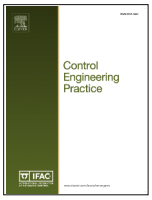
Control Engineering Practice 169 (2026) 106787

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac



Stochastic model predictive control with reinforcement learning for greenhouse production systems under parametric uncertainty

Bart van Laatum [a,*], Salim Msaad [b], Eldert J. van Henten [a], Robert D. Mcallister [b], Sjoerd Boersma [a,c]

[a] Agricultural Biosystems Engineering, Wageningen University & Research, Droevendaalsesteeg 4, Wageningen, 6708 PB, The Netherlands
[b] Delft Center for Systems and Control (DCSC), Delft University of Technology, Mekelweg 5, Delft, 2628 CD, The Netherlands
[c] Biometris, Wageningen Research, Droevendaalsesteeg 4, Wageningen, 6708 PB, The Netherlands

ARTICLE INFO

Keywords:
Stochastic model predictive control
Reinforcement learning
Terminal costs
Terminal region constraints
Feedback policies
Greenhouse production control

ABSTRACT

Uncertainty, if not explicitly accounted for in controller design, can significantly degrade the optimal control performance of greenhouse production systems. Scenario-based stochastic MPC (SMPC) addresses uncertainty by approximating its underlying probability distributions through sampling. However, SMPC rapidly becomes computationally intractable and can suffer from growing uncertainty with longer prediction horizons. Terminal costs and constraints ensure closed-loop performance of SMPC, but designing these for greenhouse systems is challenging since they rely on steady-state targets that often do not exist in greenhouse production systems. To overcome these challenges, this work introduces RL-SMPC, which uses reinforcement learning (RL) to learn a control policy that constructs both terminal region constraints and a terminal cost function. Additionally, this policy serves as a nonlinear feedback policy to attenuate uncertainty growth in the open-loop solution of scenario-based SMPC. RL-SMPC's closed-loop performance is compared against standalone RL, MPC, and scenario-based SMPC on a greenhouse lettuce model under parametric uncertainty. Simulation results showed that RL-SMPC outperformed MPC across all prediction horizons and surpassed SMPC for horizons shorter than five hours. Moreover, the results indicated that at equal online computational cost, RL-SMPC outperformed SMPC.

1. Introduction

Greenhouse crop production systems can produce fresh food economically and reliably year-round, regardless of seasonal weather variability. This is achieved by maintaining an ideal growing climate for greenhouse crops while minimizing resource consumption such as gas, electricity, and CO_2. This climate is regulated through actuators such as CO_2 injection, ventilation, and heating. Currently, growers define climate strategies, i.e., steady-state targets, that climate computers aim to follow through using traditional bang-bang or PID controllers (Chaudhary et al., 2019; Hamza et al., 2019; Lafont et al., 2015). However, these control methods do not consider optimal crop yield or resource efficiency, as they target steady-state setpoints rather than economic objectives, nor can they anticipate future disturbances. Moreover, applying PID control in multiple-input multiple-output (MIMO) greenhouse systems with complex dynamics and tightly coupled variables such as temperature and humidity often requires extensive controller tuning. To overcome these limitations, control methods are needed that enforce hard constraints while optimizing long-term objectives. Optimal control and reinforcement learning (RL) are advanced control methods that can balance long-term performance and real-time system constraints by optimizing an economic objective.

Over the past decades, optimal control methods have been applied to support the operation of greenhouse production systems (Blasco et al., 2007; Ding et al., 2018; Gruber et al., 2011; Ito, 2012; Kuijpers et al., 2021; Montoya et al., 2016; Van Henten, 1994). Model predictive control (MPC) is a specific optimal control implementation that optimizes an objective function over a finite horizon using a prediction model. Given the current state of the system, MPC computes the solution to the finite-horizon optimization problem, applies only the first optimal input, and repeats this procedure at the subsequent time step, i.e., a receding horizon method. The MPC framework addresses MIMO systems and can incorporate specific constraints on the state and input of the system. Since optimization occurs online, MPC naturally handles future disturbances like outdoor weather and market price fluctuations. However, greenhouse prediction models do not

* Corresponding author.
E-mail addresses: bart.vanlaatum@wur.nl (B. van Laatum), s.msaad@tudelft.nl (S. Msaad), eldert.vanhenten@wur.nl (E.J. van Henten), R.D.McAllister@tudelft.nl (R.D. Mcallister), sjoerd.boersma@wur.nl (S. Boersma).

https://doi.org/10.1016/j.conengprac.2026.106787
Received 7 August 2025; Received in revised form 9 January 2026; Accepted 9 January 2026
Available online 21 January 2026

perfectly mimic real-world greenhouse systems due to modeling errors related to complex crop physiology and system variability. These errors can be represented as parametric model uncertainties, which can significantly affect control performance when not accounted for (Boersma et al., 2022; Mondaca-Duarte et al., 2020; Van Henten, 2003). The majority of greenhouse optimal control or MPC applications assumed deterministic models (Blasco et al., 2007; Ding et al., 2018; Kuijpers et al., 2021; Van Henten, 1994). Studies focusing on uncertainty in greenhouse systems have mainly addressed external disturbances such as weather or price forecasts rather than parametric uncertainties (Chen & You, 2020; García-Mañas et al., 2024; Kuijpers et al., 2022). Therefore, there is a need for control methods that adapt inputs to mitigate the impact of parametric uncertainties.

Most literature on greenhouse control treats parametric uncertainty via worst-case bounds by using a linearized model for computational reasons. González et al. (2014), Hamza et al. (2019), Piñón et al. (2001) studied robust MPC by linearizing the greenhouse model and propagating uncertainty through bounded constraints. While effective, these worst-case methods could result in more conservative controllers. Svensen et al. (2024) addressed this conservatism by explicitly handling parametric uncertainty through stochastic MPC (SMPC) with chance constraints. However, all of these approaches rely on linearization of the greenhouse model, which may increase uncertainty when the system deviates from the linearization point. In contrast, Boersma et al. (2022) applied robust scenario-based MPC to a nonlinear greenhouse model, eliminating the need for model linearization. But this robust MPC framework yielded a conservative controller, as evidenced by the decreased crop growth and increased CO_2 and heating demand. Another line of work focuses on parameter estimation and learning in greenhouse control problems. Xu et al. (2018) proposed online parameter estimation for time-varying parameters of the greenhouse model, yielding an adaptive MPC formulation. More recently, Mallick et al. (2025) integrated RL and MPC by adapting a parameterized MPC controller as an RL policy and cost function approximator. Specifically, this work uses RL to learn the parameterization of the MPC framework. Although this approach reduced constraint violations, it does not explicitly propagate uncertainty distributions throughout the prediction horizon.

Scenario-based SMPC offers a promising direction for explicitly propagating uncertainty in nonlinear optimal control problems (Mesbah, 2016). By sampling a finite set of uncertainty realizations to approximate the uncertainty distribution, this approach enables optimization of uncertain processes, such as crop growth, under probabilistic system constraints. However, scenario-based SMPC can become computationally intractable as the number of samples and the prediction horizon grow. Reducing the number of scenarios compromises coverage of the uncertainty space, while shortening the horizon risks neglecting long-term climate-crop interaction, leading to suboptimal performance. Existing approaches aim to ensure closed-loop performance with short horizons by incorporating terminal costs and constraints in the SMPC formulation. For linear systems, conventional strategies handle additive or multiplicative disturbances using stochastic Lyapunov functions and/or tube-based methods as terminal costs and constraint functions to guarantee stability and recursive feasibility (Cannon et al., 2011; Primbs & Sung, 2009). The construction of terminal costs and constraints for performance guarantees is addressed in (Lorenzen et al., 2017), where a tradeoff is shown between feasible region size and average performance. In Chatterjee and Lygeros (2015), stochastic Lyapunov functions were also used as terminal costs in a general nonlinear SMPC framework to establish stability and performance guarantees. However, these approaches typically require defining an appropriate steady state for the system. This requirement limits their use in greenhouse production control, where a steady state is either unavailable or undesirable. Though generalized terminal constraint methods avoid defining steady-state targets a priori, they still enforce that the terminal state-input pair satisfies a feasible steady state condition during online optimization (Fagiano & Teel, 2013; MÃ¼ller et al., 2014). To the authors' knowledge, the design of terminal constraints, terminal costs, and feedback policies for scenario-based SMPC with economic objectives remains largely unexplored.

Moreover, uncertainty naturally grows with longer prediction horizons, causing the planned trajectory to be overly conservative by keeping an unrealistically high distance from the defined system constraints. Incorporating a feedback control policy for each sampled realization can attenuate this growth. A common approach is to approximate this with a linear feedback policy (Mayne et al., 2011; Messerer & Diehl, 2021). However, deriving such a linear policy for a nonlinear economic optimal control problem is nontrivial for similar reasons as the construction of terminal costs and constraints. The reliance of linear feedback policies on steady-state targets conflicts with the objectives of greenhouse production systems, where such steady states are undesirable. Consequently, there is a need for a control policy that can simultaneously provide terminal costs and constraints, as well as feedback laws within a tractable SMPC framework.

As an alternative to (S)MPC, RL is a framework that learns (near-)optimal control policies by optimizing an objective through numerous system interactions (Sutton & Barto, 2018). Policies, usually parameterized by deep neural networks (DNNs), map observations to control inputs. Since training predominantly occurs offline using a model of the system, RL has relatively low online computational time. The RL objective function is defined to maximize the expected cumulative reward, enabling the learned policy to account for the long-term performance. Moreover, RL can naturally incorporate uncertainty by injecting sampled uncertainty into the system dynamics during training, eliminating the need for online optimization and reducing computational overhead. Recent work has evaluated RL for greenhouse production control, comparing MPC with the deep deterministic policy gradient algorithm (Morcego et al., 2023), and developing stochastic simulation environments for RL-based greenhouse control (Van Laatum et al., 2025). Despite these advancements, RL struggles to enforce system constraints, which are usually formulated as a penalty in the objective function (Brunke et al., 2022). This method requires careful tuning to balance the objective and the penalty function, often yielding overly conservative policies or constraint violations. In greenhouse production systems, this means that growers cannot be certain the controller will maintain climate variables within safe ranges, with potential negative implications on crop growth and health.

The properties of RL and SMPC offer complementary strengths: RL can achieve long-term performance under stochastic dynamics with low online computation time, while SMPC can consider constraints more precisely but becomes computationally demanding over long prediction horizons. This synergy motivates exploring integrated RL-SMPC methods. For RL and *nominal* MPC, there are a variety of recently proposed combinations that demonstrate the potential for performance improvements achieved by the synergy between these methods (see the review in Reiter et al. (2025) for more details). However, there are fewer methods available to combine RL and SMPC. Two such methods were proposed in Chen et al. (2020), which used SMPC to track a trajectory generated by RL, and Zarrouki et al. (2024), which learn relevant parameters in the SMPC optimization problem (such as horizon length and constraint tightening) through closed-loop simulations.

Of these various methods to combine RL and MPC, this work focuses specifically on the capability of RL to provide terminal ingredients for (S)MPC. Msaad et al. (2025) and Reiter et al. (2025) demonstrated that RL can improve over *nominal* MPC solutions in a deterministic setting at short prediction horizons by providing warm starts, terminal constraints, and a terminal cost. This study proposes extending these methods to combine RL and SMPC, with the aim of achieving similar improvements in a stochastic environment and reducing the computational demand of scenario-based SMPC methods.

To this end, *RL-SMPC* is introduced, integrating a trained RL policy into the scenario-based SMPC framework. RL-SMPC extends the RL-guided MPC algorithm (Msaad et al., 2025) into a scenario-based SMPC

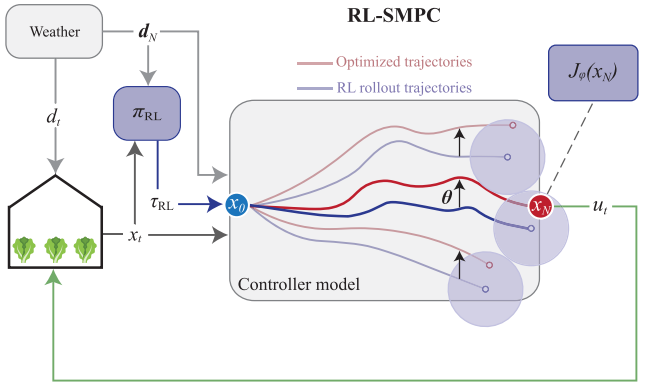


Fig. 1. Illustrative sketch of RL-SMPC during online optimization. At time step t, the greenhouse state $x(t)$ is measured. Starting from this state, the trained RL policy π_{RL} generates a set of rollout trajectories τ_{RL} over the prediction horizon N. The blue trajectories delineate these trajectories in the RL-SMPC controller. The shaded areas represent the terminal region constraints provided by the trajectories. RL-SMPC then optimizes the decision variable θ, a uniform shift applied to all rollout trajectories, resulting in the predicted RL-SMPC trajectories, visualized in red. The terminal-cost function J_ϕ evaluates each terminal state. Finally, the first control action $u(t)$ is applied to the system. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

framework for greenhouse production control under parametric uncertainty. At each control iteration, the policy provides a terminal region constraint and an approximation of the terminal cost, without the need for steady-state targets. Additionally, this RL policy serves as a nonlinear feedback law, circumventing the design of an ad hoc linear feedback law that is typically required for SMPC. RL-SMPC modifies the standard scenario-based SMPC framework through three key components:

1. A trained RL policy generates rollout trajectories to define terminal region constraints for each sampled uncertainty scenario.
2. A DNN-based terminal cost function approximates state values at the end of the prediction horizon, learned from closed-loop trajectories of the RL policy.
3. The RL policy serves as a nonlinear feedback policy.

By providing a terminal region constraint and cost function across all sampled scenarios, these RL components can transfer long-term information to the SMPC framework, enabling better performance at shorter prediction horizons. Furthermore, employing the RL policy as a nonlinear feedback law attenuates the growth of uncertainty over extended horizons, without relying on linear feedback approximations as most SMPC methods do (Mayne et al., 2011; Messerer & Diehl, 2021). This work compares RL-SMPC against standalone RL, MPC, and scenario-based SMPC on a lettuce greenhouse model under parametric uncertainty, addressing the research question: *To what extent does RL-SMPC improve greenhouse production control, compared to SMPC and MPC, under parametric uncertainty?* Fig. 1 illustrates the RL-SMPC algorithm during online deployment.

To the best of current knowledge, this work is the first to integrate RL with SMPC using terminal region constraints, terminal costs, and nonlinear feedback policies. In simulation experiments, the RL-SMPC framework was compared against standalone MPC, SMPC, and RL for a greenhouse production control problem under parametric model uncertainty. The results showed that RL-SMPC outperformed nominal MPC across all eight tested prediction horizons ($1-8$ hours), and varying uncertainty levels ($5\%-20\%$). Moreover, RL-SMPC surpassed SMPC in performance for prediction horizons up to four hours, while maintaining comparable computational costs. Ablation studies indicated that the terminal region constraints primarily drive the performance gains.

The remainder of this paper is organized as follows: Section 2 introduces the greenhouse production model and control problem. Section 3 details the RL-SMPC methodology and implementation. Section 4 presents results, which are discussed in Section 5. Finally, Section 6 concludes the paper and outlines future research directions.

2. Problem formalization

2.1. Lettuce greenhouse model

The nonlinear dynamical greenhouse lettuce model in this work is taken from (Van Henten, 1994). The discrete-time form is obtained via numerical integration using the fourth-order Runge-Kutta method:

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

with discrete time step $t \in \mathbb{Z}^{0+}$, state variable $x(t) \in \mathbb{R}^4$, measurement $y(t) \in \mathbb{R}^4$, input $u(t) \in \mathbb{R}^3$, weather disturbance $d(t) \in \mathbb{R}^4$, and $p \in \mathbb{R}^{22}$ represents the model parameters. The nonlinear functions $f(\cdot)$ and $h(\cdot)$ represent the greenhouse model dynamics and measurement function, respectively. The state, output, control input, and disturbance are defined as:

$$\begin{aligned} x(t) &= \left(x_{\text{DW}}(t),\ x_{\text{CO}_2}(t),\ x_{\text{T}}(t),\ x_{\text{H}_2\text{O}}(t)\right)^\top, \\ y(t) &= \left(y_{\text{DW}}(t),\ y_{\text{CO}_2}(t),\ y_{\text{T}}(t),\ y_{\text{RH}}(t)\right)^\top, \\ u(t) &= \left(u_{\text{CO}_2}(t),\ u_{\text{vent}}(t),\ u_{\text{heat}}(t)\right)^\top, \\ d(t) &= \left(d_{\text{Iglob}}(t),\ d_{\text{CO}_2}(t),\ d_{\text{T}}(t),\ d_{\text{H}_2\text{O}}(t)\right)^\top. \end{aligned} \tag{2}$$

The model parameters p are assumed to be uncertain and distributed uniformly as follows:

$$p \sim \mathcal{U}(\Theta), \tag{3}$$

$$\Theta := \left\{ p \in \mathbb{R}^{22} \;\middle|\; p_i \in \left[\bar{p} - \frac{\delta}{2}, \bar{p} + \frac{\delta}{2}\right] \right\} \tag{4}$$

where $\bar{p}$ represents the nominal parameter values, and p_i represents the i-th element of p. The support Θ is thus defined by the nominal parameter values and the uncertainty size δ. The distribution $\mathcal{U}(\Theta)$ denotes the uniform distribution over the hyper-rectangle Θ.
A complete description of the model, together with the nominal parameter values $\bar{p}$ is provided in Appendix A.

2.2. Greenhouse production control problem

Greenhouse production control aims to maximize crop yield by regulating indoor climate: temperature, humidity, and CO_2, using actuators such as heating, ventilation, and CO_2 injection. The goal is high yield with minimal energy use. Although harvest timing affects profitability, this study imposes a fixed 40-day cultivation period T, consistent with prior work (Boersma et al., 2022; Mallick et al., 2025; Morcego et al., 2023; Msaad et al., 2025).

The Economic Profit Indicator (EPI) serves as the objective function to be maximized. It is defined as the revenue from selling the accumulated crop weight minus the costs incurred during the cultivation period:

$$\begin{aligned} EPI = c_{\text{DW}} &\left(x_{\text{DW}}(T) - x_{\text{DW}}(0)\right) \\ &- \sum_{t=0}^{T-1} \left(c_{\text{CO}_2} u_{\text{CO}_2}(t) + c_{\text{heat}} u_{\text{heat}}(t)\right) \Delta t, \end{aligned} \tag{5}$$

where c_{DW} is the unit selling price of the crop in dry weight, $x_{\text{DW}}(T)$ is the yield at harvest time T, c_{CO_2} and c_{heat} are the unit costs of CO_2 injection and heating, respectively, and Δt is the time step size. Direct incorporation of the economic objective in (5) within the MPC or RL framework requires mapping the EPI to an economic stage cost, where the benefit for plant growth is evaluated at each time step rather than solely at the end of the growing period. For this reason, the following

economic stage cost is defined and evaluated at each time step:

$$\ell_{\mathrm{e}}\big(u(t), x(t), x(t+1)\big) = - c_{\mathrm{DW}}\big(x_{\mathrm{DW}}(t+1) - x_{\mathrm{DW}}(t)\big) + \big(c_{\mathrm{CO_2}} u_{\mathrm{CO_2}}(t) + c_{\mathrm{heat}} u_{\mathrm{heat}}(t)\big)\Delta t. \tag{6}$$

By construction, the cumulative sum of the stage cost over the cultivation horizon corresponds to the EPI in (5), up to a sign reversal. The employed crop prediction model is valid only within specified climate bounds $y^{\min}$ and $y^{\max}$, as operating outside this domain can induce crop diseases. These bounds are not enforced as hard constraints in the optimization problem, since uncertainties in model parameters may cause violations or infeasibility. Similarly, reinforcement learning cannot guarantee strict satisfaction of state constraints. In order to have a fair comparison between methods, state constraint violations are penalized via linear penalty functions:

$$\ell_{\mathrm{p}}\big(x(t)\big) = g_{\mathrm{CO_2}}\big(y_{\mathrm{CO_2}}(t)\big) + g_{\mathrm{T}}\big(y_{\mathrm{T}}(t)\big) + g_{\mathrm{RH}}\big(y_{\mathrm{RH}}(t)\big), \tag{7}$$

where the outputs $y_{\mathrm{CO_2}}(t)$, $y_{\mathrm{T}}(t)$ and $y_{\mathrm{RH}}(t)$ are obtained from the measurement function as $y(t) = h(x(t))$. The functions $g_{\mathrm{CO_2}}$, g_{T}, and g_{RH} apply penalties for deviations from their respective acceptable ranges. Each penalty function $g_{(\cdot)}$ is defined as:

$$g_{(\cdot)}\big(y_{(\cdot)}(t)\big) = \begin{cases} \lambda_{(\cdot)}\big(y_{(\cdot)}(t) - y_{(\cdot)}^{\max}\big) & \text{if } y_{(\cdot)}(t) > y_{(\cdot)}^{\max}, \\ \lambda_{(\cdot)}\big(y_{(\cdot)}^{\min} - y_{(\cdot)}(t)\big) & \text{if } y_{(\cdot)}(t) < y_{(\cdot)}^{\min}, \\ 0 & \text{otherwise}, \end{cases} \tag{8}$$

where the values of $\lambda_{(\cdot)}$ represent the penalty coefficients that regulate the scale of each penalty term. These coefficients influence the solutions obtained by the controllers. Choosing the coefficients to scale the penalty terms of comparable magnitude to the economic stage cost (6) yielded controllers with balanced economic performance and constraint satisfaction. The effect of the penalty coefficients on controller performance is discussed in Section 5.1. Combining the economic stage cost (6) and the penalty function (7) results in the following stage cost function:

$$\ell\big(u(t), x(t), x(t+1)\big) = \ell_{\mathrm{e}}\big(u(t), x(t), x(t+1)\big) + \ell_{\mathrm{p}}(x(t)). \tag{9}$$

In addition to limitations on the system states, the control inputs are also constrained. Specifically, the CO_2 injection $u_{\mathrm{CO_2}}$, ventilation rate u_{vent}, and heating input u_{heat} are restricted to lie within predefined bounds. To ensure smooth actuation, rate constraints are also imposed on the control inputs. For each control input, the maximum allowed change per time step is limited to 10% of its respective upper bound. Taking into account the objective function in (9), along with the dynamic model and the input and rate constraints, the greenhouse production control problem is formulated as:

$$\min_{\pi_0,\dots,\pi_{T-1}} \quad \mathbb{E}_{\mathcal{U}(\Theta)}\left[\sum_{t=0}^{T-1} \ell\big(u(t), x(t), x(t+1)\big)\right] \tag{10a}$$

$$\text{s.t.} \quad x(t+1) = f\big(x(t), u(t), d(t), p\big) \tag{10b}$$

$$u(t) = \pi_t(x(t)) \tag{10c}$$

$$u_{\min} \le u(t) \le u_{\max} \tag{10d}$$

$$|u(t) - u(t-1)| \le \delta u_{\max} \tag{10e}$$

$$x(0) = x_0,\ u(0) = u_0 \tag{10f}$$

where constraints (10b–10e) need to be enforced for each $t \in \{0, \dots, T-1\}$ and for each $p \in \Theta$. It is important to notice how this formulation involves solving for an optimal trajectory of feedback policies $\{\pi_0, \dots, \pi_{T-1}\}$ in which $u(t) = \pi_k(x(t))$, rather than an optimal trajectory of inputs. Optimizing feedback policies is preferred under uncertainty, as it makes it possible to restrain the spread of the trajectories resulting from uncertainty. However, optimizing over feedback policies is a difficult task, especially if nonlinearities or constraints characterize the optimal control problem. This challenge is faced in Sections 3.3 and 3.4, where feedback parametrization is used in the proposed RL-SMPC formulation to find an approximate solution.

This greenhouse production control problem in (10) serves as the common control formulation approximated by the RL agent, MPC, SMPC, and RL-SMPC. The resulting controllers are evaluated by closed-loop performance metrics, which are introduced in Section 3.5

3. Methodology

3.1. Reinforcement learning

This work leverages the Soft Actor-Critic (SAC) algorithm (Haarnoja et al., 2018), an off-policy actor-critic method known for its robustness and sample efficiency. SAC augments the standard reward objective with an entropy maximization term, promoting a trade-off between exploiting known strategies and exploring new ones. The entropy term promotes exploration by encouraging varied action selection, which helps the agent discover diverse strategies. It also mitigates premature convergence by preventing the policy from committing too early to suboptimal behaviors.

The observation used in the RL policy includes the current state $x(t)$, previous input $u(t-1)$, time t, and a sequence of future weather disturbances

$$\mathbf{d}_N(t) = (d(t), \dots, d(t+N-1)), \tag{11}$$

where N corresponds to a six-hour horizon. The RL policy π_{RL} maps this observation space to a control input:

$$u(t) = \pi_{\mathrm{RL}}(x(t), u(t-1), \mathbf{d}_N(t), t) \tag{12}$$

The policy π_{RL} is implemented as a neural network that maps the current observation to an intermediate action $a(t)$, followed by a transformation layer that converts $a(t)$ into the control input $u(t)$. This transformation layer is necessary to ensure that the final control input respects the system's constraints. To account for the control input bounds and rate constraints in (10d) and (10e), the previous input $u(t-1)$ is included in the observation space, and the action $a(t)$ is interpreted as a scaled adjustment to $u(t-1)$. This adjustment from action to control input is defined as:

$$u(t) = \max\big(u_{\min}, \min\big(u_{\max}, u(t-1) + a(t)\,\delta u_{\max}\big)\big). \tag{13}$$

Moreover, each one of the three components of $a(t)$ is constrained to lie within the interval $[-1, 1]$ by the activation functions in the neural network.

The choice of the observation space is critical for the agent's performance. In practice, greenhouse operators do not directly observe the dry weight of the crop. Nevertheless, this study assumes that the dry mass, $y_{\mathrm{DW}}(t)$, is available to the agent, the MPC, the SMPC, and the RL-SMPC to facilitate performance benchmarking. The observation space also includes sensor-based measurements commonly available in real greenhouses: CO_2 level $y_{\mathrm{CO_2}}(t)$, indoor temperature $y_{\mathrm{T}}(t)$, and relative humidity $y_{\mathrm{RH}}(t)$. These outputs form the measurement vector $y(t)$, which is derived from the state of the system via the mapping $y(t) = h(x(t))$. The observation space also includes external weather conditions. Specifically, instead of using only the current condition $d(t)$, the agent receives an entire sequence of values $\mathbf{d}_N(t)$. This ensures consistency across all control strategies, as all are provided with similar information. Lastly, the agent receives the current time step as part of its observation. This temporal information allows the policy to adjust its behavior according to the specific moment in the growing cycle.

A key hyperparameter in this setup is the discount factor $\gamma \in [0, 1]$. Setting a discount factor $\gamma = 1$ would allow full consideration of long-term effects over the entire growing period, which is ideal for optimizing economic returns. However, $\gamma = 1$ leads to unstable learning dynamics in this problem formulation. To balance foresight and training stability, a slightly reduced value of $\gamma = 0.95$ was used, which yields more reliable and effective policy learning. This discount factor was adopted

Table 1
Hyperparameters for SAC.

Hyperparameter	Value
Total time steps	192,000
Learn starts	17,460
Learning rate (α)	5×10^{-3}
Batch size	1024
Discount factor (γ)	0.95
Activation fn	ReLU
Buffer size	10^5
Polyak-coefficient	0.005
Train freq	1
Grad steps	1

from closely related work on RL policy-guided MPC by Msaad et al. (2025). It was observed that larger values resulted in less stable training without improved performance. A discount factor of $\gamma = 0.95$ provided a sufficiently long horizon for the RL agent to capture the relevant climate-crop dynamics required for efficient greenhouse lettuce production control with the employed model. Lower values of γ would create myopic agents that prioritize short-term rewards (e.g., minimize costs) while neglecting long-term economic performance through sustained crop growth.

The reward function used to train the RL agent is defined as the negative of the stage cost in (9). During the training process, the agent's objective is then defined by the expected value of the discounted cumulative reward:

$$\max_{\pi_{\mathrm{RL}}} \mathbb{E}_{\substack{p\sim\mathcal{U}(\Theta)\\(\mathbf{x},\mathbf{u})\sim\rho(p,\pi_{\mathrm{RL}})}} \left[\sum_{t=0}^{T-1} -\gamma^t \ell\left(u(t), x(t), x(t+1)\right)\right], \tag{14}$$

subject to the dynamics in (1), and where the state-input trajectory follows the distribution ρ induced jointly by the randomized model parameter and the exploration noise of the stochastic policy. The sign inversion enables the agent to maximize cumulative reward while maintaining alignment with the optimization goals of the MPC and hybrid approaches, allowing for fair performance comparisons across methods. As mentioned in Section 2, state constraint violations are penalized through the stage cost function (9). RL methods are sensitive to the relative scaling of these penalty terms. Excessively large penalty coefficients may cause the penalty term to dominate policy updates, resulting in conservative policies. By scaling the penalty terms to a magnitude comparable to the economic stage cost (6), stable policy learning and a balanced trade-off between economic performance and constraint satisfaction were observed. Section 5.1 discusses the effect of the penalty coefficients on controller performance in more detail.

The RL policy was trained with Stable-Baselines-3 (Raffin et al., 2021). The complete set of hyperparameters is listed in Table 1.

3.2. Model predictive control

At each time step t, given the current state $x(t)$ and previous input $u(t-1)$, the predicted state and input sequences over a horizon of length N are defined as:

$$\begin{aligned}\mathbf{x} &= (x(t), x(t+1), \dots, x(t+N)),\\ \mathbf{u} &= (u(t), u(t+1), \dots, u(t+N-1)).\end{aligned} \tag{15}$$

The controller then solves the following finite-horizon optimal control problem:

$$\min_{\mathbf{u},\mathbf{x}} \quad \sum_{k=t}^{t+N-1} \ell\left(u(k), x(k), x(k+1)\right) \tag{16a}$$

$$\text{s.t.} \quad x(k+1) = f(x(k), u(k), d(k), p) \tag{16b}$$

$$y(k) = h(x(k)) \tag{16c}$$

$$u_{\min} \le u(k) \le u_{\max} \tag{16d}$$

$$|u(k) - u(k-1)| \le \delta u_{\max} \tag{16e}$$

where all constraints must be satisfied at each prediction step $k \in \{t, \dots, t+N-1\}$. After solving this problem, only the first optimal input is applied to the system. The optimization is then solved again at every subsequent time step until the end of the growing period. This procedure implicitly defines a policy π_{MPC} that maps the current state $x(t)$, previous input $u(t-1)$, and future disturbances $\mathbf{d}_N(t)$ to the control input $u(t)$:

$$u(t) = \pi_{\mathrm{MPC}}\left(x(t), u(t-1), \mathbf{d}_N(t)\right), \tag{17}$$

where $\mathbf{d}_N(t)$ represents the future weather disturbances for the prediction horizon N. Note the difference in the prediction horizon with the RL agent, which uses a fixed six-hour horizon in its observation space.

The MPC optimization problem was implemented with the automatic differentiation framework CasADi (Andersson et al., 2019) and solved with the interior-point solver IPOPT (Wächter & Biegler, 2006).

3.3. Stochastic model predictive control

Stochastic Model Predictive Control (SMPC) extends the traditional MPC framework by explicitly incorporating uncertainty into the optimization problem. Unlike deterministic MPC, which assumes perfect knowledge of the prediction model, SMPC incorporates probabilistic information to optimize control inputs over a finite horizon. Many SMPC formulations propagate full probability distributions through the prediction horizon, but this quickly becomes intractable for nonlinear systems. Therefore, this work builds on scenario-based SMPC to approximate the optimization problem in (10). The scenario-based approach assumes that the uncertainty follows a known distribution, which is approximated by sampling a finite number of disturbance realizations. By optimizing over this scenario set, the controller captures the effect of uncertainty without propagating complete probability distributions through the prediction horizon. This discretization of the probability space yields a computationally tractable optimization problem.

At each time step, S scenarios are sampled from the uncertainty set. These sampled scenarios are then explicitly included in the optimization problem, allowing the controller to account for the uncertainty in the system dynamics. The optimization problem is then solved by minimizing the expected cost over all scenarios. Ideally, it would minimize over a trajectory of feedback policies as discussed in Section 2.2. However, in practice, this is computationally infeasible. Therefore, the optimization is performed over one trajectory of control inputs $\mathbf{u} = (u(t), u(t+1), \dots, u(t+N-1))$ shared by all scenarios and over S trajectories of predicted states $\mathbf{x}^{(i)} = (x^{(i)}(t), x^{(i)}(t+1), \dots, x^{(i)}(t+N))$, one for each scenario i. At each time step t, the following optimization problem is solved:

$$\min_{\mathbf{u},\mathbf{x}^{(1)},\dots,\mathbf{x}^{(S)}} \quad \sum_{i=1}^{S} \sum_{k=t}^{t+N-1} \ell\left(u(k), x^{(i)}(k), x^{(i)}(k+1)\right) \tag{18a}$$

$$\text{s.t.} \quad x^{(i)}(k+1) = f(x^{(i)}(k), u(k), d(k), p^{(i)}) \tag{18b}$$

$$y^{(i)}(k) = h(x^{(i)}(k)) \tag{18c}$$

$$u_{\min} \le u(k) \le u_{\max} \tag{18d}$$

$$|u(k) - u(k-1)| \le \delta u_{\max} \tag{18e}$$

in which, $p^{(i)}$ are the sampled parameters, and $x^{(i)}(t)$, $y^{(i)}(t)$ represent the corresponding state and output trajectories. Moreover, the constraints are enforced for each $k \in \{t, \dots, t+N-1\}$ and for each $i \in \{1, \dots, S\}$. As in MPC (Section 3.2), only the first control input is applied at each time step, and the procedure is repeated, thereby defining policy πSMPC:

$$u(t) = \pi_{\mathrm{SMPC}}\left(x(t), u(t-1), \mathbf{d}_N(t)\right) \tag{19}$$

3.4. RL-SMPC

The RL-SMPC proposed in this work combines the strengths of both RL and SMPC to enhance the performance of greenhouse production systems under parametric uncertainties. Ideally, the optimization should

be performed over a trajectory of feedback policies as discussed in Section 2.2. Since this is not feasible in practice, the RL policy obtained in Section 3.1 is used to define a feedback parameterization for the SMPC optimization problem. A natural idea is to directly embed the RL policy in the feedback parameterization. While theoretically appealing, this approach would embed the neural network of the policy π_{RL} into the SMPC optimization problem, which is not computationally appealing and should be avoided.

Instead, at each time step, S scenarios are sampled from the uncertainty set $\mathcal{U}(\Theta)$, as with the scenario-based SMPC method proposed in the previous subsection. A rollout (simulation) of the RL policy is then considered for each scenario from t to $t+N$. Thus, for a given scenario i, a state trajectory $\hat{x}^{(i)}(t)$ and the corresponding input trajectory $\hat{u}^{(i)}(t)$ are obtained from the RL policy. These trajectories are then used to define a feedback parameterization $\pi^{(i)}(x^{(i)}(t), \theta(t))$.

For a zero-order approximation, the feedback parameterization for each sample i is given by:

$$\pi^{(i)}(x^{(i)}(t), \theta(t)) = \hat{u}^{(i)}(t) + \theta(t), \tag{20}$$

in which $\hat{u}^{(i)}(t)$ is now a constant defined before the optimization problem. In other words, the optimization is performed over a trajectory of decision variables $\boldsymbol{\theta} = (\theta(0), \dots, \theta(N-1))$ where each $\theta(t)$ represents a shift from the prescribed RL policy's control input $\hat{u}^{(i)}(t)$, applied uniformly across all samples i.

Terminal cost

The proposed RL-SMPC incorporates a terminal cost and a terminal region constraint into the optimization problem of each scenario. The terminal cost is intended to approximate the cumulative cost beyond the finite prediction horizon using the policy π_{RL}, potentially enhancing long-term performance. While the true return of the RL policy π_{RL} from time t to the terminal harvest time T is given by:

$$\begin{aligned} J_{\pi_{\text{RL}}}(x(t), t) = &\sum_{k=t}^{T} \ell\big(u_{RL}(k), x(k), x(k+1)\big) \\ &\text{s.t. } x(k+1) = f(x(k), u_{RL}(k), d(k), p) \\ &\quad u_{RL}(k) = \pi_{\text{RL}}(x(k), u(k-1), \mathbf{d}_N(k), k) \end{aligned} \tag{21}$$

this return cannot be directly used as the terminal cost. Instead, $J_{\pi_{\text{RL}}}(x(t), t)$ is approximated by sampling multiple trajectories and training a neural network $\tilde{J}_\phi$ to predict the return of policy π_{RL} given a certain state. This setup allows the controller to incorporate long-term effects into its decision-making without increasing the prediction horizon. To train $\tilde{J}_\phi$, an approach based on expected return learning is used. From a reference trajectory $x^{\text{n}}(t)$, 1000 time steps $t^{(i)}$ are randomly selected from the interval $\{0, 1, \dots, T\}$. For each $t^{(i)}$, a corresponding initial state $x^{(i)}(t^{(i)})$ is sampled uniformly from a range around the nominal value

$$x^{(i)}(t^{(i)}) \sim \mathcal{U}\big(x^{\text{n}}_{\min}(t^{(i)}), x^{\text{n}}_{\max}(t^{(i)})\big), \tag{22}$$

with bounds defined as:

$$\begin{aligned} x^{\text{n}}_{\min}(t) &= x^{\text{n}}(t)(1-\sigma), \\ x^{\text{n}}_{\max}(t) &= x^{\text{n}}(t)(1+\sigma), \end{aligned} \tag{23}$$

where $\sigma = 0.5$ controls the spread of the sampling space. For each sampled state, a closed-loop rollout under policy π_{RL} is simulated until time T. The cost of each trajectory is computed by (21). Each sample $\big(x_{\text{DW}}(t^{(i)}), t^{(i)}, J_{\pi_{\text{RL}}}(\cdot)\big)$ becomes part of dataset $\mathcal{D}$, which is split 80/20 into training and validation sets. A neural network with parameters ϕ is then trained to approximate $J_{\pi_{\text{RL}}}(\cdot)$ based on $x_{\text{DW}}(t^{(i)})$ and $t^{(i)}$. The model minimizes the following loss using the Adam optimizer Table 2:

$$\mathcal{L}(\phi, \mathcal{D}) = \frac{1}{N_s} \sum_{i=1}^{N_s} \Big(\tilde{J}_\phi(x^{(i)}_{\text{DW}}(t^{(i)}), t^{(i)}) - J_{\pi_{\text{RL}}}(x^{(i)}(t^{(i)}), t^{(i)}) \Big)^2.$$

Finally, $\tilde{J}_\phi$ is integrated into the RL-SMPC's cost function via a second-order Taylor approximation around the center of the terminal region, rather than being integrated directly.

Table 2
Neural Network Hyperparameters of $\tilde{J}_\phi$.

Parameter	Value
Hidden layers	2
Neurons per layer	128
Batch size	1024
Learning rate	$1 \cdot 10^{-3}$
Buffer size	1024
Activation	`tanh`

Terminal region constraint

For each scenario trajectory $\hat{x}^{(i)}(t)$, a corresponding terminal region constraint is defined based on its predicted state at the end of the horizon. Specifically, at each time step t, S terminal regions $\mathbb{X}_f^{(i)}$ are constructed around the final states $\hat{x}^{(i)}(t+N)$ as:

$$\mathbb{X}_f^{(i)}(\hat{x}^{(i)}(t+N)) = \big\{ x \in \mathbb{X} \mid \|x - \hat{x}^{(i)}(t+N)\| \le \epsilon \big\} \tag{24}$$

where $\epsilon > 0$ determines the size of each region. The terminal region constraint for each scenario is then enforced as $x^{(i)}(t+N) \in \mathbb{X}_f^{(i)}(\hat{x}^{(i)}(t+N))$. This process is repeated at every time step following the scenario rollout under the current policy.

Finite-horizon optimal control problem

The RL-SMPC optimization problem solved at each time step t is then defined as:

$$\min_{\boldsymbol{\theta}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(S)}} \quad \sum_{i=1}^{S} \sum_{k=t}^{t+N-1} \Big(\ell(u^{(i)}(k), x^{(i)}(k), x^{(i)}(k+1)) + \tilde{J}_\phi(x^{(i)}(t+N), t+N) \Big) \tag{25a}$$

$$\text{s.t.} \quad x^{(i)}(k+1) = f(x^{(i)}(k), u^{(i)}(k), d(k), p^{(i)}) \tag{25b}$$

$$y^{(i)}(k) = h(x^{(i)}(k)) \tag{25c}$$

$$u^{(i)}(k) = \hat{u}^{(i)}(k) + \theta(k) \tag{25d}$$

$$u_{\min} \le u^{(i)}(k) \le u_{\max} \tag{25e}$$

$$|u^{(i)}(k) - u^{(i)}(k-1)| \le \delta u_{\max} \tag{25f}$$

$$x^{(i)}(t+N) \in \mathbb{X}_f^{(i)}(\hat{x}^{(i)}(t+N)) \tag{25g}$$

where the constraints (25b–25f) need to be enforced for each $k \in \{0, \dots, N-1\}$ and each scenario $i \in \{1, \dots, S\}$, while (25g) needs to be enforced for each scenario i. At each time step, the RL-SMPC optimization problem in (25) is solved, and only the first predicted input is applied to the system. This procedure is repeated at every subsequent time step until the end of the growing period, thereby defining policy $\pi_{\text{RL-SMPC}}$:

$$u(t) = \pi_{\text{RL-SMPC}}\big(x(t), u(t-1), \mathbf{d}_N(t)\big). \tag{26}$$

The learning framework L4CasADi (Salzmann et al., 2024, 2023) is used to incorporate the learned terminal cost function into the optimal control problem.

3.5. Performance metrics

This section introduces four metrics for assessing closed-loop controller performance of the four controllers presented in Sections 3.1–3.4.

The first metric is based on the cumulative sum of the stage cost (9) over the entire growing period. This metric, referred to as cumulative reward $\mathcal{J}$, quantifies how well each controller optimizes the given cost function. Specifically, for a given controller $\pi_{(\cdot)}$, the cumulative reward is defined as:

$$\begin{aligned} \mathcal{J}(\pi_{(\cdot)}) = &\sum_{t=0}^{T-1} -\ell\big(u(t), x(t), x(t+1)\big) \\ &\text{s.t. } x(t+1) = f(x(t), u(t), d(t)), \\ &\quad u(t) = \pi_{(\cdot)}\big(x(t), u(t-1), \mathbf{d}_N(t), t\big). \end{aligned} \tag{27}$$

Negating the stage cost function inverts the objective, so a higher value means better performance for this metric.

The next two metrics decompose the stage cost into its economic component (6), and its penalty component (7). Yielding the Economic Performance Indicator (EPI) and the cumulative penalty. The closed-loop EPI is:

$$\begin{aligned} \text{EPI}(\pi_{(\cdot)}) = &\sum_{t=0}^{T-1} -\ell_{\text{e}}\big(u(t), x(t), x(t+1)\big), \\ \text{s.t. } &x(t+1) = f(x(t), u(t), d(t)), \\ &u(t) = \pi_{(\cdot)}\big(x(t), u(t-1), \mathbf{d}_N(t), t\big). \end{aligned} \tag{28}$$

and the cumulative penalty is:

$$\begin{aligned} \text{Cumulative penalty}(\pi_{(\cdot)}) = &\\ &\sum_{t=0}^{T-1} -\ell_{\text{p}}\big(x(t)\big), \\ \text{s.t. } &x(t+1) = f(x(t), u(t), d(t)), \\ &u(t) = \pi_{(\cdot)}\big(x(t), u(t-1), \mathbf{d}_N(t), t\big). \end{aligned} \tag{29}$$

Note that the closed-loop EPI metric is equivalent to (5). Again, the two metrics are summed with a reversed sign, maximizing the EPI and minimizing the cumulative penalty. Both metrics serve an illustrative purpose, indicating which component of the cost function the controller emphasizes.

Finally, a relative performance metric is introduced. This metric allows for a pairwise comparison between RL-SMPC and the three standalone controllers, and is defined as follows:

$$\Delta\%\text{Cumulative reward} = 100 \times \frac{\mathcal{J}(\pi_{\text{RL-SMPC}}) - \mathcal{J}(\pi_{(\cdot)})}{\mathcal{J}(\pi_{(\cdot)})}. \tag{30}$$

4. Simulation results

To evaluate the performance of RL-SMPC, four simulation experiments were performed. Each simulation spans a fixed interval of 40 days, beginning on February 9, 2014, with a discretization time of Δt = 1800(s) = 30(min). The weather data used in the simulations are real-world measurements recorded at Bleiswijk, The Netherlands (Kempkes et al., 2014).

Closed-loop controller performance was evaluated based on the cumulative reward (27), the EPI (28), and the cumulative penalty (29). The coefficients of the underlying cost functions (6) and (7) are given in Table 3. The three indoor climate states were constrained with upper and lower bounds, while the crop state was unbounded. Similarly, all three control inputs are constrained. The bounds for these constraints are defined in Table 4. For consistency, the initial conditions are fixed across all simulations as follows:

$$u(0) = (0\ 0\ 50)^{\text{T}},$$
$$x(0) = (0.0035\ 0.001\ 15\ 0.008)^{\text{T}}$$

Both SMPC and RL-SMPC employ an identical cost function (9) in their finite-horizon optimization frameworks (18), (25), with cost and penalty coefficients, and input and output bounds as listed in Tables 3 and 4. In contrast, the nominal MPC formulation (16) imposes an upper bound of 78% on relative humidity yRH. In industrial practice, this style of empirical constraint tightening is common and therefore replicated here to ensure a realistic comparison.

The remainder of this section presents the results of four simulation experiments. First, Section 4.1 compared all four controllers, RL-SMPC, SMPC, MPC, and RL, under 10% ($\delta = 0.1$) parametric model uncertainty. Next, Section 4.2 presents a comparison of the computational cost between RL-SMPC and SMPC. The third experiment, in Section 4.3, investigated the relative improvement in performance of RL-SMPC over the other three controllers across eight uncertainty levels and eight prediction horizons. Finally, Section 4.4 presents the results of the ablation study.

Table 3
Economic stage cost and linear penalty function coefficients.

Variable	Value	Unit	Description
Δt	1800	s	Discretization interval
c_{CO_2}	0.1906	€/kg	CO_2 price coefficient
c_{heat}	0.1281	€/kWh	Heating price coefficient
c_{DW}	22.29	€/kg{DW}/m^2	Crop dry-weight price
λ_{CO_2}	5×10^{-5}	–	Penalty for CO_2 violations
$\lambda_T^{\min}$	3×10^{-3}	–	Penalty for T lower-bound violations
$\lambda_T^{\max}$	5×10^{-3}	–	Penalty for T upper-bound violations
λ_{RH}	7×10^{-4}	–	Penalty for relative-humidity violations

Table 4
Output (y) and control input (u) constraints used in the (S)MPC formulation.

Variable	Bounds Lower	Upper	Unit
Output			
y_{CO_2}	500	1600	ppm
y_T	10	20	°C
y_{RH}	0	80	%
Input			
u_{CO_2}	0	1.2	mg/m^2/s
u_{heat}	0	150	W/m^2
u_{vent}	0	7.5	m^3/m^2/s

4.1. Performance under parametric model uncertainty

For the first simulation experiment, all four controllers (RL, MPC, SMPC, and RL-SMPC) approximated the optimal control problem defined in (10) under 10% ($\delta = 0.1$) parametric model uncertainty. This experiment assessed whether, and to what extent, RL-SMPC can improve greenhouse production control under parametric uncertainty. Each controller was evaluated over ten simulation runs using unique random seeds to sample different model parameters in each run. Additional runs did not narrow the confidence interval but significantly increased the computation time, so ten runs were deemed sufficient. The finite-horizon controllers, i.e., RL-SMPC, SMPC, and MPC, were evaluated with a prediction horizon H ranging from $1-8$ hours. With a discretization time of 30 minutes, this corresponds to horizons of $N = [2-16]$ control steps. Each controller was executed for ten simulation runs for all eight prediction horizons. The closed-loop performance metrics are averaged across ten simulation runs and visualized in Fig. 2.

The comparison shows that RL-SMPC outperformed SMPC on short prediction horizons (Fig. 2). At a one-hour prediction horizon, RL-SMPC achieved a cumulative reward of (2.547 ± 0.032), an increase of 12.6% with respect to SMPC. This advantage remained significant up to a four-hour horizon. At a four-hour prediction horizon, RL-SMPC converged to its asymptotic performance (2.849 ± 0.029) and matched the performance of SMPC with an eight-hour prediction horizon (2.871 ± 0.029). Beyond a prediction horizon of six hours, RL-SMPC and SMPC converged toward similar asymptotic performance.

The improvement of RL-SMPC over SMPC was driven by a higher EPI of 3.641 ± 0.014, an increase of (0.347) at H = 1, while incurring a slightly larger cumulative penalty (0.065). The experiment also showed that the RL policy performed similarly to MPC with a four-hour prediction horizon, and outperformed SMPC up to a two-hour prediction horizon. RL's high economic performance (3.905) came at the expense of more frequent output constraint violations, reflected by a higher cumulative penalty (1.323).

An illustrative snapshot of the open-loop output (y) trajectories just before sunrise on day 13 of the simulation highlights the differences

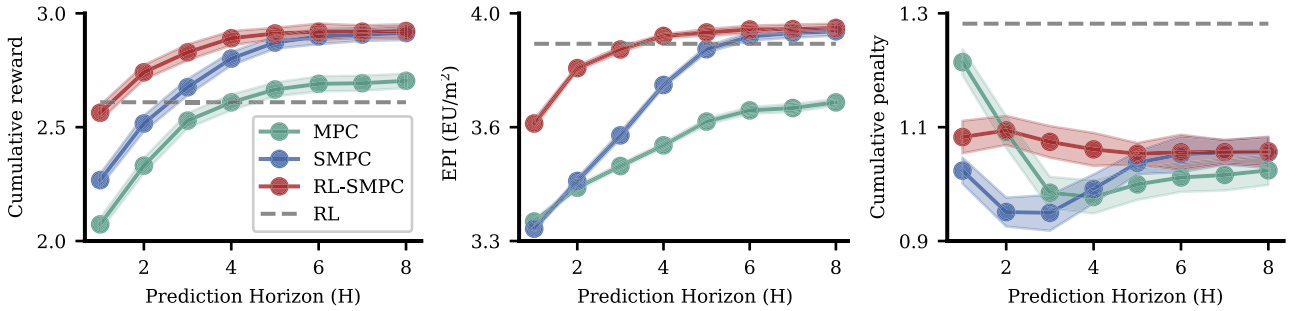


Fig. 2. Comparison of closed-loop controller performance under parametric uncertainty ($\delta = 0.1$). This figure visualizes the cumulative sum of three metrics averaged across ten simulation runs for MPC, SMPC, RL, and RL-SMPC against the controller's prediction horizon. The shaded area represents the 99% confidence interval. Simulations were executed under parametric uncertainty with $\delta = 0.1$. **Left:** The cumulative reward as defined in (27). **Middle:** The EPI as defined in (28). **Right:** The cumulative sum of the linear penalty function as defined (29). Note that higher penalty values correspond to more state violations.

between RL-SMPC and SMPC (Fig. 3). RL-SMPC raised the indoor CO_2 concentration before the global radiation increases, while SMPC remained at the lower bound of 500 (ppm) for the entire prediction horizon. The raised CO_2 levels were induced by the RL-rollouts, which constrain the terminal state to lie within the accentuated region. The closed-loop trajectories (Fig. 4) confirm that RL-SMPC begins to raise the CO2 levels earlier in the day by increasing the CO_2 supply rate and reducing the ventilation rate, thereby accelerating crop growth during daytime. In contrast, both controllers find similar closed- and open-loop solutions for temperature and relative humidity. Interestingly, raising indoor temperature through heating during the daytime was not considered beneficial.

Both the open-loop solution and the closed-loop solution exhibited the greatest uncertainty in relative humidity (Figs. 3, 4). For the open-loop trajectories, relative humidity showed the largest uncertainty bounds. This variability was reflected in the closed-loop results, where relative humidity fluctuated rapidly near its upper boundary of 80%. These fluctuations indicate that the controllers put considerable effort into imposing the humidity constraint.

Note that the results presented for RL and RL-SMPC are based on a single RL policy trained with a fixed random seed. Appendix B.2 provides an analysis of the performance variability of both RL and RL-SMPC across different random seeds.

4.2. Computational complexity

The second experiment evaluated the online computational cost of SMPC and RL-SMPC for six prediction horizons ($H = [1, \dots, 6]$ hours) with a fixed number of sampled scenarios ($S = 10$). This experiment compared the average computation time per time step, measured in seconds, against closed-loop controller performance quantified by the cumulative reward (27). The results illustrate the trade-off between computational cost and controller performance for both SMPC and RL-SMPC. Each controller was simulated for ten independent runs for all six prediction horizons. Fig. 5 shows the average computation time per time step plotted against the corresponding closed-loop performance.

At short prediction horizons, RL-SMPC achieved significant improvements over SMPC in terms of cumulative reward while maintaining comparable computational costs. Specifically, at a one-hour prediction horizon, the improvement of 12.6%, as reported in Section 4.1, had a small trade-off in computational cost, 0.21 seconds per time step for RL-SMPC versus 0.22 seconds for SMPC. This pattern persisted across all tested prediction horizons, where RL-SMPC consistently outperformed SMPC while requiring slightly more average computation time per time step. For a six-hour horizon, RL-SMPC was even marginally faster than SMPC, with 3.06 versus 3.24 seconds.

RL-SMPC achieved its asymptotic performance at a four-hour prediction horizon, whereas SMPC required longer horizons to reach similar performance levels. Furthermore, the average computation time increased approximately exponentially with the prediction horizon for both controllers. Consequently, RL-SMPC reached its asymptotic performance with an average computation time of 1.10 seconds, while SMPC required 1.33 seconds at $H = 5$ and up to 3.24 seconds at $H = 6$.

Appendix B.1 provides an analysis of the effect of the number of sampled scenarios (S) on computational cost and closed-loop performance.

All runtime measurements were obtained on a workstation with an Intel(R) Xeon(R) W-2133 CPU @ 3.60GHz. The SMPC and RL-SMPC nonlinear programs were implemented in CasADi version 3.6.7 using Python 3.11.0 and solved using IPOPT version 3.14.3.

4.3. Varying model parameter uncertainty and prediction horizons

The second experiment evaluated controller performance across eight levels of parametric model uncertainty ($\delta = [0.025, 0.05 \dots, 0.175, 0.20]$) and eight prediction horizons ($H = [1, \dots, 8]$ hours). This experiment was conducted to evaluate the robustness of RL-SMPC across varying prediction horizons and degrees of parametric uncertainty. Fig. 6 visualizes the relative performance of RL-SMPC against each of the other three controllers.

RL-SMPC outperformed MPC at every uncertainty level and prediction horizon. The margin between the two controllers increased with shorter horizons and higher uncertainty. Conversely, at a one-hour prediction horizon, RL matched RL-SMPC and even surpassed it when $\delta \leq 0.1$. In the most extreme case at $\delta = 0.025$ RL had an 8.1% performance gain over RL-SMPC. However, with longer horizons or increased uncertainty, RL-SMPC generally improved over RL.

In comparison with SMPC, a different pattern emerged. The uncertainty level had a low effect on the relative performance between RL-SMPC and SMPC. Regardless of the uncertainty level, RL-SMPC consistently outperformed SMPC at short horizons, while for horizons of six hours or longer, the two controllers performed nearly identically, findings in line with Fig. 2.

4.4. Ablations

In the final experiment, an ablation study was performed to quantify the contribution of individual algorithmic components to RL-SMPC's performance (Fig. 7). Within the RL part, future weather disturbances $\mathbf{d}_N(t+1)$ were removed from the observation space, leaving only the current weather measurement $d(t)$. A new RL policy and terminal cost function were trained under this new setting, and reevaluated RL-SMPC closed-loop performance in ten simulation runs at $\delta = 0.1$. Excluding future weather disturbances reduced the cumulative reward of the RL

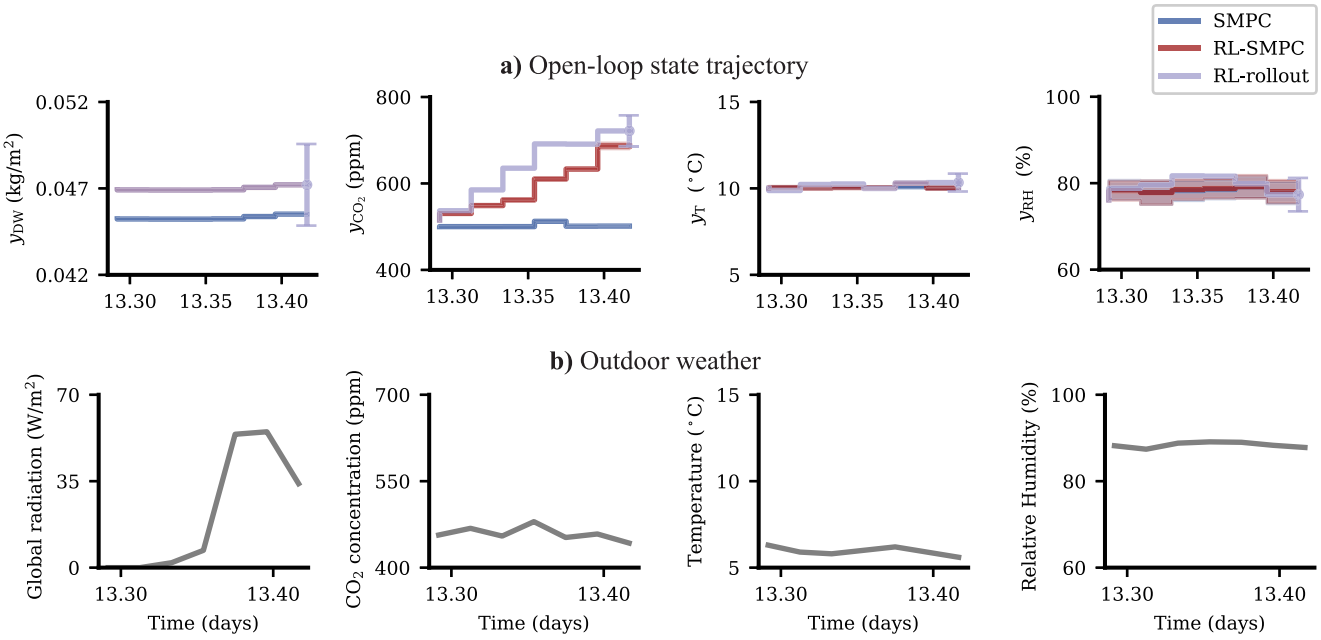


Fig. 3. Comparison of the open-loop solution of both the controllers by solving (18) **and** (25). Visualization of the average state and input trajectories, and the corresponding weather disturbance for RL-SMPC and SMPC with a three-hour prediction horizon. The graphs represent the mean prediction over $S = 20$ with one standard deviation shaded. Also, the sampled RL rollout trajectory is visualized. This trajectory represents the average over the sampled trajectories. The vertical bar at the end of the prediction horizons represents the terminal region constraint. These results were obtained under parametric model uncertainty of $\delta = 0.1$.

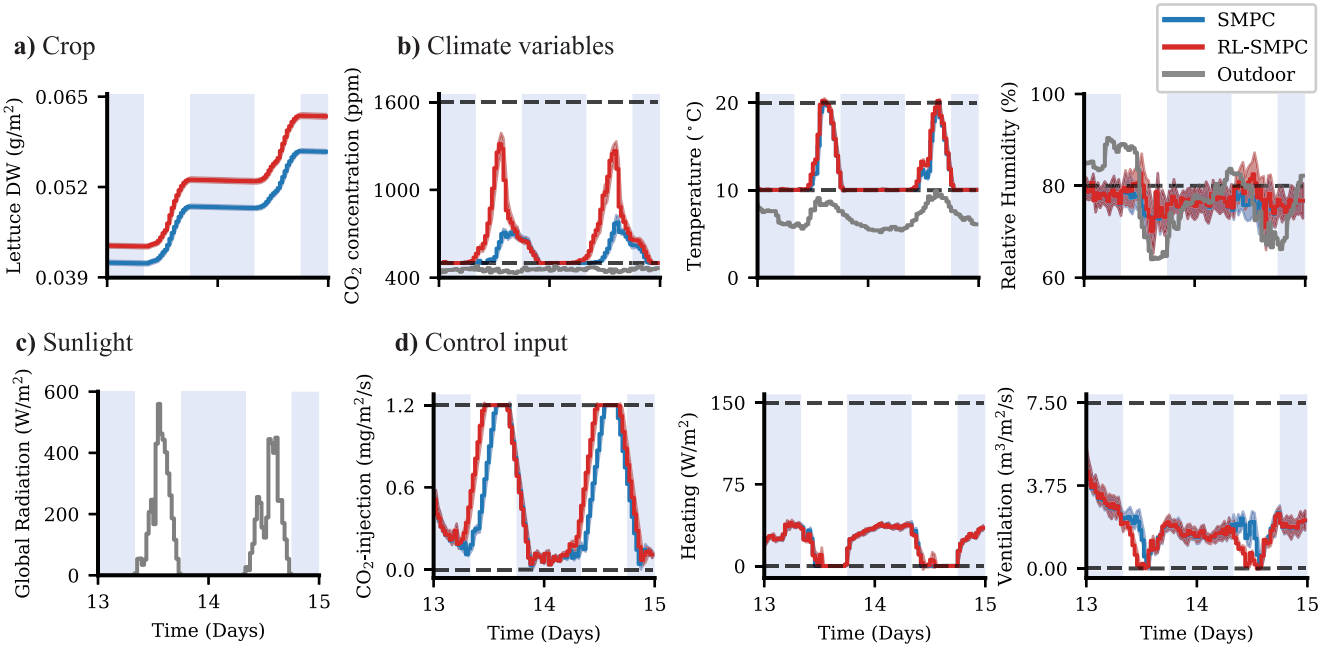


Fig. 4. Comparison of the closed-loop trajectories of both the controllers by approximating (10) with a one-hour prediction horizon. Each plot visualizes days 13 and 14 of the simulation. The solid lines represent the mean taken over ten simulation runs, with one standard deviation shaded. The results were obtained under parametric model uncertainty of $\delta = 0.1$. The horizontal dashed lines in **b** and **c** represent lower and upper bounds. The outdoor weather variables are colored grey. Shaded areas represent night time ($d_{\text{iGlob}} = 0$).

policy by 9.3%, driven by a larger drop in EPI than the improvement in the cumulative penalty. Using that new RL policy in RL-SMPC lowered RL-SMPC's performance at shorter horizon lengths. The largest drop was observed at a one-hour prediction horizon, decreasing the cumulative reward by 3.7%.

Three elements of the RL-SMPC optimization formulation (25), the terminal cost function $\tilde{J}\phi$, the non-linear RL feedback policy $\hat{u}$, and the terminal region constraint X_N, were removed one at a time. The terminal region constraint had the largest impact, primarily for prediction horizons lower than six hours. The EPI dropped significantly for these prediction horizons, while the cumulative penalty improved slightly. Overall behavior without the terminal region constraint closely matched SMPC, also see Fig. 2. Removing the non-linear RL feedback policy reduced the algorithm's asymptotic performance, mainly because of a

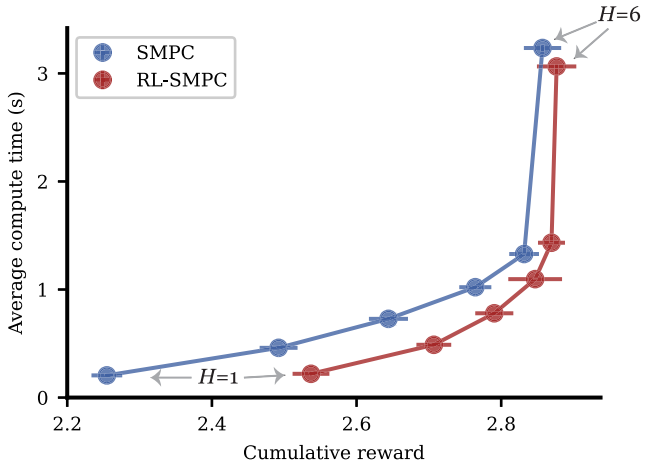


Fig. 5. Computational complexity of SMPC and RL-SMPC. Visualization of the controller's cumulative reward (27) averaged over ten simulation runs, plotted against the average compute time per time step in seconds. The horizontal error bar indicates the 95% confidence interval. The variance in average compute time was too low to report for either of the methods.

higher cumulative penalty. Setting $\tilde{J}_\phi(\cdot) = 0$ resulted in nearly identical performance of RL-SMPC, suggesting that the terminal cost function is the least critical of the three components under the tested conditions.

5. Discussion

This section discusses the results of the simulation experiments presented in Section 4 in the context of the main research question: *To what extent does RL-SMPC improve greenhouse production control, compared to SMPC and MPC, under parametric uncertainty?* More precisely, Section 5.1 discusses the performance and behavior of the RL-SMPC controller as defined in (25) compared to baseline controllers: MPC, SMPC, and RL. Section 5.2 studies the computational cost of RL-SMPC and SMPC. Next, Section 5.3 analyzes the relative performance of RL-SMPC against the other three controllers across eight levels of uncertainty. Finally, Section 5.4 discusses the contributions of individual algorithmic components through the ablation study.

5.1. Performance of RL-SMPC

This work found that RL-SMPC significantly outperformed SMPC on prediction horizons up to four hours, as shown in Fig. 2. Additionally, the RL-SMPC improved over MPC at all eight evaluated prediction horizons. These improvements were driven by the RL policy, which learned long-term relationships between model input and the optimization objective. This information was transferred to RL-SMPC's finite-horizon optimization problem defined in (25). The exemplary open-loop solution in Fig. 3 demonstrated how RL guided RL-SMPC towards different solutions that resulted in higher long-term economic performance. Increasing CO_2 concentration before sunrise to maximize crop growth during daylight demonstrated RL-SMPC's focus on long-term performance regardless of near-sighted prediction horizons.

At longer prediction horizons, RL-SMPC and SMPC converged to similar asymptotic performance. This work argues that three factors contributed to this behavior. First, as the prediction horizon lengthens, the initial control input computed by RL-SMPC relies less on the terminal cost and constraint derived from the RL policy, since the long horizon itself captures most of the long-term controller performance. Consistent with prior work showing that extending the horizon yields greater performance gains than improving the learned value or policy functions (Bertsekas, 2024; Msaad et al., 2025; Reiter et al., 2025). Second, in this problem definition (10a), SMPC did not suffer from extensive growth of uncertainty at longer horizons impacting performance, as illustrated by the relatively small uncertainty bounds of the open-loop solution in Fig. 3. However, in systems that exhibit significant growth in uncertainty over the prediction horizon, one might expect SMPC to become overly conservative. Since RL-SMPC incorporates non-linear feedback from the RL policy, it is expected to maintain its performance over SMPC with longer prediction horizons. Third, the lettuce greenhouse model employed in this simulation study is relatively simple and therefore does not capture long-term relationships between climate and crop. Consequently, RL-SMPC and SMPC converged to similar solutions at horizons longer than six hours. When using more complex greenhouse models (Katzin et al., 2020; Van Laatum et al., 2025) that capture climatic effects on crop growth over multiple days or weeks, it is expected that RL-SMPC would provide a benefit at longer horizons.

The solutions from both RL-SMPC and SMPC exhibited the largest sensitivity to relative humidity under parametric model uncertainty. This result is illustrated in Figs. 3 and 4, which show the largest uncertainty bounds for humidity in both the open-loop and closed-loop solutions, and is consistent with a previous sensitivity analysis of the employed lettuce greenhouse model (Van Henten, 2003). Unexpectedly, crop growth was much less affected by parametric model uncertainty during closed-loop operation. This outcome could be dedicated to two aspects of the problem definition. First, the controllers are optimized for a single deterministic weather trajectory, and because biomass accumulation is mainly driven by global radiation, weather-induced variability in growth is small. Second, randomizing the model parameters at each time step may have little effect on the slow biomass dynamics, so its long-term impact on crop growth remains limited.

To have a fair comparison between controllers, this work proposed a penalty-based method for handling state constraint violations, as introduced in Section 2. The choice of these penalty coefficients influences the performance of individual methods. The RL policy is most sensitive to tuning these coefficients, which in turn influences the RL-SMPC controller. If the penalty terms are not scaled to a magnitude comparable to the economic stage cost, the RL policy may become overly conservative or allow excessive state constraint violations. In such extreme cases, the learned policy may bias RL-SMPC toward sub-optimal solutions, reducing its advantage over SMPC at the same prediction horizon. With the penalty coefficients reported in Table 3, such behavior was not observed. In preliminary experiments, order-of-magnitude changes to the penalty coefficients substantially influenced the RL policy and, in turn, RL-SMPC performance.

Besides the factors discussed above that influence RL-SMPC performance, the proposed method may be sensitive to the random seed used in the RL training process. As mentioned in Section 4.1, the results presented for RL and RL-SMPC are based on a single RL policy trained with one random seed. To assess the robustness of this method against this choice, Appendix B.2 presents an additional simulation experiment evaluating the performance of RL and RL-SMPC across multiple random seeds. Though some variance is observed in the performance of standalone RL policies across different seeds, this variance is substantially reduced when the policies are integrated with RL-SMPC. These findings confirm the robustness of RL-SMPC even when using RL policies that exhibit variability in performance due to stochasticity in the training process.

The performance advantages discussed above were obtained under several simplifying assumptions. Specifically, this study assumed perfectly accurate system measurements, the use of actual future weather disturbances in the optimization problems, perfect information about market prices, and fixed harvest time. In practice, these conditions are not perfectly known due to measurement and prediction inaccuracies, resulting in errors and uncertainty. Therefore, the overall controller performance should be interpreted as a performance bound, i.e., the performance that can be achieved if perfect predictions and measurements are available to the controllers. However, previous work suggests some of these assumptions can be relaxed in practice with appropriate

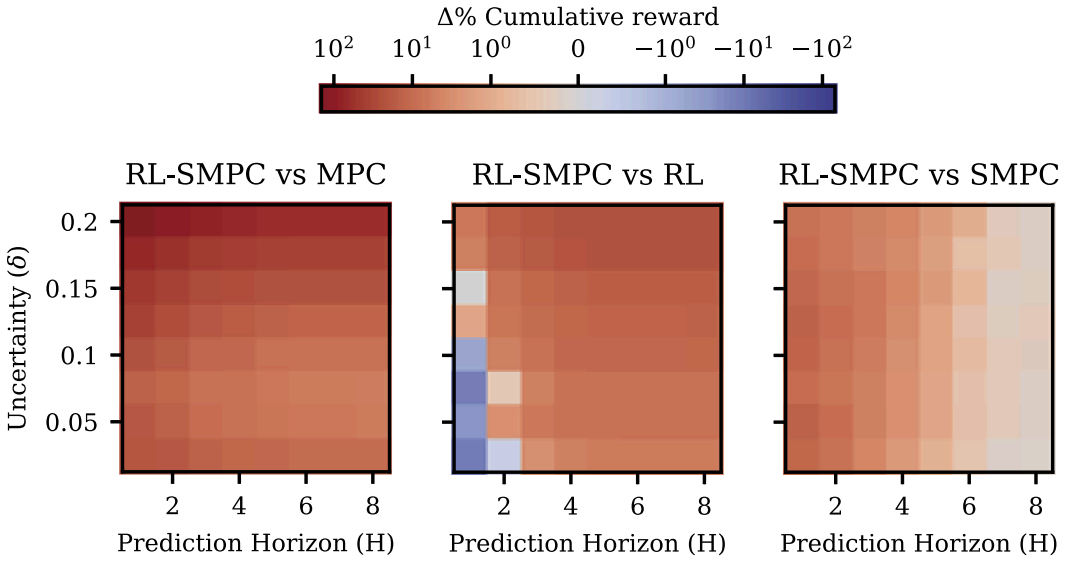


Fig. 6. Relative performance of RL-SMPC compared to MPC, RL, and SMPC for varying prediction horizons and parametric uncertainties. Each heatmap visualizes the relative performance, computed with (30), as a function of the prediction horizon and the parametric uncertainty. From left to right, RL-SMPC is compared against MPC, RL, and SMPC. Red indicates RL-SMPC outperforms the opposing controller, and blue vice versa. Note the symmetric logarithmic scale of the colorbar to also cover negative values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

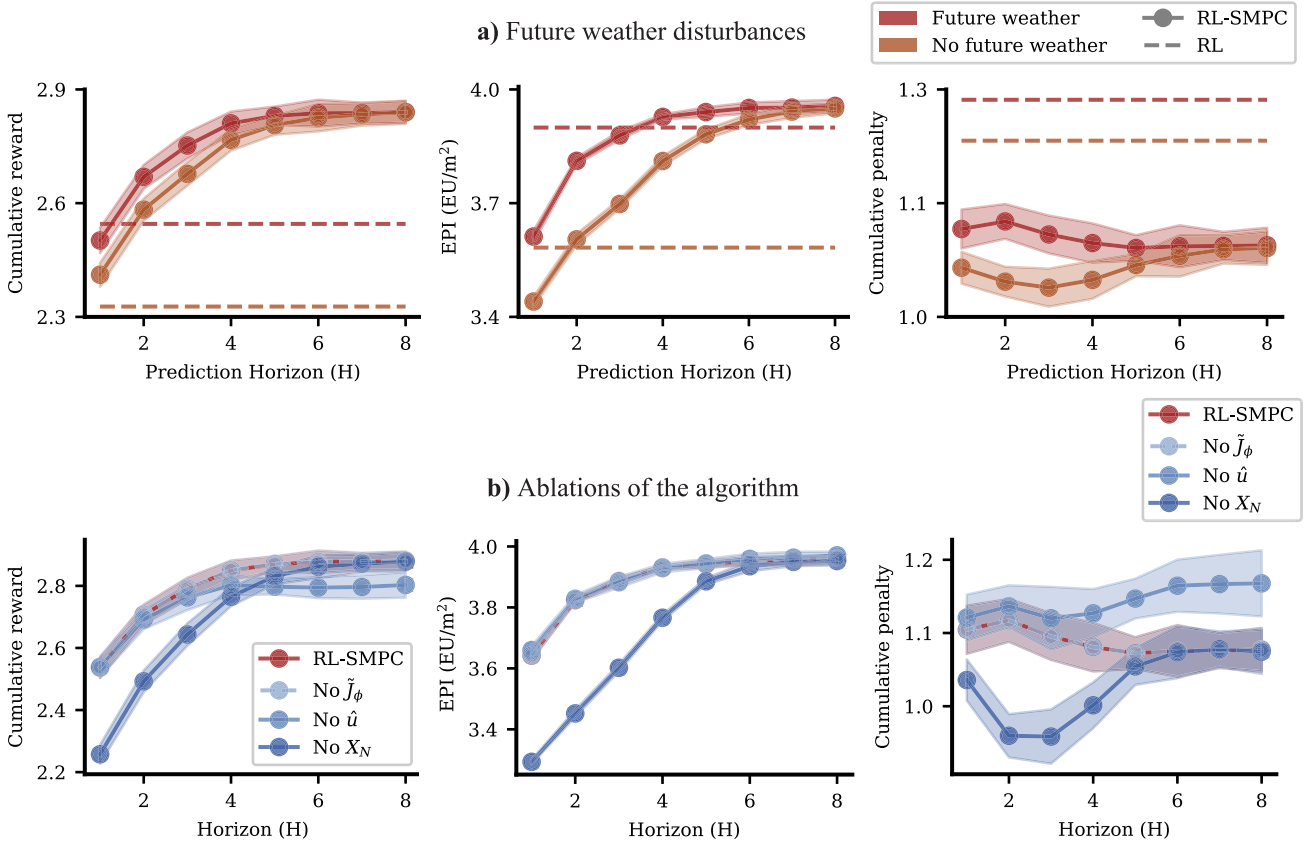


Fig. 7. Ablation study results for RL-SMPC. These graphs visualize the performance of RL-SMPC and variations of the algorithm when removing specific components. The cumulative reward, EPI, and penalty are shown. The cumulative reward is calculated via (27), the EPI and penalty calculated by (28), and (29). **a)** Shows RL-SMPC and RL optimized and trained with and without future weather disturbances in the observation space of the RL policy. The dashed line represents the performance of the RL policy. **b)** Shows RL-SMPC when removing some components from the optimization problem (25). Where $\tilde{J}\phi$ represents the terminal cost function, $\hat{u}$ the sampled feedback policy, and XN the terminal region constraint.

techniques. Boersma et al. (2022) already showed that ensemble Kalman filtering can accurately estimate crop dry weight from indoor climate measurements, even in the presence of modeling errors due to incorrect model parameters and uncertain initial conditions. Additionally, Kuijpers et al. (2022) demonstrated that realistic weather forecast errors have a minimal effect on controller performance when properly accounted for using robust MPC.

While a fixed harvest time was assumed for comparability across experiments, commercial greenhouse operations typically harvest once each lettuce head reaches a target weight. To better reflect this practice, the controller stage cost could be reformulated to minimize the time and operational costs required to reach the target weight. This potential extension to the proposed method should yield similar performance increments as presented in this work. These prior findings and potential extensions to the proposed method indicate that RL-SMPC could maintain its performance advantages under realistic conditions with suitable state estimation, robust optimization techniques, and objective reformulation.

Finally, the generalizability of the proposed approach is considered. This work presented results based on a single greenhouse crop model. However, with some modifications, the proposed RL-SMPC approach should be able to generalize to different greenhouse systems. Since the proposed RL-SMPC approach is model-agnostic, changes to the system, such as structural modifications in greenhouse design or different crop species, must be incorporated into the employed controller model. Different crops may introduce distinct challenges compared to lettuce. For instance, the economic performance of fruit-bearing crops such as tomatoes primarily depends on fruit yield, which should be incorporated into the stage cost function. Given these potential modifications and the model-agnostic nature of RL-SMPC, similar performance improvements could be expected across different greenhouse systems.

5.2. Computational complexity

The analysis of the computational cost per time step and closed-loop controller performance demonstrates the superior performance of RL-SMPC over SMPC (Fig. 5). As shown in Section 5.1, for short prediction horizons ($H \leq 4$), RL-SMPC achieved significantly higher performance while requiring a similar computational cost. Moreover, RL-SMPC reached its asymptotic performance with an average computation time of 1.10 seconds per time step, whereas SMPC required 3.24 seconds to attain a comparable performance level (2.85 ± 0.04 for RL-SMPC versus 2.86 ± 0.03 for SMPC). These results indicate that scenario-based SMPC can become computationally demanding as the prediction horizon grows, whereas RL-SMPC mitigates this effect by maintaining asymptotic closed-loop performance at shorter prediction horizons. Additional simulation experiments confirmed that this trend also holds across varying numbers of sampled scenarios (S) (Appendix B.1).

Because generating RL policy rollout trajectories is computationally inexpensive, coupling RL with SMPC in the RL-SMPC optimization framework did not substantially increase the computational cost. In fact, at a six-hour prediction horizon, RL-SMPC was even slightly more computationally efficient than SMPC ($\sim 5.6\%$ faster). This improvement may result from the RL-generated terminal region constraints and feedback parameterization, which likely guide the optimization toward better solutions. These computational results demonstrate that RL-SMPC can maintain high performance under parametric model uncertainty while requiring fewer computational resources than SMPC. Maintaining performance with reduced computation time is particularly valuable when real-time execution is critical for greenhouse control. Although all reported computation times remain well within the limits of physical greenhouse operations with control intervals of several minutes, the computational advantage of RL-SMPC could become increasingly important when moving to larger-scale greenhouse systems.

The greenhouse lettuce model employed in this work is relatively small and does not require extended prediction horizons to capture system dynamics over multiple days, making scenario-based SMPC computationally tractable in this case. However, for larger-scale greenhouse systems with more complex crops and additional controllable inputs, including lighting and shading screens, the optimization problem can quickly become intractable. In such scenarios, RL-SMPC could offer a computationally viable alternative that achieves improved closed-loop performance with lower computational cost.

5.3. Effect of uncertainty levels and prediction horizon

Fig. 6 illustrates the superior performance of RL-SMPC over MPC for all tested uncertainty levels and prediction horizons. This demonstrates that controller performance can degrade significantly if uncertainties are not accounted for. The performance margin between the two controllers widens as the horizon length increases.

Similarly, RL-SMPC demonstrated greater improvement over RL as the horizon increased. At a one-hour horizon and for $\delta \leq 0.1$, RL performed slightly better. Two effects might have contributed to this result. First, RL focuses on economic gain while tolerating more constraint violations, which the SMPC component then corrects in the combined RL-SMPC controller. For this short horizon, the immediate correction of these violations by the SMPC component can decrease the economic performance of RL-SMPC, giving RL a slight advancement. Second, RL-SMPC used scenario-based approximation to estimate the expected cost using only ten scenarios, while RL was trained on far more data than RL-SMPC. Therefore, for short horizons, RL-SMPC might have a worse estimate of the expected cost in comparison to RL, possibly decreasing closed-loop performance. At long horizons, this is less relevant since RL-SMPC's estimates of the long-term performance become better overall.

Across all uncertainty levels, RL-SMPC consistently outperformed SMPC for prediction horizons shorter than six hours. Given the significant computational cost of SMPC at long prediction horizons, these results indicate that RL-SMPC can maintain high performance under parametric model uncertainty with fewer computational resources. Maintaining performance while reducing computational cost is particularly important when execution time becomes a critical factor. This occurs when more complex greenhouse models are employed.

5.4. Ablation study

The ablation studies showed that removing future weather disturbances from the RL observation space decreased RL performance by 9.3% (Fig. 7). Without foresight of future weather disturbances, the RL policy prioritizes economic performance less, as reflected by improved cumulative penalty. However, this large decrease in performance was not observed when using that RL policy in RL-SMPC. Without future weather disturbances, the algorithm adhered more closely to output constraints while maintaining its economic performance, which may explain this observation.

Structural ablations revealed several insights into the contribution of algorithmic components to RL-SMPC performance (Fig. 7). First, the terminal region constraint X_N was the most significant contribution to RL-SMPC's performance. Without this constraint, RL-SMPC's behavior, which is defined by the RL policy, was similar to SMPC. This constraint guides RL-SMPC to states that improve long-term performance, for instance, by increasing the indoor CO_2 concentration (Figs. 3 & 4). Next, removing the non-linear feedback policy $\hat{u}$ from (20) resulted in lower asymptotic performance, which was caused by the solver failing to converge at longer horizons. Finally, the algorithm performed identically without the terminal cost function $\tilde{J}_\phi$. This invariance likely arises because the gradient of the cost function with respect to the terminal state was relatively small compared to the cumulative cost estimated from the open-loop trajectory. Consequently, selecting different terminal states produced little change in the expected cost estimate and had minimal effect on closed-loop performance.

6. Conclusion

Parametric model uncertainty can severely decrease the control performance of greenhouse crop production systems when not accounted for. Methods that explicitly handle uncertainty are either computationally demanding (scenario-based SMPC) or struggle to enforce constraints (RL). Therefore, this work integrated RL with SMPC (RL-SMPC), a finite-horizon optimization algorithm. The algorithm integrates RL with SMPC for system control under uncertainties. The framework used an RL-policy to incorporate a terminal region, feedback policy, and terminal cost function into an SMPC optimization formulation. The algorithm's efficacy was demonstrated on a lettuce-greenhouse model subject to parametric uncertainty. The source code required to reproduce the results is publicly available at https://github.com/BartvLaatum/RL-SMPC. Trained models and data for visualizations are available upon request.

The simulation experiments showed that RL-SMPC outperformed MPC across the eight evaluated prediction horizons. For horizons up to four hours, RL-SMPC achieved significantly higher cumulative rewards than SMPC, while using comparable computational cost. At longer horizons, both RL-SMPC and SMPC converged to similar asymptotic performance. These performance gains remained consistent across all eight evaluated uncertainty levels. Results of open-loop and closed-loop solutions illustrated RL's capability to transfer learned long-term information to the finite-horizon optimization framework within RL-SMPC. The ablation study indicated that this transfer predominantly occurred through the terminal region constraint provided by the RL policy. Moreover, this RL-SMPC algorithm is expected to be particularly effective in applications, like greenhouse control, for which steady-state targets are either not available or not desirable for the process, thereby precluding standard methods for terminal cost/constraint design.

This work assumed perfectly accurate state estimates and actual realizations of future weather and market prices. Moreover, harvest timing can influence auction prices more than dry weight accumulation. Therefore, future work could adopt market-price models that vary with seasonal factors and weight per lettuce head. Finally, because the used greenhouse model is relatively small and simple, it does not capture the system's full complexity. Subsequent work should evaluate how RL-SMPC scales to more complex, large-scale greenhouse models and accommodates additional uncertainty sources such as state-estimation and forecast errors.

CRediT authorship contribution statement

Bart van Laatum: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Visualization, Project administration; **Salim Msaad:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing; **Eldert J. van Henten:** Supervision; **Robert D. Mcallister:** Conceptualization, Methodology, Writing – review & editing, Supervision; **Sjoerd Boersma:** Conceptualization, Methodology, Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded by the Sector Plan for Science and Technology (project number 3183019014), established by the Dutch Ministry of Education, Culture and Science.

Appendix A. Nonlinear lettuce greenhouse model

The lettuce greenhouse model is defined as:

$$f(x(t), u(t), d(t), p) = x(t) + \Delta t \frac{dx}{dt_c}$$

where

$$\begin{aligned}
\frac{dx_{\mathrm{DW}}(t_c)}{dt_c} &= p_{\alpha\beta}\,\phi_{\mathrm{phot,c}}(t_c) - p_{\mathrm{resp,d}}\,x_{\mathrm{DW}}(t_c)\,2^{x_{\mathrm{H_2O}}(t_c)/10-\frac{5}{2}},\\
\frac{dx_{\mathrm{CO_2}}(t_c)}{dt_c} &= \frac{1}{p_{\mathrm{cap,c}}}\Big(-\phi_{\mathrm{phot,c}}(t_c) + p_{\mathrm{resp,c}}\,x_{\mathrm{DW}}(t_c)\,2^{x_{\mathrm{H_2O}}(t)/10-\frac{5}{2}}\\
&\quad \cdots + u_{\mathrm{CO_2}}(t_c)\,10^{-6} - \phi_{\mathrm{vent,c}}(t_c)\Big),\\
\frac{dx_{\mathrm{T}}(t_c)}{dt_c} &= \frac{1}{p_{\mathrm{cap,q}}}\Big(u_{\mathrm{heat}}(t_c) - \big(p_{\mathrm{cap,v}}\,u_{\mathrm{vent}}(t_c)\,10^{-3} + p_{\mathrm{cov},\tau}\big)\\
&\quad \cdots \big(x_{\mathrm{T}}(t_c) - d_{\mathrm{T}}(t_c)\big) + p_{\mathrm{cov,rad}}\,d_{\mathrm{iGlob}}(t_c)\Big),\\
\frac{dx_{\mathrm{H_2O}}(t_c)}{dt_c} &= \frac{1}{p_{\mathrm{cap,h}}}\big(\phi_{\mathrm{transp,h}}(t_c) - \phi_{\mathrm{vent,h}}(t_c)\big),
\end{aligned} \tag{A.1}$$

with

$$\begin{aligned}
\phi_{\mathrm{phot,c}}(t_c) &= \big(1 - e^{-p_{\mathrm{LAI,d}}x_{\mathrm{DW}}(t_c)}\big)\\
&\quad \cdots p^{\mathrm{I}}_{\mathrm{phot}}\,d_{\mathrm{iGlob}}(t_c)\\
&\quad \cdots \big(-p^{\mathrm{phot}}_{\mathrm{CO_2},1}\,x_{\mathrm{T}}(t_c)^2 + p^{\mathrm{phot}}_{\mathrm{CO_2},2}\,x_{\mathrm{T}}(t_c) - p^{\mathrm{phot}}_{\mathrm{CO_2},3}\big)\\
&\quad \cdots \big(x_{\mathrm{CO_2}}(t_c) - p_8\big)/\varphi(t_c),\\
\varphi(t_c) &= p^{\mathrm{I}}_{\mathrm{phot}}\,d_{\mathrm{iGlob}}(t_c) +\\
&\quad \cdots \big(-p^{\mathrm{phot}}_{\mathrm{CO_2},1}\,x_{\mathrm{T}}(t_c)^2 + p^{\mathrm{phot}}_{\mathrm{CO_2},2}\,x_{\mathrm{T}}(t_c) - p^{\mathrm{phot}}_{\mathrm{CO_2},3}\big)\\
&\quad \cdots \big(x_{\mathrm{CO_2}}(t_c) - p_8\big),\\
\phi_{\mathrm{vent,c}}(t_c) &= \big(u_{\mathrm{vent}}(t_c)\,10^{-3} + p_{\mathrm{leak}}\big)\,\big(x_{\mathrm{CO_2}}(t_c) - d_{\mathrm{CO_2}}(t_c)\big),\\
\phi_{\mathrm{vent,h}}(t_c) &= \big(u_{\mathrm{vent}}(t_c)\,10^{-3} + p_{\mathrm{leak}}\big)\,\big(x_{\mathrm{H_2O}}(t_c) - d_{\mathrm{H_2O}}(t_c)\big),\\
\phi_{\mathrm{transp,h}}(t_c) &= p_{\mathrm{vap}}\,\big(1 - e^{-p_{\mathrm{LAI,d}}x_{\mathrm{DW}}(t_c)}\big)\\
&\quad \cdots \Big(\frac{p_{\mathrm{sat,H_2O},1}}{p_{\mathrm{R}}\big(x_{\mathrm{T}}(t_c) + p_{\mathrm{T}}\big)}\,e\big(\frac{p_{\mathrm{sat,H_2O},2}\,x_{\mathrm{T}}(t_c)}{x_{\mathrm{T}}(t_c) + p_{\mathrm{sat,H_2O},3}}\big) - x_{\mathrm{H_2O}}(t_c)\Big).
\end{aligned} \tag{A.2}$$

Where $t_c \in \mathbb{R}$ represents the continuous time and $\phi_{\mathrm{phot,c}}$, $\phi_{\mathrm{vent,c}}$, $\phi_{\mathrm{vent,h}}$ and $\phi_{\mathrm{transp,h}}$ are the gross canopy photosynthesis rate, mass exchange of CO_2 through the vents, mass exchange of H_2O through the vents, and canopy transpiration. The model is discretized using the explicit fourth-order Runge-Kutta method, resulting in the discrete-time model as presented in (1).

The system output function is defined as:

$$\begin{aligned}
y(t) &= h\big(x(t)\big),\\
h\big(x(t)\big) &= \begin{bmatrix} h_1\big(x(t)\big)\\ h_2\big(x(t)\big)\\ h_3\big(x(t)\big)\\ h_4\big(x(t)\big)\big)\end{bmatrix},
\end{aligned} \tag{A.3}$$

with:

$$\begin{aligned}
h_1(x(t)) &= x_{\mathrm{DW}}(t),\\
h_2(x(t)) &= \frac{10^6 c_{\mathrm{R}}\big(x_{\mathrm{T}}(t) + c_{\mathrm{K}}\big)}{c_{\mathrm{p}}\,c_{\mathrm{M}}}\,x_{\mathrm{CO_2}}(t),\\
h_3(x(t)) &= x_{\mathrm{T}},\\
h_4(x(t)) &= \frac{c_{\mathrm{R}}\big(x_{\mathrm{T}}(t) + c_{\mathrm{K}}\big)}{e^{\frac{c_{\mathrm{sat,H_2O},4}\cdot x_{\mathrm{T}}(t)}{x_{\mathrm{T}}(t) + c_{\mathrm{sat,H_2O},5}}}}\,x_{\mathrm{H_2O}}(t).
\end{aligned} \tag{A.4}$$

The nominal model parameters ($\bar{p}$) are chosen according to Boersma et al. (2022), Van Henten (1994) and are given in Table A.1. To not

Table A.1
The nominal model parameter values ($\bar{p}$) used during simulation.

Variable	Value	Unit	Description
$p_{\alpha\beta}$	0.544	–	Yield factor
$p_{\text{cap,c}}$	4.1	m^3/m^2	Vol. CO_2 capacity of indoor air
$p_{\text{cap,q}}$	3×10^4	$J/m^2/°C$	Effective heat capacity of indoor air
$p_{\text{cap,h}}$	4.1	m^3/m^2	Vol. humidity capacity of indoor air
$p_{\text{LAI,d}}$	53	m^2/kg	Effective canopy surface
p^{f}_{phot}	3.55×10^{-9}	kg/J	Light-use efficiency
$p^{\text{phot}}_{\text{CO}_2,1}$	5.11×10^{-6}	$m/s/°C^2$	Temp. factor (1) on gross canopy photosynthesis
$p^{\text{phot}}_{\text{CO}_2,2}$	2.30×10^{-4}	m/s/°C	Temp. factor (2) on gross canopy photosynthesis
$p^{\text{phot}}_{\text{CO}_2,3}$	6.29×10^{-4}	m/s	Temp. factor (3) on gross canopy photosynthesis
$p_{\text{resp,d}}$	2.65×10^{-7}	s^{-1}	Respiration rate of crop's dry matter
p_{phot}	5.2×10^{-5}	kg/m^3	CO_2 compensation point
$p_{\text{resp,c}}$	4.87×10^{-7}	s^{-1}	CO_2 release-rate factor from respiration
p_{leak}	7.5×10^{-6}	m/s	Greenhouse-cover ventilation leakage
$p_{\text{cap,v}}$	1290	$J/m^3/°C$	Air heat capacity per volume
$p_{\text{cov,r}}$	6.1	$W/m^2/°C$	Heat-transmission factor through cover
$p^{\text{cov}}_{\text{rad}}$	0.2	–	Solar heat-load coefficient
p_{vap}	3.6×10^{-3}	m/s	Vapour mass-transfer factor (leaf-air)
$p_{\text{sat,H}_2\text{O},1}$	9348	J/m^3	Saturation-pressure poly. coefficient 1
$p_{\text{sat,H}_2\text{O},2}$	17.4	–	Saturation-pressure poly. coefficient 2
$p_{\text{sat,H}_2\text{O},3}$	239	°C	Saturation-pressure poly. coefficient 3
p_{R}	8314	J/K/kmol	Universal gas constant
p_{T}	273.15	K	Kelvin-Celsius conversion offset

Table A.2
The coefficient values for the measurement function (A.4) used during simulation.

Variable	Value	Unit	Description
c_{R}	8.3144598	$J\,mol^{-1}\,K^{-1}$	molar gas constant
c_{K}	273.15	K	Conversion from °C to K
c_{p}	101325	Pa	Atmospheric pressure
c_{M}	44.01×10^{-3}	$kg\,mol^{-1}$	molar mass of CO_2
$c_{\text{sat,H}_2\text{O},4}$	610.48	-	Saturation-pressure poly. coefficient 4
$c_{\text{sat,H}_2\text{O},5}$	17.2694	-	Saturation-pressure poly. coefficient 5

introduce noise in only half of the output variables, the coefficients of the measurement function are fixed; their values are listed in Table A.2.

Appendix B. Additional simulation results

B.1. Scenario sample size

This section examines the sensitivity of RL-SMPC to the number of sampled scenarios (S) by comparing its average computational cost and closed-loop performance with those of SMPC. In this additional simulation experiment, both controllers, as presented in Sections 3.3 and 3.4, approximated the optimal control problem (10) using a varying number of sampled scenarios (S). Specifically, for each method, the following values were tested: $S = [5, 10, 15, 20]$ with prediction horizons $H = [1, 2, 3, 4]$. For each setting, both controllers were evaluated across ten independent simulation runs with unique random seeds. The results demonstrate that, regardless of the number of sampled scenarios, RL-SMPC outperforms SMPC in terms of cumulative reward, see Table B.2, while using similar computational cost per time step, see Table B.1. These findings are consistent with those reported in Section 4.2, indicating that RL-SMPC is not more sensitive to the number of scenarios than SMPC.

B.2. Robustness RL-SMPC against RL random seed

Training RL policies is well known to be sensitive to the choice of the random seed. This seed controls several stochastic processes during training, including the initialization of the neural network parameters, the sampling of actions from the stochastic policy, and the sampling of batches from the replay buffer for computing gradients in the stochastic gradient descent algorithm. The results presented in Section 4 are based on a single RL policy trained with one random seed. To assess the robustness of RL-SMPC to variation in the RL training seed, this simulation experiment examines the variance in performance of standalone RL policies and when paired with RL-SMPC.

Specifically, five RL policies were trained using five unique random seeds. After training, each policy was evaluated across ten simulation runs. Subsequently, each policy was paired with an RL-SMPC controller and tested for the following prediction horizons of $H = [1, 2, 3, 4]$. Finally, the mean closed-loop performance of the five policies and their corresponding RL-SMPC controllers was used to compute the average and confidence interval across random seeds.

The results show some variance in the performance of RL policies across different random seeds. The differences between individual policies appear to result from their varying focus on either economic performance or state constraint satisfaction. Interestingly, this variance decreased substantially when the RL policies were paired with RL-SMPC. Moreover, increasing the prediction horizons further decreased the variance across random seeds, as RL-SMPC becomes less dependent on the performance of the underlying policy. These findings demonstrate the robustness of RL-SMPC to the random seed used during RL policy training Fig. B.1

Table B.1
The computational cost per time step per number of sampled scenarios (S) and prediction horizon length (H). for both SMPC and RL-SMPC in seconds. The computational costs are averaged across ten simulations with reported standard deviation.

Scenarios (S)	Controller	Horizon (H)			
		1	2	3	4
5	SMPC	0.10 (0.0005)	0.22 (0.0011)	0.34 (0.0012)	0.47 (0.0032)
	RL-SMPC	0.11 (0.0002)	0.24 (0.0014)	0.37 (0.0022)	0.51 (0.0011)
10	SMPC	0.21 (0.0006)	0.46 (0.0022)	0.73 (0.0022)	1.02 (0.0060)
	RL-SMPC	0.22 (0.0006)	0.49 (0.0015)	0.78 (0.0024)	1.10 (0.0058)
15	SMPC	0.34 (0.0026)	0.75 (0.0114)	1.19 (0.0072)	1.71 (0.0265)
	RL-SMPC	0.33 (0.0009)	0.74 (0.0022)	1.22 (0.0029)	1.75 (0.0048)
20	SMPC	0.46 (0.0043)	1.05 (0.0166)	1.70 (0.0367)	2.33 (0.0165)
	RL-SMPC	0.45 (0.0006)	1.00 (0.0023)	1.65 (0.0030)	2.40 (0.0139)

Table B.2
The cumulative reward (27) for SMPC and RL-SMPC per number of sampled scenarios (S) and prediction horizon length (H), with reported one standard deviation. The cumulative reward is averaged across ten simulation runs.

Scenarios (S)	Controller	Horizon (H)			
		1	2	3	4
5	SMPC	2.18 (0.04)	2.43 (0.04)	2.59 (0.05)	2.71 (0.04)
	RL-SMPC	2.47 (0.03)	2.64 (0.04)	2.74 (0.05)	2.79 (0.04)
10	SMPC	2.25 (0.04)	2.49 (0.04)	2.64 (0.04)	2.76 (0.04)
	RL-SMPC	2.54 (0.04)	2.70 (0.04)	2.79 (0.04)	2.85 (0.04)
15	SMPC	2.28 (0.04)	2.51 (0.04)	2.66 (0.03)	2.77 (0.03)
	RL-SMPC	2.55 (0.04)	2.72 (0.04)	2.81 (0.04)	2.87 (0.04)
20	SMPC	2.29 (0.04)	2.52 (0.03)	2.68 (0.04)	2.79 (0.04)
	RL-SMPC	2.57 (0.04)	2.74 (0.03)	2.82 (0.04)	2.88 (0.04)

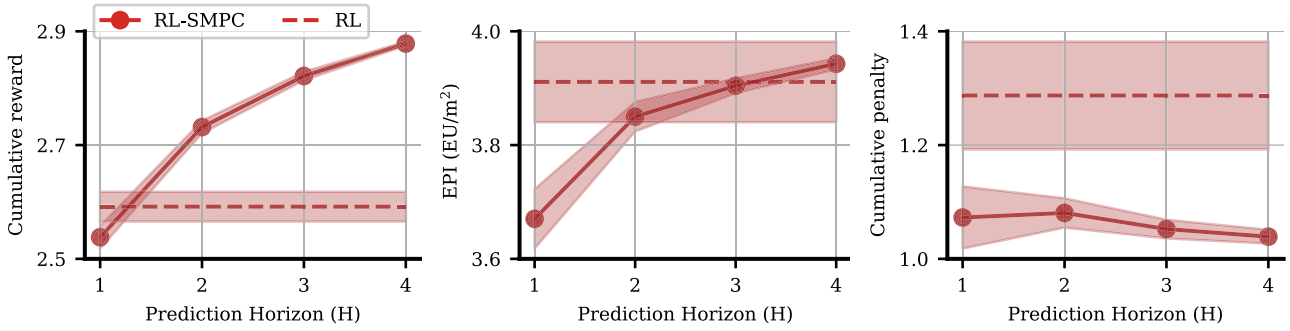


Fig. B.1. Performance variability of RL and RL-SMPC across five random seeds. This figure visualizes the mean cumulative sum of three performance metrics averaged across five random seeds for RL and RL-SMPC. The shaded area represents the 95% confidence interval. Simulations were executed under parametric uncertainty with $\delta = 0.1$. **Left:** The cumulative reward as defined in (27). **Middle:** The EPI as defined in (28). **Right:** The cumulative sum of the linear penalty function as defined (29). Note that higher penalty values correspond to more state violations.

References

Andersson, J., Gillis, J., Horn, G., Rawlings, J. B., & Diehl, M. (2019). CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, *11*(1), 1–36. https://doi.org/10.1007/s12532-018-0139-4

Bertsekas, D. P. (2024). Model predictive control and reinforcement learning: a unified framework based on dynamic programming. *IFAC-PapersOnLine*, *58*(18), 363–383. https://doi.org/10.1016/j.ifacol.2024.09.056

Blasco, X., Martínez, M., Herrero, J. M., Ramos, C., & Sanchis, J., et al. (2007). Model-based predictive control of greenhouse climate for reducing energy and water consumption. *Computers and Electronics in Agriculture*, *55*(1), 49–70. https://doi.org/10.1016/j.compag.2006.12.001

Boersma, S., Sun, C., & Van Mourik, S., et al. (2022). Robust sample-based model predictive control of a greenhouse system with parametric uncertainty. *IFAC-PapersOnLine*, *55*(32), 177–182. https://doi.org/10.1016/j.ifacol.2022.11.135

Boersma, S., Van Mourik, S., Xin, B., Kootstra, G., & Bustos-Korts, D. (2022). Nonlinear observability analysis and joint state and parameter estimation in a lettuce greenhouse using ensemble kalman filtering. *IFAC-PapersOnLine*, *55*(32), 141–146.

Brunke, L., Greeff, M., Hall, A. W., Yuan, Z., Zhou, S., Panerati, J., & Schoellig, A. P., et al. (2022). Safe learning in robotics: from learning-Based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, *5*(Volume 5, 2022), 411–444. https://doi.org/10.1146/annurev-control-042920-020211

Cannon, M., Kouvaritakis, B., Rakovic, S. V., & Cheng, Q., et al. (2011). Stochastic tubes in model predictive control with probabilistic constraints. *IEEE Transactions on Automatic Control*, *56*(1), 194–200. https://doi.org/10.1109/tac.2010.2086553

Chatterjee, D., & Lygeros, J. (2015). On stability and performance of stochastic predictive control techniques. *IEEE Transactions on Automatic Control*, *60*(2), 509–514. https://doi.org/10.1109/TAC.2014.2335274

Chaudhary, G., Kaur, S., Mehta, B., & Tewani, R., et al. (2019). Observer based fuzzy and PID controlled smart greenhouse. *Journal of Statistics and Management Systems*, *22*(2), 393–401. https://doi.org/10.1080/09720510.2019.1582880

Chen, W.-H., & You, F. (2020). Data-driven robust optimization for greenhouse temperature control using model predictive control. *Chemical Engineering Transactions*, *81*, 721–726. https://doi.org/10.3303/CET2081121

Chen, Z., Hu, H., Wu, Y., Zhang, Y., Li, G., & Liu, Y. (2020). Stochastic model predictive control for energy management of power-split plug-in hybrid electric vehicles based on reinforcement learning. *Energy*, *211*, 118931.

Ding, Y., Wang, L., Li, Y., & Li, D., et al. (2018). Model predictive control and its application in agriculture: a review. *Computers and Electronics in Agriculture*, *151*, 104–117. https://doi.org/10.1016/j.compag.2018.06.004

Fagiano, L., & Teel, A. R. (2013). Generalized terminal state constraint for model predictive control. *Automatica*, *49*(9), 2622–2631. https://doi.org/10.1016/j.automatica.2013.05.019

García-Mañas, F., Rodríguez, F., Berenguel, M., & Maestre, J. M., et al. (2024). Multi-scenario model predictive control for greenhouse crop production considering

market price uncertainty. *IEEE Transactions on Automation Science and Engineering*, *21*(3), 2936–2948. https://doi.org/10.1109/TASE.2023.3271896

González, R., Rodríguez, F., Guzmán, J. L., & Berenguel, M. (2014). Robust constrained economic receding horizon control applied to the two time-scale dynamics problem of a greenhouse. *Optimal Control Applications and Methods*, *35*(4), 435–453. https://doi.org/10.1002/oca.2080

Gruber, J. K., Guzmán, J. L., Rodríguez, F., Bordons, C., Berenguel, M., & Sánchez, J. A., et al. (2011). Nonlinear MPC based on a volterra series model for greenhouse temperature control using natural ventilation. *Control Engineering Practice*, *19*(4), 354–366. https://doi.org/10.1016/j.conengprac.2010.12.004

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S., et al. (2018). Soft actor-Critic: off-Policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th international conference on machine learning* (pp. 1861–1870). PMLR.

Hamza, A., Ramdani, M., & Bougheloum, W. (2019). Robust T-S fuzzy constrained predictive control design for greenhouse micro-climate. *International Journal of Scientific Research & Engineering Technology*, *13*, 1–4.

Ito, K. (2012). Greenhouse temperature control with wooden pellet heater via model predictive control approach. In *2012 20Th mediterranean conference on control & automation (MED)* (pp. 1542–1547). https://doi.org/10.1109/MED.2012.6265858

Katzin, D., Van Mourik, S., Kempkes, F., & Van Henten, E. J., et al. (2020). Greenlight – an open source model for greenhouses with supplemental lighting: evaluation of heat requirements under LED and HPS lamps. *Biosystems Engineering*, *194*, 61–81. https://doi.org/10.1016/j.biosystemseng.2020.03.010

Kempkes, F. L. K., Janse, J., & Hemming, S. (2014). Greenhouse concept with high insulating double glass with coatings and new climate control strategies; from design to results from tomato experiments. *Acta Horticulturae*, (1037), 83–92. https://doi.org/10.17660/ActaHortic.2014.1037.6

Kuijpers, W. J. P., Antunes, D. J., Van Mourik, S., Van Henten, E. J., & Van De Molengraft, M. J. G., et al. (2022). Weather forecast error modelling and performance analysis of automatic greenhouse climate control. *Biosystems Engineering*, *214*, 207–229. https://doi.org/10.1016/j.biosystemseng.2021.12.014

Kuijpers, W. J. P., Katzin, D., Van Mourik, S., Antunes, D. J., Hemming, S., & Van De Molengraft, M. J. G., et al. (2021). Lighting systems and strategies compared in an optimally controlled greenhouse. *Biosystems Engineering*, *202*, 195–216. https://doi.org/10.1016/j.biosystemseng.2020.12.006

Lafont, F., Balmat, J.-F., Pessel, N., & Fliess, M., et al. (2015). A model-free control strategy for an experimental greenhouse with an application to fault accommodation. *Computers and Electronics in Agriculture*, *110*, 139–149. https://doi.org/10.1016/j.compag.2014.11.008

Lorenzen, M., Dabbene, F., Tempo, R., & Allgower, F., et al. (2017). Constraint-tightening and stability in stochastic model predictive control. *IEEE Transactions on Automatic Control*, *62*(7), 3165–3177. https://doi.org/10.1109/tac.2016.2625048

Mallick, S., Airaldi, F., Dabiri, A., Sun, C., & De Schutter, B., et al. (2025). Reinforcement learning-based model predictive control for greenhouse climate control. *Smart Agricultural Technology*, *10*, 100751. https://doi.org/10.1016/j.atech.2024.100751

Mayne, D. Q., Kerrigan, E. C., van Wyk, E. J., & Falugi, P. (2011). Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, *21*(11), 1341–1353. https://doi.org/10.1002/rnc.1758

Mesbah, A. (2016). Stochastic model predictive control: an overview and perspectives for future research. *IEEE Control Systems Magazine*, *36*(6), 30–44. https://doi.org/10.1109/MCS.2016.2602087

Messerer, F., & Diehl, M. (2021). An efficient algorithm for tube-based robust nonlinear optimal control with optimal linear feedback. In *2021 60Th IEEE conference on decision and control (CDC)* (pp. 6714–6721). Austin, TX, USA: IEEE. https://doi.org/10.1109/CDC45484.2021.9683712

Mondaca-Duarte, F. D., van Mourik, S., Balendonck, J., Voogt, W., Heinen, M., & van Henten, E. J., et al. (2020). Irrigation, crop stress and drainage reduction under uncertainty: a scenario study. *Agricultural Water Management*, *230*, 105990. https://doi.org/10.1016/j.agwat.2019.105990

Montoya, A. P., Guzmán, J. L., Rodríguez, F., & Sánchez-Molina, J. A., et al. (2016). A hybrid-controlled approach for maintaining nocturnal greenhouse temperature: simulation study. *Computers and Electronics in Agriculture*, *123*, 116–124. https://doi.org/10.1016/j.compag.2016.02.014

Morcego, B., Yin, W., Boersma, S., Van Henten, E., Puig, V., & Sun, C., et al. (2023). Reinforcement learning versus model predictive control on greenhouse climate control. *Computers and Electronics in Agriculture*, *215*, 108372. https://doi.org/10.1016/j.compag.2023.108372

Msaad, S., Harraway, M., & McAllister, R. D. (2025). RL-Guided MPC For autonomous greenhouse control. *IFAC-PapersOnLine*, *59*(23), 449–454. https://doi.org/10.1016/j.ifacol.2025.11.829

MÃ¼ller, M. A., Angeli, D., & AllgÃ¶wer, F. (2014). On the performance of economic model predictive control with self-tuning terminal cost. *Journal of Process Control*, *24*(8), 1179–1186. https://doi.org/10.1016/j.jprocont.2014.05.009

Piñón, S., Peña, M., Camacho, E. F., & Kuchen, B., et al. (2001). Robust predictive control for a greenhouse using input/output linearization and linear matrix inequalities. *IFAC Proceedings Volumes*, *34*(29), 82–87. https://doi.org/10.1016/S1474-6670(17)32797-0

Primbs, J. A., & Sung, C. H. (2009). Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise. *IEEE Transactions on Automatic Control*, *54*(2), 221–230. https://doi.org/10.1109/tac.2008.2010886

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N., et al. (2021). Stable-baselines3: reliable reinforcement learning implementations. *Journal of Machine Learning Research*, *22*(268), 1–8.

Reiter, R., Ghezzi, A., BaumgÃ¤rtner, K., Hoffmann, J., McAllister, R. D., & Diehl, M. (2025). Ac4mpc: Actor-critic reinforcement learning for guiding model predictive control. *IEEE Transactions on Control Systems Technology*, (pp. 1–16). https://doi.org/10.1109/TCST.2025.3620521

Reiter, R., Hoffmann, J., Reinhardt, D., Messerer, F., Baumgärtner, K., Sawant, S., Boedecker, J., Diehl, M., & Gros, S. (2025). Synthesis of model predictive control and reinforcement learning: Survey and classification. arXiv preprint arXiv:2502.02133.

Salzmann, T., Arrizabalaga, J., Andersson, J., Pavone, M., & Ryll, M., et al. (2024). Learning for casADi: data-driven models in numerical optimization. In *Proceedings of the 6th annual learning for dynamics & control conference* (pp. 541–553). PMLR.

Salzmann, T., Kaufmann, E., Arrizabalaga, J., Pavone, M., Scaramuzza, D., & Ryll, M., et al. (2023). Real-time neural MPC: deep learning model predictive control for quadrotors and agile robotic platforms. *IEEE Robotics and Automation Letters*, *8*(4), 2397–2404. https://doi.org/10.1109/LRA.2023.3246839

Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. Cambridge, MA, USA: A Bradford Book.

Svensen, J. L., Cheng, X., Boersma, S., & Sun, C., et al. (2024). Chance-constrained stochastic MPC of greenhouse production systems with parametric uncertainty. *Computers and Electronics in Agriculture*, *217*, 108578. https://doi.org/10.1016/j.compag.2023.108578

Van Henten, E. J. (1994). Greenhouse climate management : an optimal control approach. Ph.D. thesis. Agricultural University. https://doi.org/10.18174/205106

Van Henten, E. J. (2003). Sensitivity analysis of an optimal control problem in greenhouse climate management. *Biosystems Engineering*, *85*(3), 355–364. https://doi.org/10.1016/S1537-5110(03)00068-0

Van Laatum, B., Van Henten, E. J., & Boersma, S. (2025). Greenlight-Gym: Reinforcement learning benchmark environment for control of greenhouse production systems. *IFAC-PapersOnLine*, *59*(23), 437–442. https://doi.org/10.1016/j.ifacol.2025.11.827

Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, *106*(1), 25–57. https://doi.org/10.1007/s10107-004-0559-y

Xu, D., Du, S., & Van Willigenburg, G., et al. (2018). Adaptive two time-scale receding horizon optimal control for greenhouse lettuce cultivation. *Computers and Electronics in Agriculture*, *146*, 93–103. https://doi.org/10.1016/j.compag.2018.02.001

Zarrouki, B., Wang, C., & Betz, J. (2024). Adaptive stochastic nonlinear model predictive control with look-ahead deep reinforcement learning for autonomous vehicle motion control. In *2024 IEEE/RSJ International conference on intelligent robots and systems (IROS)* (pp. 12726–12733). IEEE.