2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Deep Neural Network Based Optimal Control of Greenhouses

Kiran Kumar Sathyanarayanan[1], Philipp Sauerteig[1], and Stefan Streif[1,2]

***Abstract*— Automatic control of greenhouse crop production is of great interest owing to the increasing energy and labor costs. In this work, we use two-level control, where the upper level generates suitable reference trajectories for states and control inputs based on day-ahead predictions. These references are tracked in the lower level using Nonlinear Model Predictive Control (NMPC). In order to apply NMPC, a model of the greenhouse dynamics is essential. However, the complex nature of the underlying model including discontinuities and nonlinearities results in intractable computational complexity and long sampling times. As a remedy, we employ NMPC as a data generator to learn the tracking control policy using deep neural networks. Then, the references are tracked using the trained Deep Neural Network (DNN) to reduce the computational burden. The efficiency of our approach under real-time disturbances is demonstrated by means of a simulation study.**

I. Introduction

The world population is projected to increase from 7 billion in 2010 to 9.8 billion in 2050, resulting in a 50% increase in food demand [1]. Despite the need to increase food production, it cannot be achieved by expanding agricultural land due to the challenge of ever-growing urban space and climate change. Controlled Environment Agriculture (CEA) like greenhouse farms and indoor vertical farms have gained attention due to their capability to produce high yield in limited space. However, high energy spikes and the need for energy-efficient operations have posed a question on the environmental impact and sustainable operation of CEA [2].

The usage of renewable energy and the development of solar greenhouses is a viable option to achieve green and sustainable food production [3]. With frequently shifting rules and unpredictable energy prices, automatic control systems are the most efficient alternative for greenhouse climate control and crop growth. Several approaches to improve energy efficiency in greenhouses, reduce CO_2 emissions, and enhance water use efficiency have been proposed [4], [5]. Classical control approaches to maintain the pre-defined greenhouse climate are discussed in [6], [7].

Optimal control approaches using mathematical models to maintain greenhouse climate (temperature and humidity) for plant production demonstrating their economical benefits are shown in [8]. Here, Hierarchical Model Predictive Control (HMPC) is employed where the states are divided into different timescales, and the control system is designed for each levels to maintain the greenhouse climate. The upper-level control system generates the optimal climate reference trajectories using the greenhouse model and the weather forecast data, whereas the lower-level control system controls the fast-changing climate by tracking the reference. In [9], multiobjective optimization is proposed to achieve profitable crop growth along with maintaining crop quality and irrigation efficiency using the model of greenhouse and crop growth. Optimal control of a solar greenhouse with an economic objective to increase the yield and decrease the gas usage is discussed in [10]. In general, a longer prediction horizon is considered to generate an optimal control input in NMPC. Assuming a larger sampling time reduces computational demand for longer predictions.

The system equations governed by NMPC are continuous over time and need to be discretized for implementation on embedded or similar devices. The choice of sampling time significantly influences accuracy, making shorter sampling times preferable. However, the discretization method employed to implement optimal control using greenhouse climate and crop models, as well as the impact of sampling time on integration accuracy are not commonly discussed. The effect of small sampling time to be chosen to adequately approximate the system behavior on a given prediction horizon is dealt with in [11]. However, the smaller sampling time demands a shorter horizon for sufficiently fast computation of nonlinear optimization problems. In [12], [13], [14], the problem of higher computation demand of NMPC due to longer prediction horizons has been overcome by the use of DNNs to generate an approximate solution to the NMPC problem. The universal approximation property of the DNNs has motivated the usage of machine learning for optimal control [13], [15].

In this paper, we propose to design a two-level hierarchical control for a semi-closed solar collector greenhouse. The upper-level optimization layer generates the economically optimal input and state reference trajectory for a single day using the weather forecast data, while the lower-level NMPC tracks the above reference trajectory on satisfying the desired bounds with a short-term disturbance forecast. The main advantage is that the approximated NMPC can be deployed in low-cost embedded hardware, which is more desirable in remote locations like refugee camps. Indeed, the approximate DNN based controller can be applied to any CEA system, where fast computation is desirable.

First, a mathematical model describing the greenhouse climate dynamics and crop growth is presented in Section II. To describe the plant growth, we use the tomato model presented in [10]. Following that, Section III introduces the two-level hierarchical control of the greenhouse. Then, Section IV discusses the training of DNN to approximate

The authors are with [1]Technische Universität Chemnitz, 09126 Chemnitz, Germany, Automatic Control and System Dynamics Lab and [2]Fraunhofer Institute for Molecular Biology and Applied Ecology, Department of Bioresources, Giessen, Germany (e-mail: {kiran.sathyanarayanan,philipp.sauerteig,stefan.streif}@etit.tu-chemnitz.de)

the optimal control inputs given the current state, reference and disturbances. Finally, a two-level control of a greenhouse with a lower level of control using trained DNN is implemented and its performance is reviewed in Section V, with the conclusion in Section VI.

II. Greenhouse and Crop Model

This section summarizes the mathematical model of the greenhouse climate along with the crop. For our study, a Venlo-type greenhouse growing tomato crop fitted with a blackout screen, vent opening through the roof, CO_2 injection, heating, and cooling source is considered. In this approach, the greenhouse climate incorporates temperature (T), carbon dioxide concentration (C), and absolute humidity (H_a) of the air inside the greenhouse. A mathematical model of the greenhouse climate is derived via the mass and energy balance equations, see also [8], [16], [17].

A. Dynamic Model of the Greenhouse Climate

1) Temperature: The temperature T of greenhouse air is influenced by various heat fluxes and the energy balance equation describing the same is given by

$$\dot{T} = \left(Q_{\text{sun}} + Q_{\text{vent}} + Q_{\text{cov}} + Q_{\text{trans}} + Q_{\text{heat}} - Q_{\text{cool}}\right)/k_{\text{C,gh}}, \tag{1}$$

where $k_{\text{C,gh}}$ is the total heat capacity of the greenhouse per unit area ($\text{J}\,\text{m}^{-2}\,\text{K}^{-1}$). Moreover, $Q_{\text{sun}} = k_{\text{tot}} \cdot Q_{\text{rad}}$ denotes the heat flux term contributed by the incoming radiation in $\text{W}\,\text{m}^{-2}$, where k_{tot} is the total transmittance of the cover. The heat flux component due to vent opening is given by

$$Q_{\text{vent}} = k_{\text{c,air}} \cdot k_{\rho,\text{air}} \cdot k_{\text{u,vent}} \cdot u_{\text{V}}(T_{\text{out}} - T) \quad [\text{W}\,\text{m}^{-2}],$$

where $k_{\text{c,air}}$ and $k_{\rho,\text{air}}$ denoting the specific heat capacity and density of the air, respectively, and $k_{\text{u,vent}}$ is the maximum ventilation of the greenhouse. Leakage through the cover is assumed to be negligible. The convective heat loss through the cover of the greenhouse is given by

$$Q_{\text{cov}} = k_{\text{A}} \cdot k_{\text{U}} \cdot (T_{\text{out}} - T) \quad [\text{W}\,\text{m}^{-2}],$$

where k_{U} is the coefficient of heat transfer through the cover, k_{A} is the ratio of the surface area of the cover and the greenhouse floor.

We consider the transpiration of tomato crops in a greenhouse given by [17]. Here, Q_{trans} is the heat energy absorbed due to crop evapotranspiration,and can be obtained by

$$Q_{\text{trans}} = g_{\text{e}} k_{\text{L}} (H_{\text{v}} - x_{\text{P,H}}) \quad [\text{W}\,\text{m}^{-2}],$$

where g_{e} is the transpiration conductance, k_{L} is the amount of energy needed to evaporate water from a leaf, and H_{v} is the absolute water vapor concentration at crop/vegetation level. The quantities are computed as follows. The transpiration conductance is $g_{\text{e}} = 2k_{\text{LAI}}/((1+\varepsilon)r_{\text{b}} + r_{\text{s}})\,[\text{m}\,\text{s}^{-1}]$, where k_{LAI} is the leaf area index, ε is the ratio of latent to sensible heat content of saturated air for a change of 1°C in T, r_{b} is the resistance to heat transfer of the leaf boundary layer, and r_{s} is the stomatal resistance. Moreover, H_{v} is calculated as

$$H_{\text{v}} = H_{\text{sat}} + \varepsilon r_{\text{b}}/(2k_{\text{LAI}}) \cdot R_{\text{n}}/k_{\text{L}} \quad [\text{g}\,\text{m}^{-3}],$$

where H_{sat} is the saturated vapor concentration of the greenhouse air that can be approximated by, $H_{\text{sat}} = 5.5638\,\text{e}^{0.0572T}$ for $T \in [10, 40]$. The net radiation R_{n} at crop level, and its relation with the global radiation Q_{rad} is given by

$$R_{\text{n}} = 0.86(1 - \text{e}^{-0.7k_{\text{LAI}}})Q_{\text{rad}} \quad [\text{W}\,\text{m}^{-2}].$$

The heating and cooling power Q_{heat} and Q_{cool}, respectively, in $\text{W}\,\text{m}^{-2}$ are given by $Q_{\text{heat}} = u_{\text{Q}_\text{h}} \cdot \text{P}_{\text{h,max}}$ and $Q_{\text{cool}} = u_{\text{Q}_\text{c}} \cdot \text{P}_{\text{c,max}}$, where $\text{P}_{\text{h,max}}$ and $\text{P}_{\text{c,max}}$ are the maximum heating and cooling power per unit greenhouse floor area, respectively.

2) Carbon-dioxide concentration: The mass balance equation of the CO_2 concentration is described as

$$\dot{C} = k_{\text{A,gh}}/k_{\text{V,gh}} \cdot \left(C_{\text{inj}} + C_{\text{vent}} - C_{\text{phot}}\right), \tag{2}$$

where $k_{\text{V,gh}}$ is the volume of the greenhouse. The pure industrial CO_2 injection rate $C_{\text{inj}} = u_{\text{C}} \cdot C_{\text{max}}/k_{\text{A,gh}}\,[\text{g}\,\text{m}^{-2}\text{s}^{-1}]$. The CO_2 exchange due to ventilation C_{vent} is described as

$$C_{\text{vent}} = k_{\text{u,vent}} \cdot u_{\text{V}}(C_{\text{out}} - C) \quad [\text{g}\,\text{m}^{-2}\text{s}^{-1}].$$

Finally, the net CO_2 consumption due to photosynthesis and transpiration by the crop is given by

$$C_{\text{phot}} = \phi_{\text{CO}_2} \cdot 10^{-3}/k_{\text{A,gh}} \quad [\text{g}\,\text{m}^{-2}\text{s}^{-1}],$$

where ϕ_{CO_2} is defined in Section II-B.

3) Absolute humidity: The absolute humidity of the greenhouse air is influenced by various vapor fluxes and the mass balance equation reads as

$$\dot{H}_{\text{a}} = k_{\text{A,gh}}(H_{\text{trans}} - H_{\text{vent}} - H_{\text{cov}} + H_{\text{heat}} - H_{\text{cool}})/k_{\text{V,gh}}. \tag{3}$$

The amount of water vapor produced due to plant transpiration $H_{\text{trans}} = g_{\text{e}}(H_{\text{v}} - H_{\text{a}})\,[\text{g}\,\text{m}^{-2}\text{s}^{-1}]$. Condensation of vapor to the cover H_{cov} occurs when the temperature of the cover T_{cvr} is below the dew point temperature of the air, and it is calculated as

$$H_{\text{cov}} = g_{\text{C}} \left(0.2522\text{e}^{0.0485T}(T - T_{\text{out}}) - (H_{\text{sat}} - H_{\text{a}})\right) [\text{g}\,\text{m}^{-2}\text{s}^{-1}].$$

Here, condensation conductance g_{C} is described in [18] as,

$$g_{\text{C}} = \max(0, k_{\text{A}}\, 1.64 \times 10^{-3}(T - T_{\text{cvr}})^{\frac{1}{3}}) \quad [\text{m}\,\text{s}^{-1}].$$

The vapor flux due to ventilation H_{vent} and direct air heater H_{heat} are described as

$$H_{\text{vent}} = k_{u,vent} \cdot u_{\text{V}}(H_{\text{a}} - H_{\text{out}}), \quad H_{\text{heat}} = k_{\eta} Q_{\text{heat}} \quad [\text{g}\,\text{m}^{-2}\text{s}^{-1}],$$

where k_{η} $[\text{g}\,\text{J}^{-1}]$ is the amount of vapor released when 1 J of sensible energy is produced by the heater [19]. The cooling of air happens through the heat exchanger, where the pipe inlet is maintained at a constant temperature. The condensation of vapor on the pipes H_{cool} is given by

$$H_{\text{cool}} = \max(0, u_{\text{Q}_\text{h}} k_{\text{A,pipe}} \cdot k_{\text{cool}}(H_{\text{a}} - H_{\text{sat,cool}})),$$

as described in [10]. Here, $k_{\text{A,pipe}}$ is the surface area of the cooling pipe, k_{cool} is the mass transfer coefficient of water vapor from indoor air to the cooling tube, and $H_{\text{sat,cool}}$ is the saturation concentration of water vapor at the temperature of the cooling pipe.

B. Crop Model

Photosynthesis is the process through which crops assimilate carbon dioxide and water to support their maintenance, growth, and development. This process is affected by higher radiation intensity and elevated CO_2 concentration [20]. Here, the CG4 photosynthesis model presented in [10] is used. Furthermore, it is assumed that required water and nutrients for crop growth are available at all times. This model describes the net CO_2 consumption rate of the crop as a function of outdoor radiation Q_{rad}, CO_2 concentration C, temperature of the crop, and humidity H_{a}. As the temperature of the crop is not measured, temperature T of the greenhouse air is used. The consumed CO_2 are converted into carbohydrate assimilate, resulting in the biomass production

$$\dot{B} = (k_{\mathrm{B,CO_2}}\phi_{\mathrm{CO_2}})/k_{\mathrm{A,s}} \quad [\mathrm{kg\,m^{-2}s^{-1}}]. \tag{4}$$

Here, B is the fresh biomass weight, $\phi_{\mathrm{CO_2}}$ is the net photosynthesis rate of the canopy, and $k_{\mathrm{B,CO_2}}$ is the conversion factor to convert the consumed CO_2 to the fresh weight biomass. For further description of the net photosynthesis, see [10].

III. Optimal Greenhouse Control

The dynamics of the greenhouse climate and the crop growth were summarized in (1), (2), (3), and (4). Four control inputs are defined to maintain the states - temperature T, CO_2 concentration C, and relative humidity H of the greenhouse air within the desired bounds while minimizing the production costs for high yields using a cost function.

The system behavior is of the form

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) \tag{5}$$

with state vector $\mathbf{x} = \begin{bmatrix} T & C & RH & B \end{bmatrix}^\top$, input vector $\mathbf{u} = \begin{bmatrix} u_{\mathrm{V}} & u_{\mathrm{C}} & u_{\mathrm{Q_h}} & u_{\mathrm{Q_c}} \end{bmatrix}^\top$, and disturbance vector $\mathbf{d} = \begin{bmatrix} T_{\mathrm{out}} & C_{\mathrm{out}} & H_{\mathrm{out}} & Q_{\mathrm{rad}} \end{bmatrix}^\top$.

The model represented by (5) exhibits discontinuities in the form of maximum operators. To enable various options for optimization algorithms, the maximum operators are approximated as described in [21], and is given as $\max(a,b) \approx \left(a + b + \sqrt{(a-b)^2 + \gamma}\right)/2$, where $\gamma = 10^{-4}$ is a tuning parameter.

In the following, a framework for the optimal control of the greenhouse is implemented. As the task of observer design to get the biomass from other measurements in outlook is omitted due to page brevity, full state availability at every time instance is assumed. The framework includes two levels of control: an open-loop optimization layer and a closed-loop reference tracking Nonlinear Model Predictive Control (NMPC) layer. The upper level solves an economic objective function with the forecast weather data to generate a reference trajectory for the NMPC. Subsequently, the NMPC tracks the obtained reference trajectory in real time.

Optimization problem and predictive control is performed at discrete time step, for which the dynamics

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + \Delta t \cdot f(\mathbf{x}(t_k), \mathbf{u}(t_k), \mathbf{d}(t_k)) \tag{6}$$

are utilized, where $t_k = n_{\mathrm{o}}\Delta t_{\mathrm{o}}$, $n_{\mathrm{o}} \in \mathbb{N}_0$, and $\Delta t_{\mathrm{o}} > 0$ is the sample time for the open loop with T being the total simulation time. Similarly, the closed-loop system is discretized, where $t_k = n\Delta t_{\mathrm{c}}$, $n \in \mathbb{N}_0$, and $\Delta t_{\mathrm{c}} > 0$ is the sample time, with $\Delta t_{\mathrm{o}} > \Delta t_{\mathrm{c}}$. The discretization is performed using orthogonal collocation of degree 4.

A. Open-loop control

The objective is to maximize crop production while minimizing the total operating costs which include costs for heating/cooling, ventilation, and CO_2 injection. To this end, we define the day-ahead cost function

$$J_{\mathrm{o}}(\mathbf{u}) = \sum_{k=0}^{N_{\mathrm{o}}-1} L(\mathbf{x}(t_k), \mathbf{u}(t_k)) - V(\mathbf{x}(t_{N_{\mathrm{o}}})) \tag{7}$$

with $N_{\mathrm{o}} = \frac{\mathsf{T}}{\Delta t_{\mathrm{o}}} \in \mathbb{N}^+$. The stage costs $L(\mathbf{x}(t_k), \mathbf{u}(t_k))$ and terminal costs $V(\mathbf{x}(t_{N_{\mathrm{o}}}))$ and are given by

$$\begin{aligned} L(\mathbf{x}(t_k), \mathbf{u}(t_k)) &= p_{\mathrm{E}} \cdot P_{\mathrm{v}}^{\max} \cdot u_{\mathrm{V}}(t_k) + p_{\mathrm{C}} \cdot C_{\mathrm{inj}}^{\max} \cdot u_{\mathrm{C}}(t_k) \\ &\quad + p_{\mathrm{E}} \cdot P_{\mathrm{Q_h}}^{\max} \cdot u_{\mathrm{Q_h}}(t_k) + p_{\mathrm{E}} \cdot P_{\mathrm{Q_c}}^{\max} \cdot u_{\mathrm{Q_c}}(t_k) \\ &\quad + \mathsf{P}_T(\mathbf{x}) + \mathsf{P}_C(\mathbf{x}) + \mathsf{P}_{RH}(\mathbf{x}) \\ V(\mathbf{x}(t_{N_{\mathrm{o}}})) &= p_{\mathrm{y}} \cdot k_{\mathrm{B_{fruit}}} \cdot k_{\mathrm{A,s}} \cdot B(t_k). \end{aligned} \tag{8}$$

Here, penalties for temperature $\mathsf{P}_T(\mathbf{x})$, relative humidity $\mathsf{P}_{RH}(\mathbf{x})$ and CO_2 concentration $\mathsf{P}_C(\mathbf{x})$ are defined as

$$\mathsf{P}_{\mathbf{x}}(\mathbf{x}) = \begin{cases} c_{\mathbf{x}} \cdot |\mathbf{x}^{\min} - \mathbf{x}|, & \text{if } \mathbf{x} < \mathbf{x}^{\min} \\ 0, & \text{if } \mathbf{x}^{\min} \leq \mathbf{x} \leq \mathbf{x}^{\max} \\ c_{\mathbf{x}} \cdot |\mathbf{x}^{\max} - \mathbf{x}|, & \text{if } \mathbf{x} > \mathbf{x}^{\max}, \end{cases} \tag{9}$$

where $c_{\mathbf{x}}$ is the weight factor associated with the state $\mathbf{x}$ exceeding the bounds $\mathbf{x}^{\min}$ and $\mathbf{x}^{\max}$. With the penalty function increasing linearly in value with respect to the deviation, the discontinuities in (9) are smoothed via

$$\begin{aligned} \mathsf{P}_{\mathbf{x}}(\mathbf{x}) \approx \frac{c_{\mathbf{x}}}{2} \cdot \Big(&\sqrt{(\mathbf{x}^{\min} - \mathbf{x})^2 + \gamma} + \sqrt{(\mathbf{x}^{\max} - \mathbf{x})^2 + \gamma} \\ &- (\mathbf{x}^{\max} - \mathbf{x}^{\min}) \Big). \end{aligned} \tag{10}$$

These penalty functions are used as soft constraints. Economic profit is represented by a negative cost term, which is calculated from the difference between running and terminal cost. In (8), p_{E} is the price of electricity, p_{C} is the price of industrial CO_2, p_{y} is the selling price of yield, and $k_{\mathrm{B_{fruit}}}$ is the percentage of biomass corresponding to fruit. The operating cost of the vent is obtained by multiplying the maximum power rating of the ventilation motor $P_{\mathrm{v}}^{\max}$, p_{E} and the input u_{V}. Likewise, the cost of CO_2 injection and the operating cost of heating and cooling are obtained. We restrict inputs $\mathbf{u}(t) \in \mathscr{U}$ and states $\mathbf{x}(t) \in \mathscr{X}$ in order to maintain temperature, CO_2 concentration, and humidity within the safe limit, despite being penalized as soft constraints. These considerations motivate the Optimal Control Problem (OCP)

$$
\begin{aligned}
\mathbf{u}_{\mathrm{o}}^{*} = \arg\min_{\mathbf{u}} \quad & J_{\mathrm{o}}(\mathbf{u}) && (11)\\
\text{s.t.} \quad & \mathbf{x}t_{k+1}) = \mathbf{x}(t_k) + \Delta t_{\mathrm{o}} \cdot \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k), \mathbf{d}(t_k)),\\
& \mathbf{x}(t_0) = \mathbf{x}_0,\\
& \mathbf{u}(t_k) \in \mathscr{U} \forall k \in \{0, \ldots.., N_{\mathrm{o}} - 1\},\\
& \mathbf{x}(t_k) \in \mathscr{X} \forall k \in \{0, \ldots.., N_{\mathrm{o}} - 1\},
\end{aligned}
$$

where $\mathbf{x}_0$ is the initial state and $\mathbf{u}_{\mathrm{o}}^{*}$ is the optimal reference input for the lower-level NMPC.

B. Closed-loop NMPC

The lower level NMPC is formulated to track the reference trajectories obtained from upper level. Nonlinear MPC is employed here to handle the sudden large deviations from reference, due to the disturbance and high nonlinearity. Here, the optimal input variables $\mathbf{u}_{\mathrm{o}}^{*}$ obtained from (11) are taken as input reference trajectories $\mathbf{u}^{\mathrm{ref}}$. Similarly, the corresponding state variables is taken as the reference state trajectories $\mathbf{x}^{\mathrm{ref}}$ to the closed loop controller. The NMPC problem with a prediction horizon $N \in \mathbb{N}^{+}$ is defined by

$$
\begin{aligned}
\min_{\mathbf{u}} \quad & J_{\mathrm{c}}(\mathbf{u}) = \sum_{j=i}^{i+N-1} (\|\mathbf{x}(t_j) - \mathbf{x}_k^{\mathrm{ref}}\|_Q^2 + \|\mathbf{u}(t_j) - \mathbf{u}_k^{\mathrm{ref}}\|_R^2)\\
& \quad + \|\mathbf{x}(t_N) - \mathbf{x}_k^{\mathrm{ref}}\|_P^2 && (12a)\\
\text{s.t.} \quad & \mathbf{x}(t_{j+1}) = \mathbf{x}(t_j) + \Delta t_{\mathrm{c}} \cdot \mathbf{f}(\mathbf{x}(t_j), \mathbf{u}(t_j), \mathbf{d}(t_j)), && (12b)\\
& \mathbf{x}_0(t_j) = \mathbf{x}_0, && (12c)\\
& \mathbf{u}(t_j) \in \mathscr{U} \forall j \in \{0, \ldots.., N_s - 1\}, && (12d)\\
& \mathbf{x}(t_j) \in \mathscr{X} \forall j \in \{0, \ldots.., N_s - 1\}, && (12e)
\end{aligned}
$$

where $Q \succeq 0$, $R \succ 0$ and $P \succeq 0$ are the weighting matrices and $N_{\mathrm{s}} = \frac{\mathsf{T}}{\Delta t_{\mathrm{c}}} \in \mathbb{N}^{+}$. The NMPC problem is solved with a short-term disturbance forecast of horizon N. We solve OCP (12) for the optimal input trajectories $\mathbf{u}_j^{\star}$ and apply $\mathbf{u}_0^{\star}$ to system (12b). The closed-loop NMPC feedback controller based on (12) can be expressed as the function

$$
\mathbf{u} = \pi_{\mathrm{MPC}}(\mathsf{w}), \qquad (13)
$$

where $\mathsf{w} = \begin{bmatrix} \mathbf{x} & \mathbf{x}^{\mathrm{ref}} & \mathbf{u}^{\mathrm{ref}} & \mathbf{d} \end{bmatrix}^{\top} \in \mathbb{R}^{16}$.

IV. Deep Neural Network Controller

The main idea of this paper is to approximate the reference tracking closed-loop NMPC discussed in the previous section using a DNN. Now, let us recall the structure of feed-forward neural networks, network training, and finally the implementation of closed-loop control using DNNs.

A. Deep Neural Networks

We aim to approximate the function (13) using a feed-forward Neural Network (NN) by defining the mapping $\mathsf{u} = \pi_{\mathrm{NN}}(\mathsf{w}; \Theta)$, where Θ represents the unknown NN parameters. In general, a neural network consists of one input layer, one output layer, and N_{h} hidden layers, where an NN is called DNN for $N_{\mathrm{h}} \geq 2$ [22]. Moreover, it was shown in [12] that the performance of DNN compared to a shallow network ($N_h = 1$) results in better performance for less number of parameters Θ.

Each layer l contains n_l neurons, and the hidden layers are structured as

$$
h^l = \beta(b^l + W^l h^{l-1}), \qquad (14)
$$

where $h^{l-1} \in \mathbb{R}^{n_{l-1}}$ is the output of the previous layer with $h^0 = \mathsf{w}$. Here $b^l \in \mathbb{R}^{n_l}$ is called the bias vector, $W^l \in \mathbb{R}^{n_l \times n_{l-1}}$ is called the weight matrix, and β is a nonlinear activation function. Typical activation functions are Rectifying Linear Unit (ReLU), tangent hyperbolic, and the sigmoid function. The most commonly used activation function ReLU is defined as $\beta(h) = \max(0, h)$, which can enhance the training of DNN by avoiding vanishing gradients.

The parameter $\Theta = \{\Theta_1, \ldots, \Theta_{N_h+1}\}$ contains all the weights and biases that define the operation (14) of each layer. The number of parameters to be stored defines the memory footprint of the proposed controller and is given by $N_{\Theta} = \sum_{l=1}^{N_h+1} n_l(n_{l-1} + 1)$.

With the architecture being defined, training of the neural networks involves finding the optimal weights W^l and biases b^l by minimizing the defined loss function. We consider Mean Squared Error (MSE) between the optimal input ($\mathbf{u}$) and the approximate control input (u) as the loss function.

The optimization problem solved at training is given by

$$
\min_{W^l, b^l} \quad \frac{1}{N_D} \sum_{m=1}^{N_D} (\mathbf{u}(\mathsf{w}^m) - \mathsf{u}(\mathsf{w}^m))^2,
$$

where N_D is the number of training data pairs in the training data matrix $\mathscr{D} = [\mathsf{w}^1, \mathsf{w}^2, \ldots, \mathsf{w}^{N_D}] \in \mathbb{R}^{16 \times N_D}$, and $\mathscr{D}$ should be sufficiently large enough. After the network training, we use $\pi_{\mathrm{NN}}(\mathsf{w}; \Theta)$ to infer the control input $\mathsf{u} \approx \pi_{\mathrm{MPC}}(\mathsf{w})$.

Normalizing the training data helps to improve the numerical properties of the network when inputs to NN are in different ranges. The normalization operation results in a normalized data matrix $\mathscr{D}_N$. Similarly, inverse transformation is applied during the NN inference to recover the output in the range of training data.

In addition, the hyperparameters like the number N_h of hidden layers, the number n_l of neurons within a layer, the activation function β, and the gradient-based optimization algorithm along with its parameters like learning rate affect the performance of the network. Typically, experimentation is necessary to determine an optimal combination of these hyper-parameters [23].

V. Simulation and Results

As discussed in the previous sections, we use two-level hierarchical control to first generate economic reference trajectories for the inputs and states and, then track them using a DNN. The OCP (11) is solved subject to input constraints $0 \leq \mathbf{u} \leq 1$ and soft constraints on states with $\mathbf{x}_{\min} = [18, 500, 60]$ and $\mathbf{x}_{\max} = [26, 900, 90]$. The constraint on the state C is generally mentioned in ppm. Additionally,

$$
\mathscr{X} = [14, 30] \times [300, 1000] \times [10, 100] \times [0, 100]
$$

defines a hard box constraint set. Now, the upper-level OCP is solved with sampling time $\Delta t_\mathrm{o} = 300\,\mathrm{s}$ for a single day $\mathrm{T} = 86400\,\mathrm{s}$, resulting in $N_\mathrm{o} = 288$. On the other hand, the NMPC is solved for every 300 s with $\Delta t_\mathrm{c} = 60\,\mathrm{s}$ and prediction horizon $N = 5$. The weighing matrices for the closed loop are chosen as $Q = 100 \cdot \mathrm{I}_4$, $R = \mathrm{I}_4$, and $P = Q$.

A. Training of the reference tracking DNN controller

For achieving a sufficiently large data matrix $\mathscr{D}$, 8 months of disturbance data during which the greenhouse operation was active is considered. This results in 285,000 data pairs with 80% used for training and the remainder for testing network accuracy. `Adam` [24], a stochastic gradient descent optimization method is used to perform training via `Tensorflow` [25]. The last timestamp values of the previous day are taken as the initial condition for the next-day optimization and reference tracking problem.

To solve the optimization problems (11) and (12), we use `CasADi` [26] with `IPOPT` [27]. Moreover, the nonlinear dynamics are discretized using orthogonal collocation for predictive control implementation and the integration operations are performed using the `SUNDIALS` toolbox. In order to provide a good approximation of π_MPC while keeping the number of parameters N_Θ small [23] a random search using `KerasTuner` is employed to determine suitable hyperparameters like N_h, n_l and β. The resulting DNN includes 5 hidden layers with almost 4,000 parameters, is trained with a batch size of 512 data pairs, a learning rate of 0.001, and occupies less than 14 kB of memory.

B. Performance of the DNN controller

The results of the simulation comparing the tracking performance of NMPC and DNN with respect to the reference trajectory generated by the upper-level OCP for one day are shown in Figure 1. The initial state vector considered consist of the last real-time state values of the previous day. The NMPC optimization as well as DNN inference are performed on a Personal Computer (PC).

First, notice that in both scenarios, the states cross the soft constraints but obey the hard constraints defined by the set $\mathscr{X}$. The CO_2 concentration is increased during the onset of solar radiation and maintained at this level. The biomass production starts once the temperature reaches around 19°C. It can be observed that both the NMPC and DNN are able to track the reference trajectories. Figure 2 shows a comparison of the NMPC and DNN controllers under 10% uncertainty of $\mathbf{d}$ with respect to the forecast data. It can be seen that the performance of the DNN is almost identical to the NMPC controller. However, the biomass production by the approximate DNN controller - $1.2497\,\mathrm{kg\,m^{-2}}$ is slightly higher than the NMPC - $1.2512\,\mathrm{kg\,m^{-2}}$.

The performance of the controllers are examined using the closed-loop net cost $S = V(\mathbf{x}(N_\mathrm{s})) - \sum_i L(\mathbf{x}(i), \mathbf{u}(i))$ of the greenhouse operation on a day for varying uncertainty in $\mathbf{d}$. Table I comprises the total costs, operating costs L, and terminal cost/income from the crop yield V, of the greenhouse operation on a day. With perfect knowledge of the disturbance $\mathbf{d}$, total net costs using NMPC are less than the ones with DNN. Alternatively, in the case of uncertainty in $\mathbf{d}$, the DNN outperforms the NMPC. The higher operation

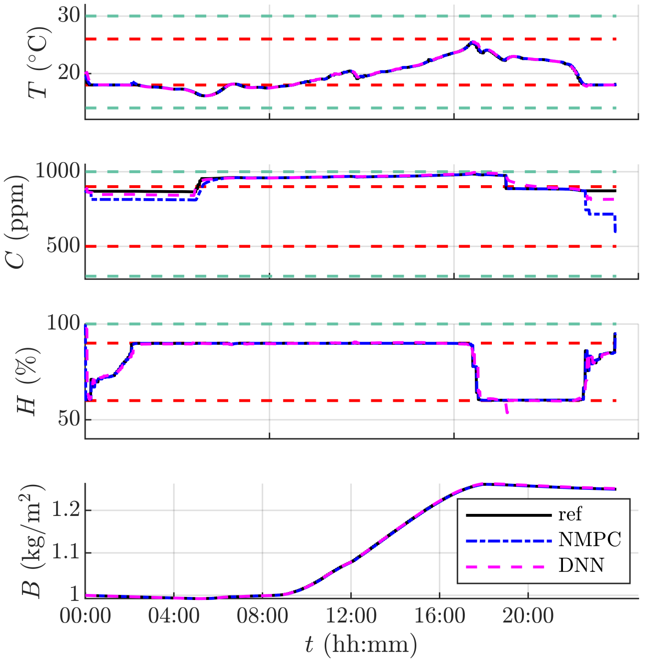

Fig. 1: Tracking performance of NMPC vs. DNN. The red and green dashed lines indicate the soft and hard constraints, respectively. The hard constraint lower bound of H = 10% is not shown here.

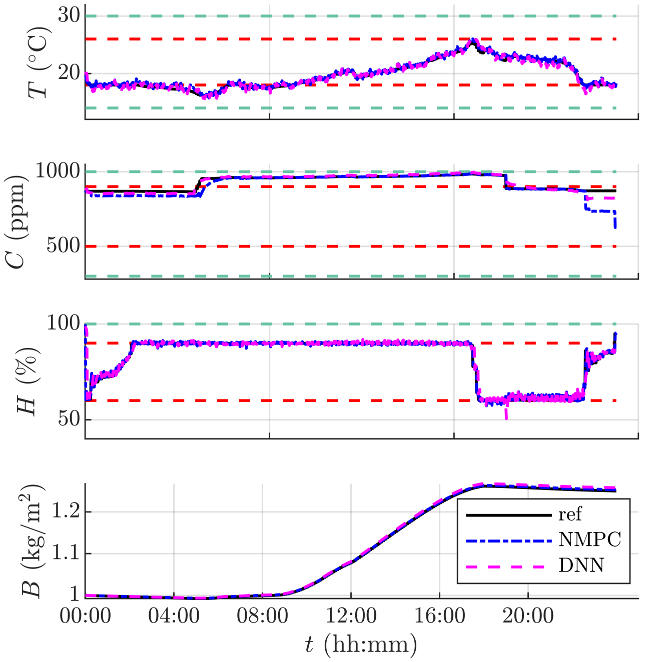

Fig. 2: Comparison of the convergence of states in NMPC vs. DNN under 10% uncertainty of $\mathbf{d}$ with respect to forecast data. The red and green dashed lines indicate the soft and hard constraints of the states respectively. The hard constraint lower bound of RH - 10% is not shown here.

TABLE I: Comparison of net cost of greenhouse operation using NMPC vs. DNN under uncertain disturbance

Uncertainty [%]	NMPC			DNN		
	S[€]	L[€]	V[€]	S[€]	L[€]	V[€]
1%	113.33	12.85	126.18	100.24	12.78	113.02
10%	68.76	30.36	99.13	103.08	13.91	117.01
20%	76.44	32.65	109.09	101.63	16.44	118.08
30%	70.25	38.52	108.77	99.21	19.04	118.25

TABLE II: Comparison of computation time on a PC

Uncertainty[%]	NMPC		DNN	
	Mean [ms]	SD [ms]	Mean [ms]	SD [ms]
1%	155.7561	74.9314	0.0208	0.0048
10%	162.4664	87.5212	0.0203	0.0035
20%	159.4011	85.8751	0.0204	0.0028
30%	158.1797	81.8533	0.0227	0.0082

costs of NMPC indicates the increased control effort to handle $\mathbf{d}$ in the pursuit of reference tracking. Note that the approximate nature of DNN controller and its training with varied disturbance over 8-month training data results in a better performance compared to NMPC.

The computation effort of both controllers under different uncertainty levels of $\mathbf{d}$ is shown in Table II. Compared to NMPC, the average execution time of a DNN controller is faster by four orders of magnitude for a PC execution.

VI. Conclusion

This paper presents the implementation of a hierarchical model predictive control of a greenhouse with a lower level of fast control using deep neural networks. Our simulations show a faster computation time and better performance in the presence of uncertain forecast data using a DNN controller compared to NMPC. The low memory requirement of an approximate DNN controller makes it more desirable for constrained embedded hardware implementation, which results in less power consumption. Future work will consider the implementation of the controller using low-cost embedded devices on an experimental setup.

References

[1] T. Searchinger, R. Waite, C. Hanson, J. Ranganathan, P. Dumas, E. Matthews, and C. Klirs, "Creating a sustainable food future: A menu of solutions to feed nearly 10 billion people by 2050. final report," 2019.

[2] D. Liu, X. Liu, K. Guo, Q. Ji, and Y. Chang, "Spillover effects among electricity prices, traditional energy prices and carbon market under climate risk," *Int. J. Environ. Res. Public Health*, vol. 20, no. 2, 2023.

[3] R. Hassanien, M. Li, and W. Dong Lin, "Advanced applications of solar energy in agricultural greenhouses," *Renewable Sustainable Energy Rev.*, vol. 54, pp. 989–1001, 2016.

[4] E. Iddio, L. Wang, Y. Thomas, G. McMorrow, and A. Denzer, "Energy efficient operation and modeling for greenhouses: A literature review," *Renewable and Sustainable Energy Rev.*, vol. 117, p. 109480, 2020.

[5] Z. Ren, Y. Dong, D. Lin, L. Zhang, Y. Fan, and X. Xia, "Managing energy-water-carbon-food nexus for cleaner agricultural greenhouse production: A control system approach," *Sci. Total Environ.*, vol. 848, p. 157756, 2022.

[6] H. Hu, L. Xu, R. Wei, and B. Zhu, "Multi-objective control optimization for greenhouse environment using evolutionary algorithms," *Sensors*, vol. 11, no. 6, pp. 5792–5807, 2011.

[7] Y. Su, L. Xu, and D. Li, "Adaptive fuzzy control of a class of mimo nonlinear system with actuator saturation for greenhouse climate control problem," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, p. 772 – 788, 2016.

[8] P. Van Beveren, J. Bontsema, G. Van Straten, and E. Van Henten, "Minimal heating and cooling in a modern rose greenhouse," *Appl. Energy*, vol. 137, pp. 97–109, 2015.

[9] 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.

[10] R. van Ooteghem, "Optimal control design for a solar greenhouse," Ph.D. dissertation, 2007, wageningen University, Wageningen, The Netherlands. 304 p.

[11] M. Padmanabha, L. Beckenbach, and S. Streif, "Model predictive control of a food production unit: A case study for lettuce production," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15 771–15 776, 2020, 21st IFAC World Congress.

[12] S. Lucia and B. Karg, "A deep learning-based approach to robust nonlinear model predictive control," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 511–516, 2018, 6th IFAC Conference on Nonlinear Model Predictive Control, 2018.

[13] T. Parisini and R. Zoppoli, "A receding-horizon regulator for nonlinear systems and a neural approximation," *Automatica*, vol. 31, no. 10, pp. 1443–1451, 1995.

[14] S. Pon Kumar, A. Tulsyan, B. Gopaluni, and P. Loewen, "A deep learning architecture for predictive control," *IFAC-PapersOnLine*, vol. 51, no. 18, pp. 512–517, 2018, 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018.

[15] M. Hertneck, J. Köhler, F. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 543–548, 2018.

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

[17] J. Bontsema, J. Hemming, C. Stanghellini, P. De Visser, E. Van Henten, J. Budding, T. Rieswijk, and S. Nieboer, "On-line estimation of the transpiration in greenhouse horticulture," in *Proceedings Agricontrol 2007. 2nd IFAC International Conference on Modeling and Design of Control Systems in Agriculture, Osijek, Croatia, 3-5 September, 2007*, 2008, pp. 29–34.

[18] C. Stanghellini and T. de Jong, "A model of humidity and its applications in a greenhouse," *Agric. For. Meteorol.*, vol. 76, no. 2, pp. 129–148, 1995.

[19] B. Vanthoor, *A model-based greenhouse design method.* Wageningen University and Research, 2011.

[20] D. Dannehl, H. Kläring, and U. Schmidt, "Light-mediated reduction in photosynthesis in closed greenhouses can be compensated for by CO_2 enrichment in tomato production," *Plants*, vol. 10, no. 12, 2021.

[21] K. Biswas, S. Kumar, S. Banerjee, and A. Pandey, "SMU: smooth activation function for deep networks using smoothing maximum technique," *arXiv e-prints*, p. arXiv:2111.04682, Nov. 2021.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016.

[23] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *J. Mach. Learn. Res.*, vol. 13, no. 2, 2012.

[24] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[25] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[26] J. Andersson, J. Gillis, G. Horn, J. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, 2019.

[27] A. Wächter and L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.