. Here, $\mathbf{n}(\mathscr{B})$ denotes the smallest order of the minimal realisation of the underlying system. Then in an offline procedure a sequence of T inputs $u^d = \text{col}(u_1^d, ..., u_T^d) \in \mathbb{R}^{mT}$ is applied to the unknown system and the corresponding outputs $y^d = \text{col}(y_1^d, ..., y_T^d) \in \mathbb{R}^{pT}$ are collected. Here, the superscript d is used to denote the offline collected data. Next, the data is used to form Hankel matrices with $T_{\text{ini}} + T_f$ block rows, denoted by $\mathscr{H}_{T_{\text{ini}}+T_f}(.)$, and these matrices are separated into a past and future part:

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

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

Now using the result of Lemma 4.2 in [11]: with the collected data, any trajectory of $\mathscr{B}_{T_{\text{ini}}+T_f}$ of length $T_{\text{ini}} + T_f$ could be constructed, assuming the control input data u^d is persistently exciting the system. Let $\text{col}(u_{\text{ini}}, y_{\text{ini}})$ denote an initial trajectory of the system with length T_{ini} and $\text{col}(u_f, y_f)$ a future trajectory of the system of length T_f. It follows that a trajectory $\text{col}(u_{\text{ini}}, u_f, y_{\text{ini}}, y_f)$ belongs to $\mathscr{B}_{T_{\text{ini}}+T_f}$ if and only if there exists $g \in \mathbb{R}^{T-T_{\text{ini}}-T_f+1}$ such that:

$$\begin{pmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_f \\ y_f \end{pmatrix} \tag{6}$$

Now using the result of Lemma 4.1 in [11]: if $T_{\text{ini}} \geq l(\mathscr{B})$, y_f is uniquely determined by solving the first three block rows of (6) for g. Here, $l(\mathscr{B})$ denotes the lag of the underlying system. Hence, a unique output y_f can be computed based on inputs u_f and the initial trajectory $\text{col}(u_{\text{ini}}, y_{\text{ini}})$.

C. Data-Enabled Predictive Control

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

$$\begin{aligned} \min_{g} \quad & \sum_{k=0}^{T_f-1} (\|y_{f,k} - r_{t+k}\|_Q^2 + \|u_{f,k}\|_R^2) \\ \text{subject to} \quad & \begin{pmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_f \\ y_f \end{pmatrix}, \\ & u_k \in \mathcal{U}, \quad \forall k \in \{0, ..., T_f - 1\}, \\ & y_k \in \mathcal{Y}, \quad \forall k \in \{0, ..., T_f - 1\}. \end{aligned} \tag{7}$$

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

The optimization problem in (7) is solved at every time step $t \in \mathbb{Z}_+$ as part of the DeePC scheme described in Algorithm 1.

Algorithm 1: DeePC

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

1) Solve (7) 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.

V. Greenhouse Controlled by DeePC

A. Including Exogenous Signals

In this section, the DeePC algorithm is used to control the greenhouse system. Since the exogenous inputs such as solar radiation and outside temperature have large effects on the states of the greenhouse, these exogenous input signals need to be taken into account as well besides only the control input and output signals. Hence, the DeePC algorithm in (7) needs to be extended in order to include these exogenous signals. In case of the greenhouse system, the exogenous inputs are the outside weather conditions. These conditions are measured in the past and forecasts are available for the future. Therefore, a Hankel matrix is constructed in the same way as is done in (5) but it is built from measured exogenous signal data. Hence, given the recorded data $v^d = \text{col}(v_1, ..., v_T) \in \mathbb{R}^{qT}$ a Hankel matrix is constructed such that:

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

where V_p consists of the first T_{ini} block rows of $\mathscr{H}_{T_{\text{ini}}+T_f}(v^d)$ and V_f consists of the last T_f block rows of $\mathscr{H}_{T_{\text{ini}}+T_f}(v^d)$. In the same way as is done in (6), the past T_{ini} measurements of the external signals are stored in $v_{\text{ini}} \in \mathbb{R}^{qT_{\text{ini}}}$ and the future T_f forecasts of the external signals are stored in $v_f \in \mathbb{R}^{qT_f}$. The equality constraints in (6) are then augmented with the aforementioned data matrices and vectors such that:

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

B. Extension to Nonlinear Systems

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

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

C. Window Constraint

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

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

Here, $U_{f,\text{lee}}$ and $U_{f,\text{wind}}$ denote the rows of the U_f Hankel matrix that correspond to the lee side and wind side window openings, respectively.

D. Regularized DeePC

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

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

Here, $\lambda_y \in \mathbb{R}_+$ and $\lambda_v \in \mathbb{R}_+$ are regularization parameters on the constraint violations and $\sigma_v = [\sigma_{v_1}^T, \sigma_{v_2}^T]^T$. The optimization problem in (12) is subsequently solved at each time step $t \in \mathbb{Z}_+$ as part of the extended regularized DeePC scheme shown in Algorithm 2.

VI. Case-Study: NMPC vs DeePC

In this section a case study will be performed in order to compare the NMPC and DeePC algorithms. The objective for both algorithms will be to track a temperature reference with the greenhouse air temperature on a similar day. From this

Algorithm 2: Extended regularized DeePC

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

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

day, the recorded weather conditions will be used as exogenous inputs. Afterwards, both algorithms will be compared based on tracking accuracy and computational time.

In practice it is often not desired or even impossible to apply a random generated excitation signal to a system. In case of the greenhouse system, the greenhouse crop is sensitive to rapid changes in greenhouse air temperature which may occur by applying a random control input. Therefore, a control input sequence will be generated using NMPC in order to represent past operational data from previous days such that a safe input can be applied to the greenhouse in order to capture its dynamics.

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

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

A. Weather conditions and control input signals

Identical weather conditions are used in order to compare both algorithms. These weather conditions are shown in Fig. 3. This data is used to construct the data matrices V_p and V_f in (6). Besides, in Fig. 3 is shown which part of the data is used to construct the data matrices. The right green line until the beginning is used for V_p and the left green line until the red line is used for V_f as is indicated by the arrows.

Furthermore, in Fig. 4 the NMPC control input signal is shown that is used to construct the Hankel matrices U_p and U_f. The data is selected in the same way as is done in Fig. 3. Furthermore, a necessary condition for (12) to be feasible is that matrix U_p has full rank (persistence of excitation),

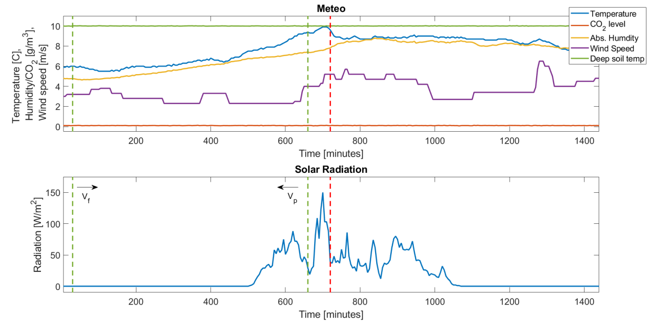

Fig. 3. Measured weather conditions used in the DeePC algorithm.

which holds for this data set. Hence, the control input data in Fig. 4 seems to be appropriate to be used as past recorded control input data for the DeePC algorithm.

Now with both the recorded weather data shown in Fig. 3 and the control input signal shown in Fig. 4, the data is available to generate the output data signals (not shown here) to construct the final two Hankel matrices Y_p and Y_f.

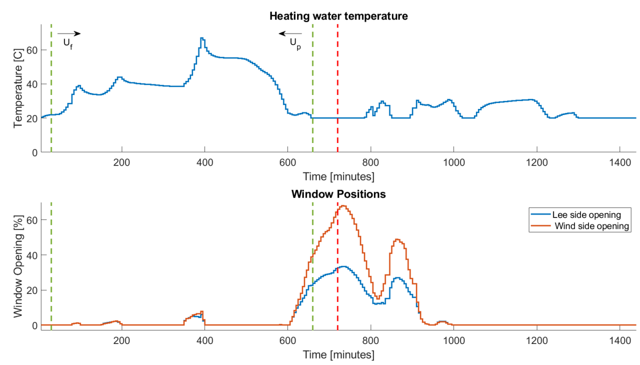

Fig. 4. Generated NMPC input signals used in the DeePC algorithm.

B. Controlling the Greenhouse

Now all required data is available, both algorithms can be used to track a temperature reference. The reference trajectory is chosen such that the 24-hour average temperature is approximately 21 °C, which also occurs in commercial practice. Besides, the trajectory exploits the energy of the sun by having higher temperatures in the middle of the day and lower temperatures at the start and end of the day, hence saving energy. Furthermore, the aim is to minimize the absolute control input to avoid unnecessary energy consumption.

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

For the NMPC algorithm, the following parameters are used: $N = 12$, $Q = \mathrm{diag}(500, 0, ...0,)$, $R = 0.1 I_3$, $\Delta_R = I_3$,

$u_l = [10, 0, 0]^T$ and $u_u = [80, 100, 100]^T$. The parameters are chosen after careful tuning to ensure the best performance of both algorithms. The introduced slack variables modify the cost function of DeePC, hence making both functions not directly comparable. Furthermore, both algorithms are implemented in MATLAB and the SNOPT solver from TOMLAB is used to solve the optimization problems.

In Fig. 5 and Fig. 6, the states and control inputs computed by both NMPC and DeePC are shown. From these figures it can be seen that both algorithms are able to accurately track the desired reference. Furthermore, a noticeable difference is that the NMPC algorithm returned smoother control input signals compared with the DeePC algorithm.

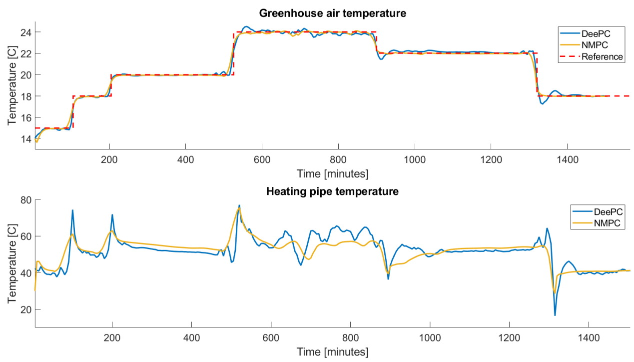

Fig. 5. Greenhouse air and heating pipe temperatures controlled by both DeePC and NMPC.

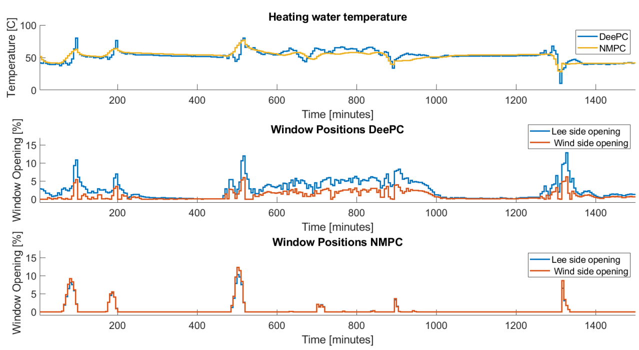

Fig. 6. Heating water temperature and window positions computed by both DeePC and NMPC.

C. Comparison of Algorithm Performance

In Table I the tracking accuracy, average computation time and total computation time are shown for both the NMPC and DeePC algorithms. The RMSE metric is used to calculate the tracking accuracy. From this table can be seen that the NMPC algorithm has a higher reference tracking accuracy compared to the DeePC algorithm. Furthermore, the average computation time for NMPC is approximately 30 times smaller. However, since the sampling time is 5 minutes, the DeePC algorithm would still be able to control the greenhouse in real-time.

TABLE I

TRACKING ACCURACY AND COMPUTATION TIME OF NMPC AND DEEPC.

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

VII. ADJUSTED GREENHOUSE DYNAMICS

This section presents the results of two numerical simulations that illustrate the effect of initialization data on performance of the DeePC algorithm. In the first simulation the input/output data in the Hankel matrices will be the same as in the previous example. This data was obtained in a period where the crop was still young. However, in this simulation the state of the crop will be initialized in such a way that it represents a full grown crop. Hence, this will influence the dynamics of the greenhouse climate while these effects are not present in the data stored in the Hankel matrices. This is done in order to show the sensitivity of the DeePC algorithm when data is used that is less representative for the current dynamics of the system. Hence, in the second simulation the Hankel matrices will be updated with data from the greenhouse where the crop is fully developed. With these updated data matrices the algorithm is used again to control the greenhouse climate to show that the DeePC algorithm is able to control the system while the dynamics have changed. Within these examples, the same values as in Section VI-B are used.

In Table II, the tracking accuracy of both simulations are shown. The tracking accuracy of the simulation with the new data is comparable with the tracking accuracy from the simulation in the previous example. However, the RMSE in case of the simulation with the old data has almost doubled compared to the case with the new data. This shows the effect that where old data does not accurately represent the dynamics of the system anymore whereas the updated data does. The computation time of both numerical simulations here are roughly equal. In Fig. 7 and Fig. 8 the results from these numerical simulations are shown. As can be seen from these figures the DeePC algorithm is sensitive to time-varying systems when the data in the Hankel matrices becomes less representative for the current greenhouse dynamics. However, when the data is updated with recent input/output data, the algorithm is able to recover and control the system accordingly. In these figures the simulation of the DeePC algorithm with the data of the young crop is indicated as 'DeePC; Old data' and the simulation with the updated data is indicated as 'DeePC; New data'.

TABLE II

TRACKING ACCURACY OF BOTH DEEPC SIMULATIONS.

	DeePC; Old data	DeePC; New data
Tracking accuracy [RMSE]	0.8045	0.3800

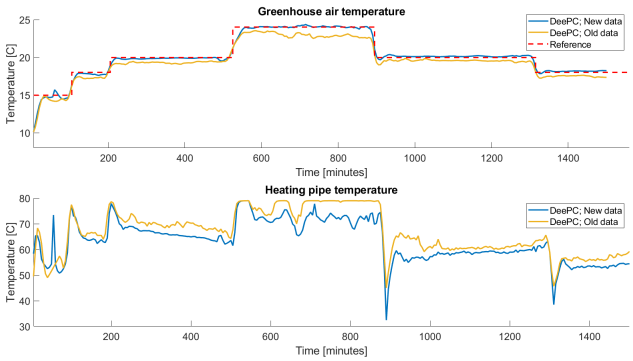

Fig. 7. Greenhouse air and heating pipe temperatures controlled by DeePC based on both the old and updated data.

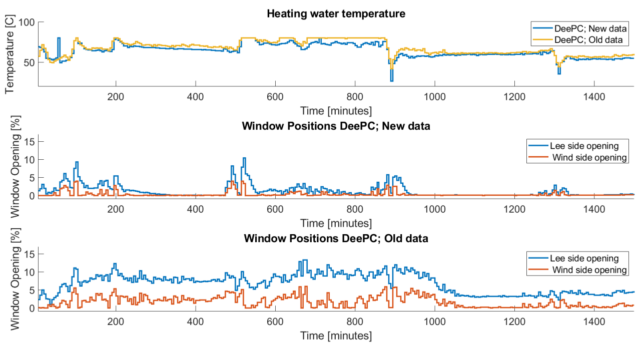

Fig. 8. Heating water temperature and window positions computed by DeePC based on both the old and updated data.

VIII. CONCLUSION

In this paper, a data-driven predictive control algorithm, DeePC, was applied in order to control a greenhouse system. The performance of the DeePC algorithm was bench-marked to a model based control algorithm, NMPC. The NMPC algorithm had direct access to a greenhouse simulator whereas the DeePC algorithm only relied on a non-parametric model which was built from data from this greenhouse simulator. The goal for both algorithms was to control the greenhouse air temperature by tracking a temperature reference.

In numerical simulations has been shown that the DeePC algorithm was able to track the reference trajectory. The performance in terms of tracking accuracy was lower for the DeePC algorithm, which is not surprising since the NMPC algorithm had direct access to the greenhouse simulator. Hence, the NMPC algorithm could accurately see the effects of the control inputs over which it optimized whereas the DeePC algorithm could only rely on the dynamics of the greenhouse captured in the data that was used to construct the Hankel matrices. Although the computation time of the DeePC algorithm was on average 30 times larger compared to the NMPC algorithm, it would still be able to control the greenhouse system in real-time.

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

IX. FURUTE WORK

Recommendations for future work are to compare the performance of the NMPC and DeePC algorithms in presence of unmodeled conditions, e.g., in a real greenhouse. Furthermore, using stochastic approaches to robustify the algorithms for uncertain weather conditions could be a topic for further research. Finally, using an economic objective in the cost function could be interesting in order to shift the focus to cost-effectiveness instead of reference tracking.

REFERENCES

[1] United Nations, "World Urbanization Prospects: The 2018 Revision", "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", "FAO", "Rome", "2018".

[3] G. van Straten and E. J. van Henten, "Optimal Greenhouse Cultivation Control: Survey and Perspectives", "IFAC Proceedings", 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.html, January, 2018, [Online; accessed 17-Sep-2019].

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

[9] S. Hemming, F. Zwart, A. Elings, I. Righini, A. Petropoulou, "Remote Control of Greenhouse Vegetable Production with Artificial Intelligence - Greenhouse Climate, Irrigation, and Crop Production", Sensors, vol. 19, no. 8, pp. 1-22, 2019.

[10] S. Zeng, H. Hu, L. Xu, G. Li, "Nonlinear Adaptive PID Control for Greenhouse Environment Based on RBF Network", Sensors, vol. 12, no. 5, pp. 5328-5348, 2012.

[11] J. Coulson, J. Lygeros, F. Dörfler, "Data-Enabled Predictive Control: In the Shallows of the DeePC", European Control Conference, pp. 307–312, 2019.

[12] W. Kuijpers, M. van de Molengraft, S. van Mourik, A. van het Ooster, S. Hemming, E. J. van Henten, "Model selection with a common structure: Tomato crop growth models", Biosystems Engineering, vol. 187, pp. 247-257, 2019.

[13] F. Tap, "Economics-based Optimal Control of Greenhouse Tomato Crop Production," Ph.D. dissertation, Systems and Control Group, Wageningen Agricultural University, Wageningen, 2000.

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

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