TU Delft

Delft University of Technology

Document Version
Final published version

Licence
CC BY

Citation (APA)
van Laatum, B., Msaad, S., van Henten, E. J., McAllister, R. D., & Boersma, S. (2026). Improving greenhouse fruit-production control by integrating reinforcement learning into short-horizon model predictive control. *Computers and Electronics in Agriculture*, *254*, Article 112501. https://doi.org/10.1016/j.compag.2026.112501

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

Computers and Electronics in Agriculture 254 (2026) 112501

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Original papers

Improving greenhouse fruit-production control by integrating reinforcement learning into short-horizon model predictive control

B. van Laatum [a,*], S. Msaad [b], E.J. van Henten [a], R.D. McAllister [b], S. 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

Dataset link: https://github.com/BartvLaatum/RL-MPC-GreenLight

Keywords:
Model predictive control
Reinforcement learning
Greenhouse fruit production control
Climate control
Finite-horizon optimization

ABSTRACT

Greenhouse fruit-production control aims to maximize the economic performance (fruit revenue minus operating costs) while operating within system constraints under external weather disturbances. Control methods need to balance the delayed economic benefit of fruit yield with current operating costs. For such problems, model predictive control (MPC) can handle constraints under future weather disturbances, but can become computationally demanding when using sufficiently long prediction horizons for (relatively large) nonlinear greenhouse fruit production models. In contrast, reinforcement learning (RL) can learn control policies before deployment while considering economic performance beyond MPC's prediction horizon. However, desired indoor climate ranges are typically incorporated through penalty terms, and performance may degrade under unseen weather trajectories. This work proposes *trajectory-selection RL–MPC*, a framework that incorporates the economic benefit of fruit yield beyond the MPC prediction horizon into a finite-horizon optimization problem with a relatively short horizon (one hour). The framework uses an RL rollout trajectory to define a terminal region constraint and terminal cost. Next, a nonlinear MPC solves a short-horizon optimization problem with these terminal ingredients to find a local optimum. Finally, the framework selects and executes the first input from the trajectory with the better objective value, either from the MPC-predicted or the RL rollout trajectory. The method is applied to GreenLight, a large-scale nonlinear greenhouse tomato production model that exhibits stiff dynamics. The simulation results show that trajectory-selection RL–MPC with a one-hour prediction horizon matches the closed-loop performance of a high-performing guiding policy while improving over standalone MPC with the same horizon. On 25 weather trajectories held out during policy training, trajectory-selection RL–MPC achieved ~52% higher closed-loop performance than standalone RL and ~80% higher performance than MPC with the same horizon length.

1. Introduction

Changing climate conditions and more extreme weather events threaten the stable production of fruits and vegetables in open-field farming (Silva et al., 2017; Bisbis et al., 2018; Mulders et al., 2024). Greenhouse fruit production systems can provide more stable food production by protecting crops against unfavorable external weather conditions (Baudoin et al., 2013). However, greenhouse economic performance remains strongly influenced by external weather through its effects on indoor climate, resource use, and operating costs (Fanourakis et al., 2025). As a result, greenhouse climate control is not only a climate management problem but also a long-horizon economic decision problem that must balance fruit yield and quality against operating costs over long horizons under exogenous weather disturbances. In practice, greenhouse growers need to make decisions over long production horizons while accounting for economic performance indicators, future weather conditions, and fruit growth and development dynamics (Van Straten et al., 2010). The increasing scarcity of experienced employees with horticultural knowledge further motivates the need for automated control systems that can support growers in maintaining optimal growing conditions for crops at minimal cost (Ariesen-Verschuur et al., 2022). Advanced automated control methods, such as optimal control and reinforcement learning, provide frameworks for addressing this economic control problem.

Greenhouse crop production control has been widely studied as an economic optimal control problem (Van Henten, 1994; Tap, 2000; Van Ooteghem, 2007; Van Straten et al., 2010; Ramírez-Arias et al.,

* Corresponding author.
E-mail address: hbpvanlaatum123@gmail.com (B. van Laatum).

https://doi.org/10.1016/j.compag.2026.112501
Received 11 June 2026; Received in revised form 14 September 2026; Accepted 28 September 2026
Available online 7 October 2026
0168-1699/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

2012; Kuijpers et al., 2021b; Fink et al., 2023). In this problem formulation, a controller determines a sequence of control inputs to optimize economic return over a given optimization horizon subject to system constraints under exogenous disturbances. In greenhouse crop production systems, climate dynamics have considerably faster timescales (minutes) compared to crop growth and development dynamics (weeks). For greenhouse systems producing fruit-bearing crops, such as tomatoes, immediate operating costs for climate control must be balanced against the delayed economic benefits from fruit yield. These delayed benefits involve fruit-development timescales of 40 to 60 days (De Koning, 1994; Adams et al., 2001). Ideally, to capture this trade-off, the controller's optimization horizon should correspond to the crop's slow development timescales. However, solving optimal control problems online over such horizon lengths is computationally intractable (Fink et al., 2023; Xu et al., 2025).

Previous work on tomato production control has therefore simplified these fruit-yield dynamics by counting all assimilates partitioned to the fruits as immediately marketable product, thereby reducing the required optimization horizon (Seginer et al., 2018; Kuijpers et al., 2021b). This modification shortens the delay between climate control and fruit yield in the prediction model. This reduction in delay is because the assimilates are produced through photosynthesis and subsequently partitioned from the crop buffer to the fruits based on the 24-hour average crop temperature. This work adopts this fruit-yield approximation and therefore studies a control problem, in which the delayed response time from fruit development is not represented. However, the numerical solution of an optimization problem using a greenhouse tomato production model still remains challenging. GreenLight (Katzin et al., 2020), for instance, is a large-scale nonlinear tomato production model that integrates climate and crop processes with large differences in response times, resulting in stiff dynamics. Consequently, finite-horizon optimization with GreenLight remains computationally demanding even after adopting the fruit-yield approximation formulation. This motivates the use of approximate numerical control methods for the greenhouse fruit production control problem considered in this work.

Model predictive control (MPC) is a specific optimal control method that approximates the long-horizon economic optimal control problem by optimizing over a finite prediction horizon. At each time step, MPC optimizes the future input trajectory from the current state, applies the first optimized control input, and repeats this procedure at the next time step in a receding-horizon manner. Since the optimization problem is solved online, MPC can explicitly account for predicted future disturbances such as external weather. However, when MPC employs large-scale nonlinear prediction models rather than typical control-oriented greenhouse models, prediction horizons that capture the relevant crop growth timescales can become computationally infeasible. GreenLight (Katzin et al., 2020), for instance, uses 27 states to model the greenhouse system, whereas the control-oriented model in Kuijpers et al. (2021b) uses eight states. As a result, practical MPC implementations for greenhouse crop production using large-scale models may rely on relatively short prediction horizons. Consequently, the effects of current control inputs on crop growth and economic performance beyond such short horizons are not represented in the optimization problem. When the timescales of fruit-yield dynamics are substantially longer than MPC's prediction horizon, this may result in myopic controllers. This challenge motivates control approaches that enhance the economic performance of MPC controllers with prediction horizons that do not span the full production cycle.

One line of research proposes a timescale decomposition of the optimal control problem (Van Henten, 1994; Tap, 2000; Van Henten and Bontsema, 2009; Van Straten and Van Henten, 2010; Xu et al., 2018). This approach first solves the optimization problem over the entire growing season for slow crop states, resulting in an open-loop trajectory for the crop state and co-state. Next, this crop state and co-state serve as economic performance indicators in a finite-horizon optimization problem with short prediction horizons (order of hours). However, extending this timescale decomposition to large, nonlinear, and stiff greenhouse models that account for delayed fruit development can become computationally challenging. More recently, Kuijpers et al. (2021a) addressed these differing timescales by modifying the underlying fruit development model that captures the delayed benefits from fruit harvest (order of weeks). This modification reduced the prediction horizon of a receding horizon controller to three days. However, this method relies on the assumption that the crop has reached a fully generative state, making it inapplicable for optimizing the initial months of the generative phase.

Reinforcement learning (RL) provides an alternative approach for approximating a controller with economic performance over longer horizons for greenhouse production systems. RL learns a control policy (i.e., controller) from numerous system interactions during the training phase (Sutton and Barto, 2018). In greenhouse applications, this training is performed offline using a simulation model, which shifts computational effort away from online deployment (Morcego et al., 2023; Van Laatum et al., 2025). Therefore, RL can be attractive when longer-term economic performance is important, but solving the full optimal control problem online is computationally intractable. However, unlike MPC, RL does not explicitly enforce system constraints. Instead, these constraints are typically incorporated through penalty terms in the objective function. This approach requires careful tuning to balance economic performance against constraint satisfaction (Morcego et al., 2023; Van Laatum et al., 2025). A second limitation is that RL policies may generalize poorly to circumstances not encountered during training (A. Zhang et al., 2018; C. Zhang et al., 2018; Kirk et al., 2023). In greenhouse control settings, RL policies may overfit to the disturbance distributions encountered during training, for instance, a narrow set of weather trajectories, which can lead to decreased performance under unseen weather conditions. While training RL policies on broader distributions of historical weather trajectories can improve overall performance, it does not completely prevent degraded performance under unseen weather conditions (C. Zhang et al., 2018; Kirk et al., 2023; Mediratta et al., 2024). These limitations motivate the combination of RL with a complementary control method, such as MPC. In such a framework, the RL policy can provide a learned approximation of economic performance beyond MPC's finite prediction horizon. At the same time, the MPC optimization can refine RL's control input trajectory over a finite horizon using weather predictions.

These complementary strengths of MPC and RL have motivated their integration in greenhouse crop production control studies (Msaad et al., 2025; Mallick et al., 2025; Van Laatum et al., 2026). Previous greenhouse production control studies have combined these methods in two main ways. One line of work used a parameterized MPC controller as an RL policy and cost function approximators (Mallick et al., 2025). By using RL to learn the parameterization of the MPC framework, the controller improved performance relative to standalone MPC and RL without substantially increasing computation. Another line of work used a trained RL policy to guide the MPC optimization (Msaad et al., 2025; Van Laatum et al., 2026). The RL policy provided terminal constraints and terminal costs to the optimization problem through a policy rollout. This policy-guided MPC approach achieved performance comparable to MPC while reducing computation time (Msaad et al., 2025; Van Laatum et al., 2026). Beyond greenhouse control, similar ideas have been explored in the general context of nonlinear control. Ghezzi et al. (2026) initialized a real-time, iteration-based MPC with an RL rollout and refined it through a single Riccati-based Newton step. They proved that the suboptimality of the RL rollout is reduced quadratically by one Newton step. The authors of Reiter et al. (2025) introduced a parallel architecture in which the MPC trajectory and the RL rollout are evaluated at each time step using a learned critic, and the lower-cost trajectory is applied to the system. These results demonstrate the promise of combining RL and MPC in a unified framework. However, existing greenhouse production control studies

have tested such controllers on relatively small (i.e., four-state) lettuce greenhouse models and under limited weather variation. Therefore, it remains unclear whether policy-guided MPC approaches scale to large, nonlinear, and stiff fruit-production greenhouse models, and whether MPC optimization further improves performance relative to the guiding RL policy under weather trajectories not encountered during policy training.

To address these gaps, this work introduces *trajectory-selection RL–MPC* for greenhouse fruit production control. The proposed framework extends the policy-guided MPC approach with an online trajectory-selection mechanism. Following earlier policy-guided MPC work, the finite-horizon optimization problem incorporates terminal region constraints and terminal costs provided by a learned RL policy via rollout trajectories (Msaad et al., 2025). The terminal region constraints and terminal costs provide a mechanism for incorporating economic information learned by the RL policy beyond the MPC prediction horizon into the finite-horizon optimization problem. Moreover, the MPC optimization locally refines the suggested RL rollout trajectories by optimizing over a finite prediction horizon using short-term weather predictions, with the aim of improving performance over the standalone RL policy in weather conditions not encountered during training. In addition to the existing framework, trajectory-selection RL–MPC evaluates the objective for both the MPC-predicted trajectory and the rollout trajectory over the prediction horizon, selects the trajectory with the best value, and applies the first input to the system. Unlike the parallel architecture of Reiter et al. (2025), which evaluates trajectories using a learned critic and a policy rollout, the proposed selection mechanism evaluates both candidate trajectories directly using the finite-horizon planning objective. This selection mechanism is designed to avoid a decrease in performance relative to the rollout when policy-guided MPC optimization does not improve this objective.

Trajectory-selection RL–MPC is evaluated using GreenLight, a large-scale nonlinear greenhouse tomato model with 27 states and stiff dynamics (Katzin et al., 2020). In contrast to the smaller control-oriented greenhouse models commonly used in previous optimal control studies, GreenLight poses a computationally demanding nonlinear optimization problem. To keep the online optimization computationally tractable when using GreenLight, the MPC prediction horizon in this study is restricted to one hour. Therefore, this work, specifically, assessed whether trajectory-selection RL–MPC can improve closed-loop performance when computational demands restrict the MPC controller to a short prediction horizon. The GreenLight model is used for both closed-loop simulation and as the prediction model in the finite-horizon optimization problem.

The closed-loop performance of trajectory-selection RL–MPC is evaluated against standalone RL and MPC. The three methods are compared in two complementary simulation experiments:

1. *Trajectory-selection RL–MPC guided by a specialist RL policy.* In this setting, the RL policy is trained on a single weather trajectory, and the three control methods are evaluated on that specific weather trajectory.
2. *Trajectory-selection RL–MPC guided by a generalist RL policy.* In this setting, the RL policy is trained on a broader distribution of weather trajectories, and the three control methods are evaluated on 25 weather trajectories held out during policy training.

The specialist-guided and generalist-guided controllers are evaluated separately because they are trained for different purposes and are intended to demonstrate different controller characteristics. The specialist-policy experiment assesses whether trajectory-selection RL–MPC with the same prediction horizon as MPC can achieve the closed-loop performance of a high-performing RL policy by transferring the policy's longer economic foresight. The generalist-policy experiment tests whether trajectory-selection RL–MPC improves closed-loop performance over the guiding RL policy when evaluated on weather trajectories held out during policy training. Because closed-loop evaluation with GreenLight as both the simulation model and the MPC prediction model is computationally demanding, the simulation experiments evaluate controller performance over closed-loop simulation periods spanning one day. Therefore, the simulations assess whether a guiding RL policy can transfer economic information learned over a substantially longer horizon to a one-hour MPC optimization within the trajectory-selection RL–MPC framework. The extrapolation of these results to full fruit production cycles is discussed in Section 4.

Specifically, the contributions of this work are threefold:

- *Trajectory-selection RL–MPC*, a new policy-guided MPC framework that compares the MPC-predicted trajectory with the RL rollout trajectory and applies the first control input from the trajectory with the best finite-horizon objective value. The closed-loop performance of trajectory-selection RL–MPC guided by a specialist or generalist policy was evaluated and compared with standalone RL and MPC.
- Simulation results showing how the framework can incorporate the benefit of fruit yield into an MPC optimization problem using a relatively short one-hour prediction horizon (short-horizon MPC) for greenhouse fruit-production systems. In addition, the short-horizon MPC optimization refines control input trajectories using short-term weather disturbances, improving performance relative to the guiding RL policy by reducing constraint violations and operating costs when evaluated on weather trajectories not encountered during policy training.
- Evaluation of MPC and RL–MPC using GreenLight, a large, nonlinear greenhouse tomato model with stiff dynamics that is substantially larger than control-oriented models employed in previous greenhouse control studies.

The remainder of this paper is organized as follows: Section 2 formulates the economic greenhouse control problem, describes the employed GreenLight model, and presents the RL, MPC, and trajectory-selection RL–MPC control methods. Section 3 presents the simulation results, and Section 4 discusses these results. Finally, Section 5 presents the paper's conclusions and outlines directions for future work.

2. Materials and methods

This work compared the closed-loop performance of the trajectory-selection RL–MPC framework for greenhouse fruit production control against standalone MPC and RL. Section 2.1 formulates the economic greenhouse control problem studied in this work, after which the reward function and prediction model are specified in Sections 2.2 and 2.3. Section 2.6 introduces the trajectory-selection RL–MPC framework, which combines the RL policy with the MPC optimization. The performance metrics used to evaluate these three control methods are described in Section 2.7, and the weather datasets used for training and evaluation are presented in Section 2.8. Finally, Section 2.9 describes the simulation experiments used to compare closed-loop controller performance.

2.1. Economic greenhouse control problem

This work focuses on the economic control of a greenhouse fruit production system by maximizing its closed-loop economic performance under varying weather trajectories. The greenhouse system is affected by controllable input $u \in \mathbb{R}^{n_u}$, and uncontrollable input $d \in \mathbb{R}^{n_d}$. The weather trajectory is denoted by $\mathbf{d} := \{d(0), \dots, d(T-1)\}$. For a given $\mathbf{d}$, the following scenario-conditioned economic control problem is defined as:

$$\max_{\substack{(x(0),\dots,x(T)) \\ (u(0),\dots,u(T-1))}} \sum_{t=0}^{T-1} r(x(t), x(t+1), u(t)) \tag{1a}$$

$$\left.\begin{aligned} & x(t+1) = f\big(x(t), u(t), d(t)\big) \\ \text{s.t.} \quad & u_{\min} \le u(t) \le u_{\max} \\ & |u(t) - u(t-1)| \le \delta u_{\max} \end{aligned}\right\} \quad \forall t \in \{0, \dots, T-1\} \tag{1b}$$

where $t \in \mathbb{Z}^{0+}$ represents the discrete time step and T the end of the closed-loop period. Let $x(t) \in \mathbb{R}^{n_x}$ denote the state at time step t, and let $r(\cdot)$ denote the economically based reward, including a soft linear penalty for violations of indoor climate variables. Section 2.3 further details the reward function. The nonlinear function $f(\cdot)$ represents the dynamical model that describes the system's next state given controllable and uncontrollable inputs. At each time step t, the employed controller computes a control input $u(t)$, which is applied to the system; iterating this procedure over $t = 0, \dots, T$ yields the closed-loop state $\big(x(0), \dots, x(T)\big)$ and input trajectory $\big(u(0), \dots, u(T-1)\big)$. In addition, the control input bounds and rate-of-change limits were implemented as hard constraints. Following previous work, to limit short-term variability in the indoor climate while ensuring smooth actuation, the control input $u(t)$ was constrained to limit the rate of change within 10% ($\delta = 0.1$) of the corresponding upper boundary $u_{\max}$ (Boersma et al., 2022; Morcego et al., 2023; Svensen et al., 2024; Mallick et al., 2025; Msaad et al., 2025; Van Laatum et al., 2025, 2026).

In this work, $\mathbf{d}$ was selected from a finite dataset of weather trajectories, i.e., $\mathbf{d} \in \mathcal{D}^{(i)}$, where $\mathcal{D}^{(i)}$ denotes the dataset used in a given simulation experiment, e.g., for training an RL policy or evaluating controller performance ($i \in \{\text{train}, \text{test}\}$). Section 2.8 defines the weather sets used in the simulation experiments. Throughout this work, boldface symbols denote trajectories or sequences, whereas variables evaluated at each time step are written in regular italic font.

2.2. Tomato greenhouse climate model

The nonlinear dynamical tomato greenhouse climate model considered in this work is based on GreenLight (Katzin et al., 2020). GreenLight is a high-fidelity tomato greenhouse climate model for simulating indoor climate in high-tech greenhouse facilities and has been validated against real-world greenhouse data. The tomato model is a simplified version of the tomato yield model of Vanthoor et al. (2011). This model approximates fruit yield by treating the assimilates partitioned to the fruits as marketable yield (Seginer et al., 2018; Kuijpers et al., 2021b). Consequently, the controller optimizes the climate to maximize fruit growth. Due to this approximation, the economic response time of fruit yield within the control problem, defined in (1), is faster than actual fruit development.

The resulting state-space model comprises indoor-climate states (temperature, CO_2 concentration, and humidity) and crop states representing dry weight and crop development. Table 9 lists all 27 state variables. Throughout this work, the nonlinear system dynamics are defined by the following continuous-time greenhouse model:

$$\dot{x} = f_c\big(x(t_c), u(t_c), d(t_c)\big), \tag{2}$$

where $f_c(\cdot)$ describes the system of differential equations, and $t_c \in \mathbb{R}$ represents the continuous time. To perform closed-loop simulations and apply the control methods considered in this work, a discrete-time form of this model is required. This discrete-time model was used both to generate the closed-loop simulation experiments and within the control methods: RL interacted with it during training, while MPC-based controllers used it in the finite-horizon optimization problem. The resulting discrete-time greenhouse model is given by:

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

where $x(t) \in \mathbb{R}^{27}$ denotes the state, $y(t) \in \mathbb{R}^{27}$ the output, $u(t) \in \mathbb{R}^{6}$ the control input, and $d(t) \in \mathbb{R}^{7}$ the weather disturbances. The nonlinear function $f(\cdot)$ denotes the numerical integration function, and $h(\cdot)$ is the output function. This work used a discretization time (and control interval) of $\Delta t = 300$ s, i.e., five minutes. This work considered six input variables that manipulate the indoor climate; their lower and upper bounds are listed in Table 1. For a complete model description, the reader is referred to Katzin et al. (2020).

The continuous model (2) exhibits stiff dynamics, which has implications for numerical integration of the model and controller design. Obtaining stable solutions of the discrete-time form of the model (3) requires small integration step sizes (1s) when using explicit solvers, such as n$^{\text{th}}$-order Runge–Kutta methods. Therefore, this work used CVODES (Serban and Hindmarsh, 2005), an implicit, variable-order, variable step-size integration method with automatic differentiation, to obtain a numerical solution to (3). Although accurate, this integration method is computationally demanding when solving nonlinear finite-horizon optimization problems. Nonetheless, this approach proved more efficient than explicit methods with sufficiently small step sizes. Because solving the economic control problem in (1) over the full horizon T is computationally intractable, approximate solution methods were considered instead. Specifically, this work studies RL, MPC, and a combined RL–MPC approach, which are introduced next.

2.3. Reward function

The reward function $r(\cdot)$ defining the economic control problem in (1) comprises two terms: an economic-based term and a penalty term that quantifies constraint violations. Combining the economic reward $r_e(\cdot)$ with the penalty function $r_p(\cdot)$ yields the following reward function:

$$r\big(x(t), x(t+1), u(t)\big) = r_e\big(x(t), x(t+1), u(t)\big) - r_p\big(x(t+1)\big), \tag{4}$$

where this definition is chosen such that maximizing $r(\cdot)$ corresponds to maximizing economic performance while minimizing constraint violations.

The economic reward is defined by two terms: revenue from tomato yield and operating costs associated with the controlled input, i.e., heating, CO_2 injection, and lighting electricity. To quantify the economic performance and enable optimization by the employed controller, the following economic reward function is defined:

$$\begin{aligned} r_e\big(x(t), x(t+1), u(t)\big) = c_{\text{frt}} \big(x_{\text{frt}}(t+1) - x_{\text{frt}}(t)\big) - \\ \Big(c_{\text{CO}_2} u_{\text{CO}_2}(t) + c_{\text{heat}} u_{\text{heat}}(t) + c_{\text{elec}} u_{\text{led}}(t)\Big) \Delta t, \end{aligned} \tag{5}$$

where c_{frt} is the unit selling price of tomato fruits, $x_{\text{frt}}(t)$ represents the crop's fruit dry weight at time t, c_i represent the unit costs of input u_i, and Δt is the control interval. The values of the unit price and costs c_i are defined in Section 2.9.

Practical greenhouse operation also requires maintaining indoor climate conditions within acceptable ranges. Unfavorable indoor climate conditions can induce crop diseases, which negatively affect fruit yield. Because the model does not capture this relationship between crop disease and yield, it does not directly affect the economic reward function. To better reflect greenhouse practice, the optimization problem includes constraints on three indoor climate variables: temperature, humidity, and CO_2 concentration, denoted by ($y_{\text{T,air}}, y_{\text{CO}_2\text{,air}}$, and $y_{\text{RH,air}}$), respectively. As in previous work, these constraints were implemented as soft constraints by penalizing violations of the constraints with a linear penalty function (Morcego et al., 2023; Msaad et al., 2025; Mallick et al., 2025; Van Laatum et al., 2026). This penalty function is defined as:

$$r_p\big(x(t)\big) = g_{\text{T,air}}\big(y_{\text{T,air}}(t)\big) + g_{\text{CO}_2\text{,air}}\big(y_{\text{CO}_2\text{,air}}(t)\big) + g_{\text{RH,air}}\big(y_{\text{RH,air}}(t)\big), \tag{6}$$

where the outputs $y_i(t)$ are obtained from the output function in (3). The functions g_i apply penalties for deviations from their *a priori* defined operating ranges. Each penalty function is defined as:

$$g_i\big(y_i(t)\big) = \begin{cases} \lambda_i\big(y_i(t) - y_i^{\max}\big) & \text{if } y_i(t) > y_i^{\max}, \\ \lambda_i\big(y_i^{\min} - y_i(t)\big) & \text{if } y_i(t) < y_i^{\min}, \\ 0 & \text{otherwise}, \end{cases} \tag{7}$$

Table 1
Control input u to the tomato greenhouse model (3) and corresponding lower and upper bounds.

Symbol	Lower bound ($u_{\min}$)	Upper bound ($u_{\max}$)	Unit	Description
u_{boil}	0	44	$\mathrm{W\,m^{-2}}$	The energy from the boiler to the heating pipes
u_{led}	0	116	$\mathrm{W\,m^{-2}}$	Electrical power to LED lighting
u_{CO_2}	0	5	$\mathrm{mg\,m^{-2}\,s^{-1}}$	Greenhouse CO_2 injection
u_{vent}	0	1	–	Opening of the ventilation vents
u_{thscr}	0	1	–	Thermal screen set (1 represents fully deployed)
u_{blscr}	0	1	–	Blackout screen set (1 represents fully deployed)

where λ_i denotes the penalty coefficients that regulate the scale of each penalty term. The choice of these coefficients strongly affects the controller's performance, as larger values emphasize constraint violations over economic performance, whereas smaller values do the opposite. The penalty coefficients were chosen such that the penalty term was sufficiently large to discourage constraint violations, yet remained of the same order of magnitude as the economic reward in (5). This scaling prevented the penalty term from dominating the objective while providing a practical balance between economic performance and constraint satisfaction. Section 2.9 defines the values of these penalty coefficients λ_i.

2.4. RL

This work leverages the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) to train an RL policy, denoted by π_{RL}. Two PPO-based policies are considered: a *specialist policy* and a *generalist policy*. Both policies share the same training objective and action-to-input mapping, but differ in their policy input, hyperparameter settings, and training weather data.

Each policy is trained by interacting with the dynamical model in (3) using historical weather trajectories $\mathbf{d}$ sampled from the training dataset. The goal is to approximate the economic control problem in (1) by optimizing the expected discounted return:

$$\max_{\pi_{\text{RL}}} \mathbb{E}_{(\mathbf{x},\mathbf{u})\sim\rho(\pi_{RL},\mathbf{d})}\left[\sum_{k=t}^{T-1}\gamma^{k-t} r\big(x(k),x(k+1),u(k)\big)\right], \tag{8}$$

where $x(k)$ is subject to the dynamics in (3) and control input is sampled from $u(k) \sim \pi_{\text{RL}}(\cdot)$. The joint distribution ρ is induced by the policy and the sampled weather trajectory $\mathbf{d}$. At $k = T$, the system transitions to an absorbing terminal state that yields zero reward indefinitely, allowing us to write the infinite sum of rewards using a finite-time horizon (Sutton and Barto, 2018). Ideally, RL would directly optimize the undiscounted objective posed in (1). However, computing undiscounted returns over long horizons introduces high variance in gradient estimates, which can cause convergence issues when training policies using deep neural networks. To address this, a discount factor of $\gamma = 0.995$ is applied to limit this variance. Given a five-minute control interval ($\Delta t = 300$(s)), this corresponds to an effective horizon of approximately $\frac{1}{1-\gamma} = 200$ time steps, i.e., roughly 16 h (Kearns and Singh, 2002). Given the simplified fruit-yield formulation described in Section 2.2, the trained RL policy could achieve good closed-loop performance using an effective horizon of approximately 16 h. Although this discounted objective acts as a surrogate for the objective posed in (1), it represents a necessary bias–variance trade-off to ensure the stable convergence of a policy that performs well on the undiscounted closed-loop objective.

At each time step t, the policy receives a feature vector $\phi(t)$ and produces an action $a(t) = \pi_{\text{RL}}(\phi(t))$. This action $a(t)$ is then mapped to the controllable input $u(t)$:

$$u(t) = \max\Big(\, u_{\min},\ \min\big(\, u(t-1) + a(t)\delta u_{\max}\ , u_{\max}\big)\ \Big). \tag{9}$$

Hence, the action scales the input rate of change relative to the previously applied control input. Because $a(t) \in [-1, 1]$, the input rate constraints are satisfied by construction. To comply with the input bounds, the resulting control input is clipped between $u_{\min}$ and $u_{\max}$.

The policy feature vector $\phi(t)$ is constructed from a common base observation vector $o(t)$. At each time step t, this vector is defined as:

$$\begin{aligned} o(t) = \Big[\ & x_{\text{T,air}}(t) \quad x_{\text{CO}_2}(t) \quad x_{\text{RH}}(t) \quad x_{\text{T,pipe}}(t) \quad x_{\text{frt}}(t) \quad \dots \\ & \dots \quad \sin\big(2\pi\,\tau_{\text{day}}(t)\big) \quad \cos\big(2\pi\,\tau_{\text{day}}(t)\big) \quad \sin\big(2\pi\,\tau_{\text{year}}(t)\big) \quad \cos\big(2\pi\,\tau_{\text{year}}(t)\big) \quad \dots \\ & \dots \quad u(t-1) \quad d_{\text{iGlob}}(t) \quad d_{\text{T}}(t) \quad d_{\text{CO}_2}(t) \quad d_{\text{RH}}(t) \quad d_{\text{wind}}(t)\ \Big] \in \mathbb{R}^{20} \end{aligned} \tag{10}$$

where $u(t-1) \in \mathbb{R}^6$ denotes the previously applied control input, $\tau_{\text{day}}(t) \in [0, 1)$ the normalized time of day, and $\tau_{\text{year}}(t) \in [0, 1)$ the normalized day of year. Outdoor soil and sky temperature are excluded, as these are not commonly measured in commercial greenhouse facilities (Petropoulou et al., 2023; Maree et al., 2025).

For the specialist policy, short-term temporal context is incorporated by stacking the current and previous base observations:

$$\phi_{\text{specialist}}(t) = \big[o(t-1)^\top\ o(t)^\top\big]^\top. \tag{11}$$

For the generalist policy, the observation is augmented with the weather disturbances over the next 12 time steps (one hour):

$$\begin{aligned} o_{\text{fc}}(t) = \Big[\ & o(t)^\top \quad d_{\text{iGlob}}(t{+}1{:}t{+}12)^\top \quad d_{\text{T}}(t{+}1{:}t{+}12)^\top \quad \dots \\ & \dots \quad d_{\text{CO}_2}(t{+}1{:}t{+}12)^\top \quad d_{\text{RH}}(t{+}1{:}t{+}12)^\top \quad d_{\text{wind}}(t{+}1{:}t{+}12)^\top\ \Big] \in \mathbb{R}^{80} \end{aligned} \tag{12}$$

where

$$d_i(t{+}1{:}t{+}12) \triangleq \big[d_i(t{+}1), \dots, d_i(t{+}12)\big]^\top$$

stacks the next 12 values of the weather variable i. Similar to the specialist policy, the historical temporal context is incorporated by stacking the current and the previous observations:

$$\phi_{\text{generalist}}(t) = \big[o_{\text{fc}}(t-1)^\top\ o_{\text{fc}}(t)^\top\big]^\top. \tag{13}$$

To isolate the effects of the trajectory-selection RL–MPC framework structure and weather-disturbance variability on controller performance, both policies were provided with uncertainty-free feature vectors $\phi(t)$. In particular, no uncertainty in the weather forecast was considered in $\phi_{\text{generalist}}(t)$.

A random search informed the PPO hyperparameters for both policies. Because the generalist policy has a higher-dimensional input space, its hyperparameters required additional tuning. The resulting hyperparameter settings are listed in Table 2.

The specialist policy was developed to obtain a high-performing policy for a single scenario by training exclusively on a single weather trajectory, to assess whether trajectory-selection RL–MPC can maintain the specialist policy's performance. The generalist policy was used to study how a policy trained across multiple weather trajectories can improve the performance of trajectory-selection RL–MPC under unseen weather conditions. Section 2.8 describes the training dataset used for each policy in more detail. The RL environment, including observations, rewards, and weather-trajectory sampling, was implemented in GreenLight-Gym (Van Laatum et al., 2025). Policy training was performed using Stable-Baselines-3 (Raffin et al., 2021). Training each RL policy for 4 million environment steps using the computationally intensive GreenLight model required approximately 80 min.

Table 2
PPO hyperparameters used for training the specialist and generalist policies. The generalist policy differs from the specialist policy only in the batch size, learning rate, and number of environment steps per update (n_{steps}).

Hyperparameter	Specialist	Generalist
Total time steps	4,000,000	4,000,000
Number of parallel envs (n_{envs})	8	8
Epochs per update (n_{epochs})	8	8
Discount factor (γ)	0.995	0.995
GAE parameter (λ)	0.95	0.95
Clipping range	0.2	0.2
Entropy coefficient (c_{ent})	1.8×10^{-4}	1.8×10^{-4}
Value function coefficient (c_{vf})	0.896	0.896
Policy network size (π)	[256, 256]	[256, 256]
Value network size (V_θ)	[256, 256]	[256, 256]
Activation function	SiLU	SiLU
Learning rate	8.75×10^{-5}	5.0×10^{-5}
environment steps per update (n_{steps})	1024	512
Batch size	128	64

2.5. MPC

Given the reward function in (4), and the discrete-time model in (3), at each time step t the MPC is provided with the current state $x(t)$ and the previously applied input $u(t-1)$, and solves the following finite-horizon optimization problem:

$$\max_{\mathbf{x},\mathbf{u}} \quad \sum_{k=t}^{t+N-1} r\big(x(k), x(k+1), u(k)\big) \tag{14a}$$

$$\text{s.t.} \quad \left.\begin{aligned} & x(k+1) = f(x(k), u(k), d(k)) \\ & u_{\min} \le u(k) \le u_{\max} \\ & |u(k) - u(k-1)| \le \delta u_{\max} \end{aligned}\right\} \ \forall k \in \{t, \dots, t+N-1\} \tag{14b}$$

where $x(t)$ and $u(t-1)$ enter the problem as fixed initial conditions. Although this optimization problem uses the same reward function (4) as RL, which includes fruit revenue, the optimization only optimizes this reward over its finite prediction horizon of one hour ($N = 12$), which is substantially shorter than RL's effective horizon of approximately 16 h. Consequently, MPC does not account for the economic effects of current control inputs that occur beyond this prediction horizon. The finite-horizon optimization problem is transcribed using direct multiple shooting, in which the states and control inputs are optimization variables and the model dynamics are imposed as equality constraints.

The solution of the optimization problem is denoted by the state and input trajectories:

$$\mathbf{x}^\star = \big(x^\star(t), \dots, x^\star(t+N)\big), \quad \mathbf{u}^\star = \big(u^\star(t), \dots, u^\star(t+N-1)\big). \tag{15}$$

From the computed input trajectory $\mathbf{u}^\star$, only the first element is applied to the system; that is, $u(t) = u^\star(t)$. The optimization problem is solved in a receding-horizon fashion. At time step $t+1$, the problem is solved again using the updated initial conditions $x(t+1)$ and $u(t)$, and the procedure is repeated. At each time step, the MPC is assumed to have access to full state information and perfect weather forecasts.

Due to the variable-step integrator required to evaluate (3), the computational demand of solving this optimization problem is high, and the MPC is therefore restricted to relatively short prediction horizons (e.g., 1 h) that do not capture the entire closed-loop simulation period ($N \ll T$). The prediction horizon used in the simulation experiments is specified in Section 2.9.

The nonlinear MPC optimization problem is implemented using CasADi (Andersson et al., 2019) and solved with the interior-point optimization software IPOPT (Wächter and Biegler, 2006). Several IPOPT settings were adjusted to make the optimization computationally tractable. Specifically, the Hessian is computed using a quasi-Newton approximation. In addition, the feasibility and acceptable tolerances are relaxed by setting the constraint violation tolerance to 0.01, the acceptable constraint violation tolerance to 0.1, and the acceptable optimality tolerance to 0.1. These settings allow IPOPT to terminate at approximately feasible and approximately (locally) optimal solutions, thereby reducing computational demand and improving convergence.[1] In preliminary simulation experiments, tightening the IPOPT tolerances by one order of magnitude changed the cumulative objective $\mathcal{J}$ by less than 0.02, while increasing the mean optimization time by approximately 50%. Therefore, the presented numerical results were obtained with the relaxed IPOPT tolerances.

2.6. Trajectory-selection RL–MPC

The proposed trajectory-selection RL–MPC framework extends policy-guided MPC, in which a trained RL policy provides terminal ingredients for a finite-horizon MPC optimization problem (Reiter et al., 2025; Msaad et al., 2025). The trained RL policy is used to provide economic information learned over substantially longer horizons (approximately 16 h) to a one-hour MPC optimization by defining a terminal region constraint and terminal cost. The MPC optimization can refine the control input trajectory by solving a finite-horizon optimization problem using the prediction model and short-term weather predictions. The RL rollout trajectory and the MPC-predicted trajectory are evaluated on the finite-horizon planning objective, and the trajectory with the better objective value is selected. This trajectory selection mechanism is important, as the computed solution from the finite-horizon optimization problem is not guaranteed to outperform the RL rollout. This observed behavior may occur due to the presence of local minima and the relatively high tolerance required for computationally tractable solutions (as stated in the previous section) to this optimization problem. Through this mechanism, we aim to avoid the possibility that the proposed framework degrades the performance of the original RL policy due to numerical or computational issues. Together, these components are designed to improve the closed-loop performance of short-horizon MPC controllers for greenhouse fruit production across weather conditions not encountered during RL policy training.

Fig. 1 illustrates the closed-loop implementation of trajectory-selection RL–MPC. At each time step t, the current greenhouse state $x(t)$ is measured and a weather forecast $d(t : t+N-1)$ is provided over the prediction horizon N. Starting from this state, the trained RL policy π_{RL} generates a rollout trajectory over the same prediction horizon. The terminal state of this rollout is then used to construct a terminal region constraint. The framework then solves a finite-horizon optimization problem over the same horizon N, yielding a predicted trajectory, whose terminal state $x^\star(t+N)$ is constrained to lie within this terminal region. After optimization, both the RL rollout trajectory and the MPC-predicted trajectory are evaluated using the same finite-horizon planning objective. The trajectory with the better objective value is selected, and only the first input $u(t)$ of the selected trajectory is applied to the greenhouse system. Similar to nominal MPC, the framework is assumed to have access to full state information and perfect weather forecasts.

The trained RL policy π_{RL} is rolled out in simulation from the current state $x(t)$, yielding a state and input trajectory over the prediction horizon N:

$$\tilde{\mathbf{x}} = (\tilde{x}(t), \dots, \tilde{x}(t+N)), \quad \tilde{\mathbf{u}} = (\tilde{u}(t), \dots \tilde{u}(t+N-1)). \tag{16}$$

This rollout uses the same policy input $\phi(t)$ construction and action-to-input mapping as described in Section 2.4.

[1] Note that local optimality is guaranteed only up to some tolerance, which is a typical (but not always discussed) aspect of numerical optimization for MPC. Hence, other trajectories may exist that achieve better performance than this computed solution, but locally optimal solutions are often sufficient for good MPC performance in industrial practice.

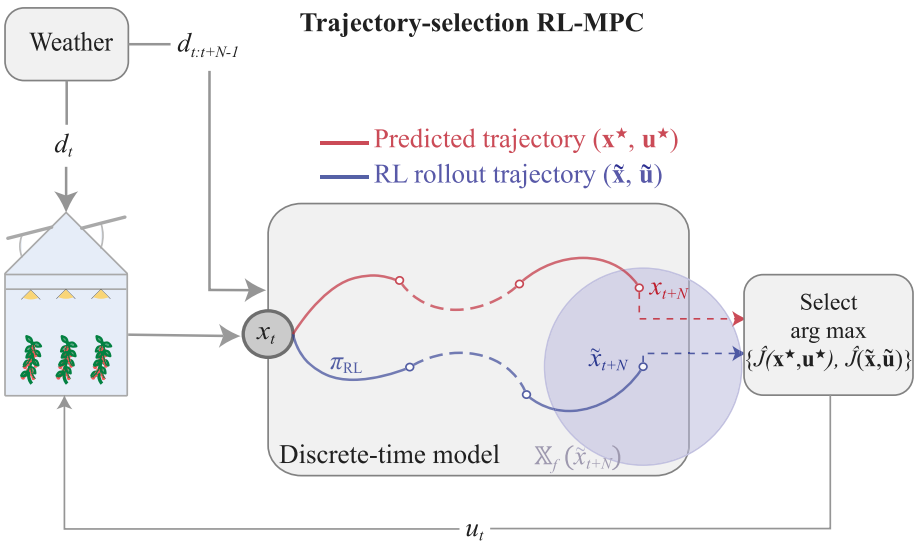


Fig. 1. Illustrative sketch of trajectory-selection RL–MPC in closed-loop simulation. At time step t, from the measured greenhouse state $x(t)$, the RL policy generates a rollout trajectory $(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$ over the prediction horizon N using future weather disturbances $d_{t:t+N-1}$. Next, MPC solves a finite-horizon optimization problem over the same horizon N, yielding the predicted trajectory $(\mathbf{x}^\star, \mathbf{u}^\star)$. The shaded region represents the terminal region constraint $\mathbb{X}_f$, constructed around the terminal state of the RL rollout, $\tilde{x}(t+N)$. The terminal state $x^\star(t+N)$ of the MPC-predicted trajectory is constrained to end in this terminal region. The first control input $u(t)$ of the trajectory with the better finite-horizon objective value $\hat{J}$ is applied to the greenhouse system. Both the RL rollout and the finite-horizon optimization use the discrete-time model $f(\cdot)$ (3) to predict the evolution of the state trajectory.

Trajectory-selection RL–MPC incorporates a terminal region constraint and a terminal cost into the optimization problem. The terminal region constraint is defined around the terminal state of the RL rollout $\tilde{x}(t+N)$ and constrains the predicted terminal state to remain within a region around this suggested terminal state:

$$\mathbb{X}_f\big(\tilde{x}(t+N)\big) = \left\{x \in \mathbb{X} \mid |x - \tilde{x}(t+N)| \leq \omega \bar{x}_{\text{ref}}\right\}, \tag{17}$$

where the inequality is applied element-wise, $\bar{x}_{\text{ref}}$ is a normalization vector, $\omega > 0$ scales the region size proportional to this vector, and $\tilde{x}(t+N) \in \mathbb{X}$ denotes the terminal state of the RL rollout.

The terminal cost penalizes deviations from the rollout terminal state, allowing the optimizer to trade off deviations from the terminal state against improvements in the objective. The MPC optimization can improve over the rollout trajectory while remaining close to the terminal state. However, if the RL policy provides a poor terminal state, these terminal ingredients may restrict the improvement MPC can achieve. The terminal cost is defined as follows:

$$\hat{V}_f\big(x, \tilde{x}(t+N)\big) = \rho \sum_{i=1}^{n} \left| \frac{x_i - \tilde{x}_i(t+N)}{\bar{x}_{\text{ref},i}} \right|, \tag{18}$$

where ρ is a scaling factor and $\bar{x}_{\text{ref}}$ normalizes the state differences element-wise. A linear L_1 penalty is used instead of a quadratic L_2 penalty to prevent large deviations in individual states from dominating the terminal cost. For all simulation experiments, the design parameters are set to $\omega = 0.05$ and $\rho = 10^{-5}$.

The normalization vector $\bar{x}_{\text{ref}}$ is computed once at the start of each simulation. First, a reference trajectory

$$x_{\text{ref}} = \big(x(0), \ldots, x(T)\big)$$

is generated by rolling out π_{RL} from the initial state $x(0)$ under the weather trajectory $\mathbf{d}$ using the dynamical model in (3). Next, the average state of this reference trajectory is computed as:

$$\bar{x}_{\text{ref}} = \frac{1}{T+1} \sum_{k=0}^{T} x_{\text{ref}}(k). \tag{19}$$

This normalization affects only the scaling of the terminal region constraint and terminal cost, not the framework structure.

Finite-horizon optimization problem

Combining the reward with the terminal cost yields the finite-horizon planning objective over the horizon N:

$$\hat{J}(\mathbf{x}, \mathbf{u}, \tilde{x}(t+N)) = \sum_{k=t}^{t+N-1} r\big(x(k), x(k+1), u(k)\big) - \hat{V}_f\big(x(t+N), \tilde{x}(t+N)\big). \tag{20}$$

Thus, the framework aims to maximize cumulative economic reward over the horizon while penalizing deviations from the RL rollout terminal state.

Given the planning objective $\hat{J}$ in (20), the dynamical model in (3), and the terminal region constraint in (17), trajectory-selection RL–MPC solves the following finite-horizon optimization problem:

$$\max_{\mathbf{x},\mathbf{u}} \quad \hat{J}(\mathbf{x}, \mathbf{u}, \tilde{x}(t+N)) \tag{21a}$$

$$\text{s.t.} \quad \left.\begin{aligned} & x(k+1) = f\big(x(k), u(k), d(k)\big) \\ & u_{\min} \leq u(k) \leq u_{\max} \\ & |u(k) - u(k-1)| \leq \delta u_{\max} \end{aligned}\right\} \quad \forall k \in \{t, \ldots, t+N-1\} \tag{21b}$$

$$x(t+N) \in \mathbb{X}_f\big(\tilde{x}(t+N)\big) \tag{21c}$$

where the current state $x(t)$ and the previously applied input $u(t-1)$ are set as fixed initial conditions. The RL rollout trajectories in (16) are used as a warm start for solving (21). Equivalent to the optimization problem of standalone MPC posed in Section 2.5, this problem is transcribed using direct multiple shooting.

The trajectory-selection RL–MPC framework is assumed to have full access to the system state. The solution of the optimization problem is denoted by the state and input trajectories:

$$\mathbf{x}^\star = \big(x^\star(t), \ldots, x^\star(t+N)\big), \quad \mathbf{u}^\star = \big(u^\star(t), \ldots, u^\star(t+N-1)\big). \tag{22}$$

After optimization, both the predicted trajectories $(\mathbf{x}^\star, \mathbf{u}^\star)$ and the rollout trajectories $(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$ are evaluated using the finite-horizon planning objective (20). For the RL rollout, the terminal penalty is zero by construction, so deviations from the rollout terminal state are penalized only for the MPC-predicted solution. Finally, the first input of the trajectory with the higher objective value is applied to the system. This *selection mechanism* allows the framework to fall back to the

Algorithm 1 Trajectory-selection RL–MPC controller in closed-loop simulation

Require: $f(\cdot)$: dynamical model (3)
$x(0)$, $u(0)$: initial state/input
π_{RL} : learned policy
$\mathbf{d}$: weather trajectory
p : model parameters
N, T : prediction horizon, length of simulation

Compute $\bar{x}_{\mathrm{ref}}$ using π_{RL}, $\mathbf{d}$, starting at $x(0)$ and subject to $f(\cdot)$
for $t = 0, \ldots, T-1$ **do**
  Obtain RL rollout over N: $(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$
  Compute objective of the rollout: $\hat{J}_{\mathrm{RL}} \leftarrow \hat{J}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$ ▷ see (20)
  Construct terminal region constraint: $\mathbb{X}_f(\tilde{x}(t+N))$ ▷ see (17)
  Solve optimization problem using MPC: $(\mathbf{x}^\star, \mathbf{u}^\star)$ ▷ see (21)
  Compute objective predicted trajectory: $\hat{J}_{\mathrm{RL-MPC}} \leftarrow \hat{J}(\mathbf{x}^\star, \mathbf{u}^\star)$ ▷ see (20)
  $u(t) \leftarrow \begin{cases} u^\star(t), & \hat{J}_{\mathrm{RL-MPC}} \geq \hat{J}_{\mathrm{RL}}, \\ \tilde{u}(t), & \text{otherwise}, \end{cases}$
  Simulate next time step: $x(t+1) \leftarrow f(x(t), u(t), d(t))$ ▷ see (3)
end for

RL rollout when the optimization does not improve upon it. Note that the planning objective is undiscounted, whereas the policy was trained using a discounted objective. This choice is justified since this work aims to optimize the undiscounted closed-loop performance; thus, selecting on a discounted objective would not align with the undiscounted evaluation criterion. As a result, the selection mechanism may favor MPC-predicted trajectories since the RL policy was optimized for a different (discounted) objective.

At time step $t+1$, the optimization problem is solved again using the updated state measurement, the new RL rollout, and the resulting new terminal region, and the procedure is repeated. Algorithm 1 provides a complete closed-loop description.

The finite-horizon optimization problem was implemented and solved using the same software and optimization settings as the nominal MPC framework; see Section 2.5.

2.7. Performance metrics

Because this work combines and compares control methods using different optimization objectives, a discounted RL objective and an undiscounted finite-horizon MPC objective, closed-loop performance was evaluated using a common undiscounted metric. This metric assumes that, under equal product prices and quality, one kilogram of tomatoes harvested today has the same economic value as one kilogram harvested later.

Closed-loop controller performance was quantified using three metrics computed over the entire simulation period. The first metric is the cumulative objective $\mathcal{J}$, defined as the cumulative undiscounted sum of the reward (4) over the controlled period T. This metric quantifies how well each controller optimized the given economic control problem in (1). The cumulative objective for a given controller is defined as:

$$\mathcal{J}_j = \sum_{t=0}^{T-1} r\big(x(t), x(t+1), u(t)\big) \tag{23}$$

where $u(t)$ denotes the control input selected by controller $j \in \{\mathrm{MPC}, \mathrm{RL}, \mathrm{RL\text{–}MPC}\}$ at time step t, and the corresponding state sequence evolves according to the discrete-time model $f(\cdot)$ in (3). The next two metrics decompose (23) into its economic component (5) and penalty component (6). This yields the Economic Performance Indicator (EPI):

$$\mathcal{E}_j = \sum_{t=0}^{T-1} r_{\mathrm{e}}(x(t), x(t+1), u(t)), \tag{24}$$

Table 3
Weather disturbances, i.e., uncontrollable input d to the tomato greenhouse model (3).

Symbol	Unit	Description
d_{iGlob}	$\mathrm{W\,m^{-2}}$	Solar global radiation
d_{T}	°C	Outdoor temperature
$d_{\mathrm{CO_2}}$	ppm	Outdoor CO_2 concentration
d_{wind}	$\mathrm{m\,s^{-1}}$	Wind speed
d_{RH}	%	Outdoor relative humidity
$d_{\mathrm{T_{sky}}}$	°C	Sky temperature
$d_{\mathrm{T_{soil}}}$	°C	Outdoor soil temperature

and the cumulative penalty:

$$\mathcal{P}_j = \sum_{t=0}^{T-1} r_{\mathrm{p}}(x(t+1)). \tag{25}$$

This cumulative penalty metric was included since this work encodes the imposed climate boundaries as soft constraints through a linear penalty function.

Paired evaluation across weather trajectories. A paired evaluation was used to compare trajectory-selection RL–MPC against standalone MPC and RL under identical weather disturbances. Specifically, for each weather trajectory $\mathbf{d}$, the difference in a performance metric (i.e., (23), (24), or (25)) was computed and then averaged across trajectories. For a finite weather set D, the paired difference in cumulative objective is defined as:

$$\Delta\mathcal{J} = \frac{1}{|D|} \sum_{\mathbf{d} \in D} \Big(\mathcal{J}_{\mathrm{RL-MPC}}(\mathbf{d}) - \mathcal{J}_j(\mathbf{d}) \Big). \tag{26}$$

and analogously for $\Delta\mathcal{E}$ and $\Delta\mathcal{P}$ using (24) and (25).

Comparing rollout and MPC-predicted trajectories of trajectory-selection RL–MPC. To analyze the selection mechanism in trajectory-selection RL–MPC, the finite-horizon planning objective (20) for the predicted solution and the guiding-policy rollout were compared:

$$\Delta\hat{J}(t) = \hat{J}\big(\mathbf{x}^\star(t), \mathbf{u}^\star(t)\big) - \hat{J}\big(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)\big), \tag{27}$$

where $(\mathbf{x}^\star(t), \mathbf{u}^\star(t))$ denotes the predicted trajectories by solving (21) and $(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t))$ denotes the RL policy rollout trajectories at time step t.

2.8. Weather datasets

Throughout this work, the dynamical model in (3) was simulated using four finite datasets of historic weather trajectories. Table 3 lists the seven disturbance variables that the model takes as uncontrollable inputs. Two datasets were used for policy training, and two were used for controller evaluation.

The specialist policy was trained on a single weather trajectory corresponding to June 1, 2023. This specialist-policy training dataset is defined as:

$$\mathcal{D}^{\mathrm{train}}_{\mathrm{specialist}} := \{\mathbf{d}_{\mathrm{June\,1,\,2023}}\} \tag{28}$$

The generalist RL policy was trained across multiple weather trajectories sampled from the period April 1 to June 1 over six years (2013–2018). This generalist-policy training dataset is denoted by:

$$\mathcal{D}^{\mathrm{train}}_{\mathrm{generalist}} := \Big\{ \mathbf{d}_{\mathrm{Apr\text{–}Jun},\,y} \;\Big|\; y \in \{2013, \ldots, 2018\} \Big\}, \tag{29}$$

where each $\mathbf{d}$ represents an individual daily weather trajectory.

The specialist policy, standalone MPC controller, and trajectory-selection RL–MPC guided by the specialist policy were evaluated on the training weather trajectory:

$$\mathcal{D}^{\mathrm{test}}_{\mathrm{specialist}} = \mathcal{D}^{\mathrm{train}}_{\mathrm{specialist}}.$$

The generalist policy, standalone MPC controller, and trajectory-selection RL–MPC guided by the generalist policy were evaluated on

Table 4
Evaluation dates for the generalist-policy dataset $\mathcal{D}^{\text{test}}_{\text{generalist}}$. Day index is zero-based (day 0 = 1 Jan). Dates selected by day-of-year differ in leap years (2012, 2020).

Year	Day of year				
	90	105	120	135	151
2011	01 Apr	16 Apr	01 May	16 May	01 Jun
2012	31 Mar	15 Apr	30 Apr	15 May	31 May
2019	01 Apr	16 Apr	01 May	16 May	01 Jun
2020	31 Mar	15 Apr	30 Apr	15 May	31 May
2023	01 Apr	16 Apr	01 May	16 May	01 Jun

Table 5
Economic reward, linear penalty function coefficients, simulation settings.

Variable	Value	Unit	Description
c_{CO_2}	0.30	€/kg	CO_2 price
c_{heat}	0.09	€/kWh	Heating price
c_{elec}	0.20	€/kWh	Electricity price
c_{frt}	20.-	€/kg{DW}	Dry matter fruit weight price[a]
λ_{CO_2}	5×10^{-5}	–	Coefficient for CO_2 violations
λ_{T}	5×10^{-3}	–	Coefficient for temperature violations
λ_{RH}	7×10^{-4}	–	Coefficient for relative humidity violations
Δt	300	s	Control interval
N	12	–	Prediction horizon (time steps)
T	288	–	Simulation length (time steps)

[a] Corresponding to a fresh weight price of 1.2 €/kg.

25 individual dates, denoted by $\mathcal{D}^{\text{test}}_{\text{generalist}}$, selected from years held out during training (2011, 2012, 2019, 2020, and 2023). Accordingly, the weather trajectories encountered during evaluation are disjoint from the generalist training set $\mathcal{D}^{\text{train}}_{\text{generalist}} \cap \mathcal{D}^{\text{test}}_{\text{generalist}} = \emptyset$. The specific dates included in $\mathcal{D}^{\text{test}}_{\text{generalist}}$ are listed in Table 4.

Weather data for 2011–2020 consists of real-world hourly measurements recorded at Schiphol Amsterdam, The Netherlands (KNMI, 2020). These measurements were interpolated to a five-minute resolution using cubic interpolation. Weather data from 2023 originates from the Autonomous Greenhouse Challenge (AGC) at Bleiswijk 2023, The Netherlands (Righini et al., 2024), and was already measured at five-minute intervals.

2.9. Simulation experiments description

The closed-loop performance of the proposed trajectory-selection RL–MPC framework was evaluated in two sets of simulation experiments, each guided by a different RL policy: the *specialist policy* and the *generalist policy*. To obtain numerical results within a computationally feasible time, all closed-loop simulations spanned a fixed period of one day, resulting in $T = 288$ time steps. All the MPC optimization problems used a one-hour prediction horizon ($N = 12$). Table 5 details all the simulation settings used in the finite-horizon optimization problems.

RL policies were trained offline using PPO, as described in Section 2.4, i.e., by maximizing the expected discounted objective in (8), with a single random seed. The training and evaluation weather datasets are defined in Section 2.8. The coefficients in the economic reward in (5) and penalty function (7) are listed in Table 5. The practical operational ranges for the indoor climate variables ($y_{\text{T,air}}$, $y_{\text{CO}_2\text{,air}}$, $y_{\text{RH,air}}$) were implemented as soft constraints and their upper and lower bounds are listed in Table 6. The next two subsections detail the specialist policy and generalist policy experiment sets.

All closed-loop simulations and RL policy training were performed on a workstation equipped with an Intel(R) Xeon(R) W-2133 CPU @ 3.60 GHz. The MPC and trajectory-selection RL–MPC nonlinear optimization problems were implemented in CasADi 3.6.7 using Python 3.11.15 and solved with IPOPT 3.14.3.

Trajectory-selection RL–MPC guided by the specialist policy

Three simulation experiments evaluated the performance and the behavior of the trajectory-selection RL–MPC framework when guided by the specialist RL policy. This policy was trained on a single weather trajectory $\mathcal{D}^{\text{train}}_{\text{specialist}}$ (defined in (28)), yielding a high-performing specialist RL policy. All controllers were evaluated on that same weather trajectory $\mathcal{D}^{\text{test}}_{\text{specialist}} = \mathcal{D}^{\text{train}}_{\text{specialist}}$.

1. First, the three resulting controllers (RL, MPC, and trajectory-selection RL–MPC guided by the specialist policy) were evaluated using the closed-loop performance metric: the cumulative objective (23), the EPI (24), and the cumulative penalty (25). This experiment assessed whether trajectory-selection RL–MPC can incorporate delayed economic information, learned by a high-performing policy, into a short-horizon MPC optimization problem using a one-hour prediction horizon. A breakdown of the underlying costs and penalties provided insight into which strengths of the underlying methods contributed to controller performance.
2. Second, the selection mechanism of trajectory-selection RL–MPC was analyzed by evaluating $\Delta\hat{J}$ in (27) over the simulation day. This analysis should show whether the input $u(t)$ from the RL rollout trajectory or from the MPC-predicted trajectory was selected and applied at each time step. It is hypothesized that the RL rollout trajectory would be selected more often during daytime, when solar radiation requires balancing current operating costs against future fruit yield.
3. Third, an ablation study removed three elements of the trajectory-selection RL–MPC framework (Algorithm 1) one at a time: the terminal cost $\hat{V}_f$, the terminal region constraint $\mathbb{X}_f$, and the selection mechanism. Each ablated version of the framework was evaluated by the cumulative objective, EPI, and cumulative penalty on $\mathcal{D}^{\text{test}}_{\text{specialist}}$. When the selection mechanism was removed, the first input was always taken from the MPC-predicted trajectory $\mathbf{u}^\star$. This experiment assessed which components of the framework are necessary for the observed performance gains.

Trajectory-selection RL–MPC guided by the generalist policy

The second set of simulation experiments evaluated trajectory-selection RL–MPC when guided by the generalist RL policy on weather trajectories held out during policy training. This policy was trained by sampling weather trajectories uniformly at random from $\mathcal{D}^{\text{train}}_{\text{generalist}}$ (described in Section 2.8). The three controllers (RL, MPC, and trajectory-selection RL–MPC guided by the generalist policy) were evaluated on 25 individual test dates, denoted by $\mathcal{D}^{\text{test}}_{\text{generalist}}$. The specific dates are listed in Table 4. Controller performance was compared using the paired evaluation metric in (26) for the cumulative objective, EPI, and cumulative penalty. This simulation experiment aimed to assess whether trajectory-selection RL–MPC can improve closed-loop performance relative to the guiding RL policy and MPC on weather trajectories not encountered during policy training. Although training RL policies on broader weather datasets can improve generalization, it does not prevent degraded performance under unseen weather conditions. Therefore, it is hypothesized that trajectory-selection RL–MPC could further improve over the guiding RL policy through finite-horizon MPC optimization using short-term weather predictions. In addition, the experiment assessed when the proposed framework benefits most from the guiding RL policy or from short-horizon MPC optimization. The relationships between daily mean outdoor temperature, global radiation, and relative humidity, and both the paired performance difference ΔJ and the trajectory-selection mechanism, were analyzed. This analysis assessed whether unseen weather conditions outside or near the boundary of the training distribution were associated with larger performance differences between trajectory-selection RL–MPC and the guiding RL policy, and whether the selection mechanism relied

Table 6
Upper and lower boundaries for the three indoor climate variables.

Symbol	Lower bound ($y_i^{\min}$)	Upper bound ($y_i^{\max}$)	Unit	Description
$y_{T,air}$	10	25	°C	Greenhouse air temperature
$y_{CO_2,air}$	400	1600	ppm	Carbon dioxide concentration
$y_{RH,air}$	0	90	%	Relative humidity

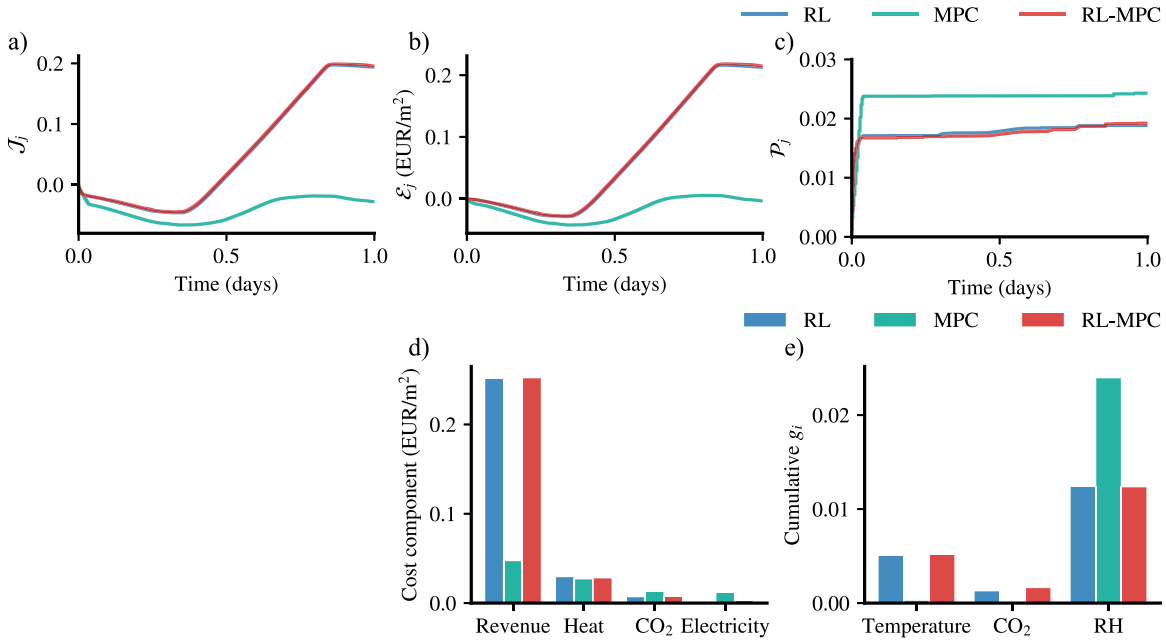


Fig. 2. Comparison of closed-loop controller performance on the specialist-policy training trajectory $\mathcal{D}_{\text{specialist}}^{\text{train}}$. This figure visualizes the cumulative sums of three performance metrics and their individual components for MPC, RL, and trajectory-selection RL–MPC, evaluated on a single weather trajectory. Note that the cumulative sums of RL and RL–MPC in (a), (b), and (c) are nearly identical and, therefore, have overlapping trajectories. The weather trajectory corresponds to the trajectory used to train the specialist RL policy, i.e., $\mathcal{D}_{\text{specialist}}^{\text{test}} = \mathcal{D}_{\text{specialist}}^{\text{train}}$. **(a)** Cumulative objective ($\mathcal{J}$) as defined in (23). **(b)** EPI ($\mathcal{E}$) as defined in (24). **(c)** Cumulative penalty ($\mathcal{P}$) as defined in (25). **(d)** Cumulative components of the economic reward as defined in (5). **(e)** Cumulative penalty of the individual penalty components g_i, as defined in (7). The objective and EPI are maximized, whereas penalty values are minimized. For the economic components, higher revenue and lower operating costs indicate better performance.

more strongly on the RL rollout or the MPC-predicted trajectory under specific weather conditions.

3. Results

3.1. Trajectory-selection RL–MPC guided by the specialist policy

The results for the specialist RL policy, MPC, and trajectory-selection RL–MPC guided by the specialist policy evaluated on the specialist policy evaluation set $\mathcal{D}_{\text{specialist}}^{\text{test}} = \mathcal{D}_{\text{specialist}}^{\text{train}}$ are presented in Fig. 2. Trajectory-selection RL–MPC, with a one-hour prediction horizon, matched the specialist policy's cumulative performance $\mathcal{J}$. The cumulative objective, EPI, and cumulative penalty exhibited nearly identical cost trajectories over the optimized day, as shown in Fig. 2. The trajectory-selection RL–MPC controller provided a marginal improvement in the cumulative objective $\mathcal{J}$ (0.196 vs 0.194) compared to the specialist RL policy. This gain was driven by a relatively small improvement in the EPI (0.215 vs 0.213), while the cumulative penalty was identical (0.019) for this specific weather trajectory. The plots of the individual components of the economic reward and the penalty function confirm that trajectory-selection RL–MPC performed nearly identically to the specialist policy. With the same prediction horizon, the trajectory-selection RL–MPC controller outperformed the MPC controller by a relatively large margin ($\Delta\mathcal{J} = 0.224$). This improvement was driven by a higher EPI (0.196 vs −0.004). The higher EPI was mainly due to increased revenue under trajectory-selection RL–MPC compared with MPC (+0.20), while similar amounts of heating and CO_2 injection were applied. In addition, the MPC controller incurred a higher cumulative penalty than the trajectory-selection RL–MPC controller (+0.005). This increase was primarily due to more severe violations of the relative humidity boundary at the start of the day.

At each time step, the first control input from the trajectory (rollout or predicted) with the best finite-horizon planning objective (20) was selected and applied to the system. Evaluating the difference in the planning objective $\Delta\hat{J}$ (27) between the policy rollout and the MPC-predicted trajectory (with terminal ingredients) shows that the specialist policy found better solutions during a consistent period of the daytime; see Fig. 3. This period starts a few hours after sunrise, i.e., the start of daytime, and ends around sunset. This period also corresponds to the period during which both RL and trajectory-selection RL–MPC improved their cumulative objective the most compared to MPC in Fig. 2(a). The heatmap confirms this behavior, as the policy rollout had a higher objective at each prediction step (k) during the daytime. Conversely, at the end of the simulated day, the MPC-predicted trajectories yielded better solutions, indicated by a higher finite-horizon planning objective $\hat{J}$. At the start of the simulated day, trajectory-selection RL–MPC frequently switched between selecting the input from the rollout or the MPC-predicted trajectory.

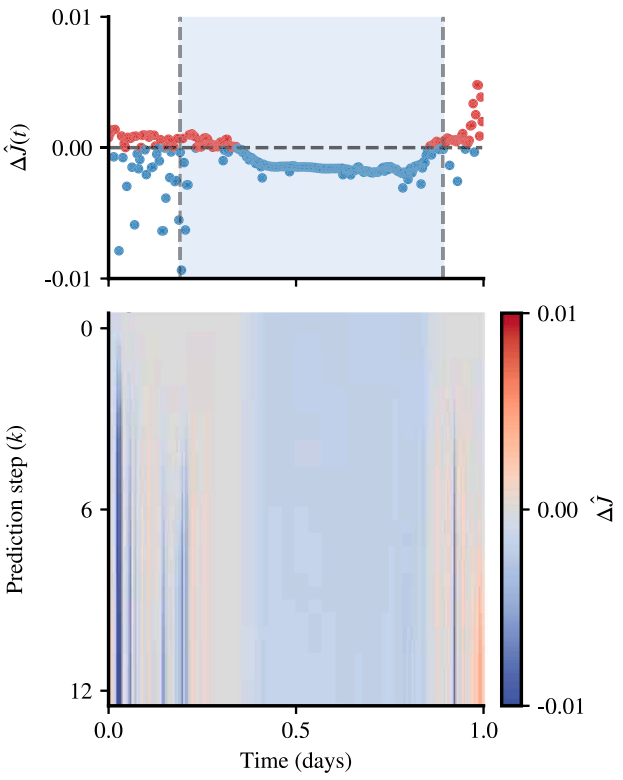


Fig. 3. Evaluating the finite-horizon planning objective of the policy rollout and the MPC-predicted trajectory for $\mathcal{D}^{\text{train}}_{\text{specialist}}$. Top plot: Planning-objective difference $\Delta\hat{J}(t)$ (27) between the rollout generated by π_{RL} and the MPC-predicted solution over the prediction horizon N at time step t, plotted against time of day. Red (positive) values indicate that the MPC-predicted trajectory (with terminal ingredients) achieved a higher planning objective, whereas blue (negative) values indicate that the policy rollout achieved a higher planning objective. **Bottom plot:** Heatmap of the planning-objective difference between the rollout and MPC-predicted trajectories evaluated up to prediction step k, plotted against time of day. Again, red indicates that the MPC optimization achieved a higher planning objective at prediction step k, whereas blue indicates that the RL rollout achieved a higher planning objective.

The ablation study revealed that removing the selection mechanism had the largest impact on controller performance, as the cumulative objective dropped by 12.2% from (0.196) to (0.172). This performance degradation is mainly due to the cumulative penalty, which increased by 95.2% from (0.021) to (0.041). This increase in cumulative penalty occurred primarily at the beginning and the end of the simulated day, when lower temperature and higher relative humidity drove the indoor climate variables closer to their constraint bounds, making penalties more likely. On the other hand, the EPI marginally decreased when discarding the input selection mechanism. Removing the terminal cost and the terminal region constraint had minimal effect on the trajectory-selection RL–MPC controller's performance when guided by the specialist policy.

3.2. Trajectory-selection RL–MPC guided by the generalist policy

Fig. 5 summarizes the simulation results on the evaluation set $\mathcal{D}^{\text{test}}_{\text{generalist}}$ for the generalist RL policy, MPC, and trajectory-selection RL–MPC guided by the generalist policy. Across all 25 evaluated dates, trajectory-selection RL–MPC outperformed both standalone controllers. On average, trajectory-selection RL–MPC increased performance by 0.041 ± 0.013 (+52.2%) relative to the generalist RL policy. This improvement was driven by a higher EPI (+0.025 EUR/m^2) and a lower cumulative penalty (−0.024). An analysis of the economic reward components shows that the trajectory-selection RL–MPC controller matched the revenue achieved by the generalist RL policy (0.222 EUR/m^2), while reducing heating costs to 0.047 EUR/m^2 (−26.8%) and lamp electricity costs to 0.056 EUR/m^2 (−12.0%). Decomposing the linear penalty further indicates that trajectory-selection RL–MPC mainly benefits from fewer relative humidity violations, with a cumulative penalty of 0.020 ± 0.012 (−47.7%).

Compared to standalone MPC, trajectory-selection RL–MPC increased the cumulative objective by a margin of 0.156 ± 0.031 (+79.9%). This gain was mainly due to an increase in EPI of 0.150 EUR/m^2. Examining the cost components suggests that the discrepancy in EPI between the two controllers is explained by the low revenue achieved by MPC (0.063 EUR/m^2). Trajectory-selection RL–MPC also spent less on electricity for the lamps (−13.8%). The difference in cumulative penalty between the trajectory-selection approach and MPC is relatively small (−0.006±0.016). The dependence of indoor temperature violations on external weather conditions combined with a relatively tight upper limit (25 °C) led to relatively large confidence intervals and cumulative penalties for all three controllers, as shown in Fig. 5(c). As outdoor temperature and global radiation increase, the controllers cannot sufficiently reduce indoor temperature, since the modeled greenhouse system lacks an active cooling system.

Fig. 6(d–f) plots the performance improvements over the two standalone controllers against the daily mean outdoor temperature, global radiation, and relative humidity ($\bar{d}_{\text{T}}$, $\bar{d}_{\text{iGlob}}$, $\bar{d}_{\text{RH}}$) for each day in $\mathcal{D}^{\text{test}}_{\text{generalist}}$. One can observe that trajectory-selection RL–MPC outperforms both standalone controllers on all evaluation days, as indicated by $\Delta J > 0$. In particular, the performance difference between trajectory-selection RL–MPC and MPC shows a positive linear relation with global radiation. Higher global radiation corresponds to larger improvements of trajectory-selection RL–MPC over MPC. In contrast, the performance difference between trajectory-selection RL–MPC and MPC shows a negative linear relation with the daily mean relative humidity, indicating that the performance gap between the two methods narrows under higher relative humidity. The selection mechanism shows consistent trends with these variables, as shown in Fig. 6(g–i): higher daily average global radiation increased the fraction of time the guiding policy rollout yielded the better solution to (20), leading to selection of its first control input. Conversely, higher daily mean relative humidity increased the fraction of time the MPC-predicted trajectory yielded a better solution than the guiding policy rollout. Temperature did not demonstrate a positive or negative relation with the relative performance or the selection mechanism. The improvement of trajectory-selection RL–MPC over the generalist RL policy appeared less sensitive to the daily mean weather variables, even on test days whose weather statistics fall outside the training distribution; see Fig. 6(a–c).

Table 7 presents the computational performance of MPC and RL–MPC on the generalist dataset $\mathcal{D}^{\text{test}}_{\text{generalist}}$. Both methods required substantial amounts of solver time; RL–MPC used 102.3 (s) compared to 94.1 (s) for MPC, corresponding to an 8.2 (s) difference per optimization on average. Additionally, both methods had relatively large standard deviations for the mean solver times, particularly for MPC, which could indicate large outliers in the solver time. Nevertheless, 96.4% of the MPC and 97.6% of the RL–MPC optimization problems were completed within the 300 s control interval. The online computation time of the RL policy was below 0.001 s per time step and was therefore negligible compared with the optimization-based controllers.

On the specialist-policy training trajectory, trajectory-selection RL–MPC matched the guiding policy while outperforming standalone MPC with the same one-hour prediction horizon. The difference relative to MPC was primarily associated with fruit revenue rather than operating costs, suggesting that the one-hour MPC prioritized immediate objectives at the expense of crop growth.

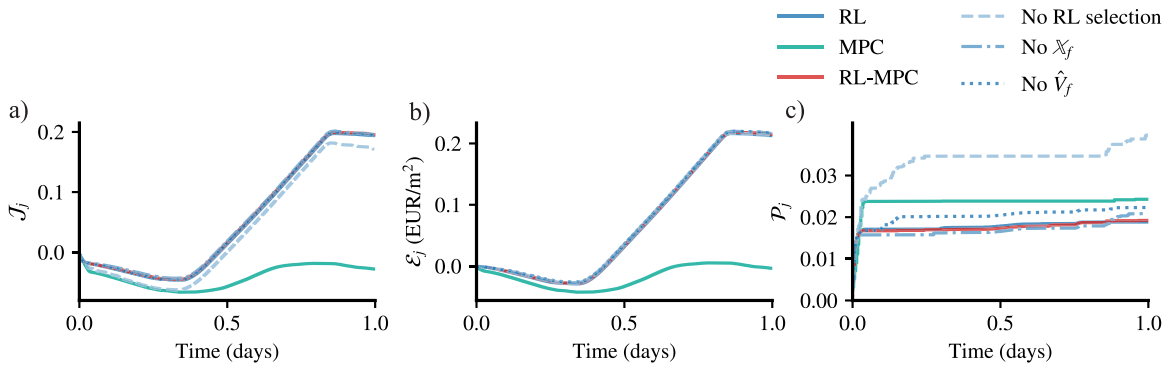


Fig. 4. Ablation study for trajectory-selection RL–MPC. These graphs show the cumulative performance metrics of MPC, RL, trajectory-selection RL–MPC, and ablated variants of the proposed framework obtained by removing individual components from the framework as defined in Algorithm 1. The resulting controllers were evaluated on the specialist-policy training trajectory, i.e., $\mathcal{D}^{\text{test}}_{\text{specialist}} = \mathcal{D}^{\text{train}}_{\text{specialist}}$. **(a)** Cumulative objective as defined in (23). **(b)** EPI ($\mathcal{E}$) as defined in (24). **(c)** Cumulative penalty ($\mathcal{P}$) as defined (25). Here, $\mathbb{X}_f$ denotes the terminal region constraint and $\hat{V}_f$ denotes the terminal cost.

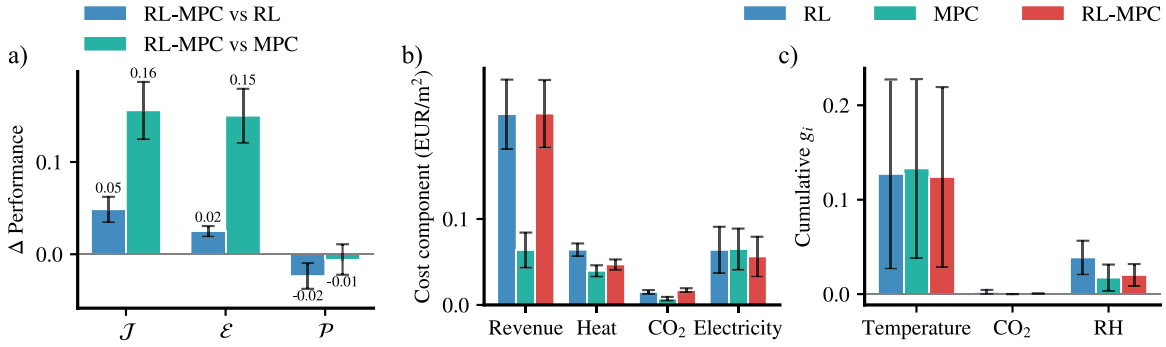


Fig. 5. Comparison of closed-loop performance of RL, MPC, and trajectory-selection RL–MPC guided by the generalist RL policy. Bars show mean values across 25 simulated dates ($\mathcal{D}^{\text{test}}_{\text{generalist}}$) with 95% confidence intervals. **(a)** Mean signed difference between RL–MPC and each standalone controller, computed using (26). Positive values of the objective $\mathcal{J}$ and EPI indicate that RL–MPC outperforms the corresponding standalone controllers. For the penalty metric, negative values indicate improvement for RL–MPC over the corresponding controller (i.e., fewer constraint violations). **(b)** Mean cumulative values of the individual components of the economic reward function defined in (5). **(c)** Mean cumulative penalty of the individual penalty components g_i, as defined in (7).

Table 7
The average cumulative objective and average solver time for approximating the MPC and RL-MPC optimization problem, including the sampled standard deviation. The solver-time statistics are computed over individual finite-horizon optimization problems, whereas cumulative-objective statistics are computed over the 25 one-day closed-loop simulations. The results originate from the generalist experiment using $\mathcal{D}^{\text{test}}_{\text{generalist}}$.

Method	Solver time		$\mathcal{J}$		Completed within Δt
	Mean (s)	Std. (s)	Mean	Std.	
MPC	94.1	184.5	−0.197	0.209	96.4%
RL-MPC	102.3	98.9	−0.041	0.227	97.6%

4. Discussion

4.1. Trajectory-selection RL–MPC guided by the specialist policy

On the specialist-policy weather trajectory $\mathcal{D}^{\text{train}}_{\text{specialist}}$, the closed-loop performance of trajectory-selection RL–MPC with a one-hour prediction horizon matched the guiding specialist policy while outperforming standalone MPC with the same prediction horizon. This improvement over MPC was mainly due to higher revenue despite only minor differences in operating costs. These results indicate that the one-hour prediction horizon is too short for MPC to balance current operating costs against the delayed benefits of crop yield. This observation suggests that MPC prioritized short-term objectives, such as minimizing constraint violations and operating costs, potentially by using the thermal and blackout screens at the expense of crop growth. Such control strategies limit PAR entering the greenhouse, thereby inhibiting crop yield. In contrast, trajectory-selection RL–MPC achieved the same revenue as the specialist RL policy despite using the same one-hour MPC prediction horizon.

These results demonstrate a general limitation of the one-hour MPC controller for the greenhouse fruit production control problem considered in this work. A one-hour prediction horizon for receding horizon approaches, such as MPC, can lead to myopic behavior in greenhouse fruit production systems. Such horizon lengths are typically too short to balance immediate operating costs against the delayed benefit of fruit yield. Ideally, MPC would use prediction horizons spanning the entire production period (Van Henten and Bontsema, 2009; Van Straten and Van Henten, 2010; Kuijpers et al., 2021a). However, as greenhouse fruit production models become more complex and uncertainties in weather forecasts, product prices, and crop dynamics are incorporated, optimizing over such long horizons becomes computationally intractable. This problem setting motivates control methods that incorporate longer-term economic information into finite-horizon control methods. In this work, trajectory-selection RL–MPC improved closed-loop performance over MPC with the same prediction horizon through terminal region constraints, terminal costs, and trajectory selection.

All simulations in this work spanned a single day. In practice, greenhouse fruit development lasts approximately 40–60 days (De Koning, 1994; Adams et al., 2001). The RL policy was trained using the discounted return over the complete episode, with a discount factor $\gamma = 0.995$, corresponding to an effective horizon of approximately

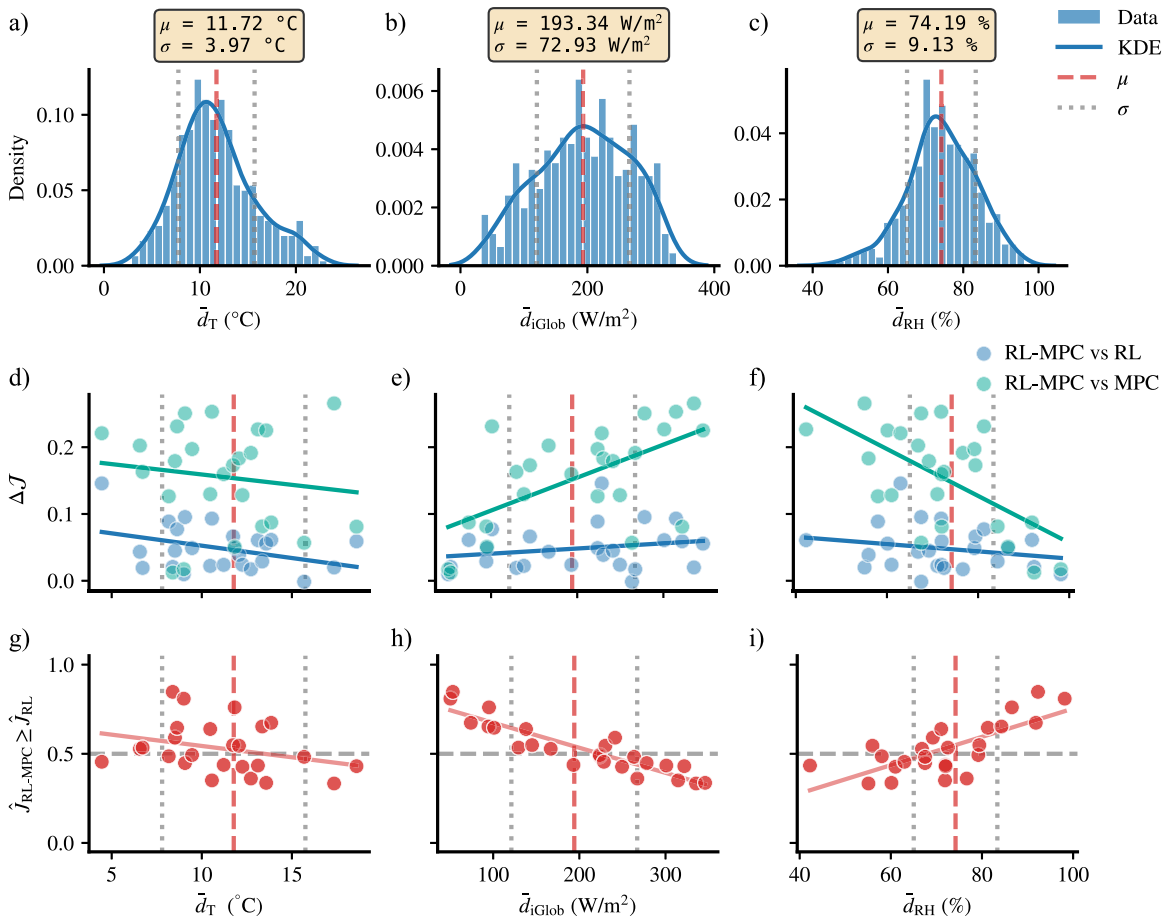


Fig. 6. Relation between daily mean weather variables, trajectory-selection RL–MPC performance gains, and the input selection mechanism. Top row: Empirical distributions and kernel density estimates (KDEs) of the daily mean weather variables in the training set $\mathcal{D}^{\text{train}}_{\text{generalist}}$: (a) outdoor temperature $\bar{d}_T$, (b) global radiation $\bar{d}_{iGlob}$, (c) relative humidity $\bar{d}_{RH}$. Daily mean was computed for all 366 days in this set. The red dashed line indicates the mean μ, and the gray dotted lines represent $\mu \pm \sigma$. **Middle row:** Closed-loop improvement ΔJ (26) of RL–MPC over the generalist RL policy and MPC on individual days in the test set $\mathcal{D}^{\text{test}}_{\text{generalist}}$. The x-axis shows the daily mean weather variables for each evaluated day: (d) outdoor temperature $\bar{d}_T$, (e) global radiation $\bar{d}_{iGlob}$, (f) relative humidity $\bar{d}_{RH}$. Solid lines indicate linear trend fits. **Bottom row:** Fraction of time the MPC-predicted trajectory attained a higher finite-horizon planning objective $\hat{J}$ (20) than the RL rollout trajectory. The x-axis shows the daily mean weather variables for each evaluated day: (g) outdoor temperature $\bar{d}_T$, (h) global radiation $\bar{d}_{iGlob}$, (i) relative humidity $\bar{d}_{RH}$. Solid lines indicate linear trend fits.

16 h. This effective horizon is substantially shorter than the timescale associated with fruit development. However, this work adopted a simplified fruit-yield formulation that treats all assimilates partitioned to fruits, minus those used for maintenance respiration, as marketable yield (Seginer et al., 2018; Katzin et al., 2020; Kuijpers et al., 2021b). This approximation reduces the response time of fruit yield compared to the actual economic return from fruit production. Using this simplified fruit-yield formulation, the RL policy achieved high closed-loop performance with an effective horizon of approximately 16 h. The presented results therefore demonstrate that trajectory-selection RL–MPC can incorporate economic information learned by an RL policy over a horizon (16 h) into a finite-horizon optimization problem with a substantially shorter prediction horizon (one hour). The results do not demonstrate that the RL policy can learn and transfer economic information of a complete fruit production cycle (spanning multiple weeks) to short-horizon MPC.

This distinction is relevant since the employed fruit yield model does not account for fruit development stages. Consequently, economic return in real-world tomato production is more complex than that represented in the studied control problem, as ignoring fruit development may overestimate short-term yield (Katzin et al., 2020). When fruit development dynamics are explicitly represented in the prediction model, current control decisions influence fruit yield over extended periods and may require an RL policy that captures economic consequences across a longer effective horizon. For that, specifically, two things should be tested: (1) can RL capture/learn delayed economic information over these extended timescales (weeks), and (2) can we still transfer this information to short-horizon MPC optimization.

The benefit of the guiding policy is further demonstrated by comparing the finite-horizon planning objective $\hat{J}$ in (20) between the trajectories from the policy rollout and the trajectory-selection RL–MPC optimization. For a consistent period during the daytime, the policy rollout attained a higher planning objective than the MPC-predicted trajectory; see Fig. 3. In this period, global radiation induced crop growth, and trajectory-selection RL–MPC therefore selects the control input from the policy rollout since the policy benefits from its relatively long effective horizon (approximately 16 h; Section 2.4) compared with the one-hour optimization horizon. This interval also coincides with the steep rise in the cumulative objective and EPI observed in Fig. 2. Near the end of the simulated day, the MPC-predicted trajectories yielded better solutions, as indicated by a positive $\Delta\hat{J}$. After sunset, the indoor temperature dropped, and relative humidity rose, increasing the importance of constraint handling, a known strength of MPC.

The decreased performance of standalone MPC regarding the constraint violations, shown in Fig. 2, and the relatively frequent selection of the RL rollout trajectory, shown in Fig. 3, may be partly explained

by the relaxed optimization tolerances used in the finite-horizon MPC problem, as mentioned in Section 2.5. With these settings, the nonlinear optimizer may terminate before converging to a locally optimal solution, with a worse finite-horizon planning objective value than the RL rollout. As a result, MPC-predicted trajectories may have been rejected by the trajectory-selection mechanism, even though stricter tolerances might have produced improved MPC solutions. The relaxed tolerances were used to keep the closed-loop simulations computationally feasible. Imposing stricter tolerances on the optimization problem could improve the performance of MPC-predicted trajectories, possibly increasing the benefit of trajectory-selection RL–MPC. However, this would increase the controller's computational demand and does not eliminate the limitation imposed by the short prediction horizon.

This work trained the RL policy using a discounted objective by introducing the discount factor $\gamma = 0.995$, as described in Section 2.4. For the considered economic greenhouse control problem, the target performance metric is undiscounted, since economic returns (e.g., fruit yield) obtained later in the production period are not worth less under equal prices and product quality. Ideally, this would correspond to RL using $\gamma = 1$. However, training RL policies using deep neural networks with an undiscounted objective introduces high variance in gradient estimates, which can make policy optimization unstable. To limit this variance and improve the convergence of RL policies, discounting was applied as a compromise between longer-term economic performance and maintaining stable RL training. This discounting therefore changes the objective for which the RL policy is optimized relative to the undiscounted control objective (1). Consequently, evaluating the RL policy, the MPC controller, and trajectory-selection RL–MPC using common undiscounted closed-loop metrics introduces an objective mismatch for RL. Similarly, this mismatch may affect the selection mechanism, which compares the MPC-predicted trajectory and the RL rollout using the undiscounted finite-horizon planning objective. Thus, the selection mechanism evaluates the candidate trajectories according to the undiscounted economic objective rather than the RL training objective. This comparison may favor MPC-predicted trajectories. Nevertheless, an RL trained with a discounted objective remains useful since it can encode economic information beyond the MPC prediction horizon. Further research should evaluate how the value of the discount factor affects the learned policy, the trajectory-selection decisions, and the resulting closed-loop performance.

The ablation study highlights the role of the selection mechanism in achieving specialist-policy performance on the training trajectory. Fig. 4 shows that removing this element increased the cumulative penalty, particularly near the start of the simulated day. Fig. 3 further demonstrates the dependence of trajectory-selection RL–MPC on the policy rollout; at multiple time steps t until sunrise, the rollout attained a substantially higher finite-horizon planning objective value than the predicted trajectory. In this tested variant of trajectory-selection RL–MPC, the optimization using terminal ingredients appeared to be insufficient to maintain a low cumulative penalty and can even perform worse than standalone MPC in terms of cumulative penalty. This behavior may be caused by the quasi-Newton approximation of the Hessian and the relatively loose tolerance afforded to the optimization routine, which were selected to ensure computational tractability but did not ensure that the optimization algorithms return local optima. All three ablated versions performed identically on EPI, indicating that economic guidance provided by the RL policy is preserved, either through the selection mechanism or the terminal ingredients.

Overall, these results indicate that, on the specialist policy training trajectory, the performance of trajectory-selection RL–MPC is mainly driven by its selection mechanism. Because the guiding policy was trained extensively on this specific weather trajectory, the RL policy rollout was already high-performing, so selecting the rollout trajectory provided most of the gains in EPI and cumulative penalty. As a result, the terminal ingredients contributed less to controller performance in this setting. However, under unseen weather conditions, the policy rollout trajectory is expected to be selected less often, and the contribution of the terminal ingredients may become more important. This is further discussed for trajectory-selection RL–MPC guided by the generalist policy in Section 4.2.

Finally, all controllers were evaluated under the assumptions of full state availability and perfect knowledge of the future weather. In practice, these measurements and forecasts are subject to error; therefore, the reported controller performance should be interpreted as an upper bound. In addition, a single random seed is used to train the guiding policy. Previous work on integrating RL and MPC indicates that the RL policy varies across random seeds, but that incorporating terminal ingredients in the trajectory-selection RL–MPC optimization framework can mitigate this variance (Van Laatum et al., 2026).

4.2. Trajectory-selection RL–MPC guided by the generalist policy

On weather trajectories not encountered during policy training, trajectory-selection RL–MPC improved closed-loop performance relative to both standalone controllers. The improvement over the generalist RL policy was driven by maintaining high revenue while reducing operating costs and relative humidity penalties. This suggests that the MPC optimization of the framework, with terminal ingredients, can refine the policy rollout trajectory, thereby reducing operating costs and the cumulative penalty while maintaining revenue. Compared with standalone MPC, the increased cumulative objective was mainly associated with a higher revenue, consistent with the specialist-policy results from Section 3.1. These observations illustrate the complementary roles of the two control approaches: the guiding policy provides economic information beyond the one-hour MPC prediction horizon, whereas the MPC optimization adapts the control input trajectory to account for short-term weather conditions.

Analyzing the difference in cumulative objective showed that trajectory-selection RL–MPC outperformed the standalone controllers on all 25 simulated days; see Fig. 6. There is a positive relationship between daily mean global radiation and the performance difference between trajectory-selection RL-MPC and standard MPC. This suggests that the guiding policy becomes more beneficial when increased radiation drives crop growth, making economic information beyond the one-hour prediction horizon much more important. In contrast, this performance difference showed a negative relation with daily mean outdoor relative humidity. This may be explained by the fact that elevated outdoor relative humidity makes maintaining the soft indoor humidity constraint more important, and, therefore, reduces the emphasis on longer-term economic performance. The changes in the selection mechanism in Fig. 6 support this interpretation: policy rollouts were selected more frequently under higher global radiation, while MPC-predicted trajectories were selected more frequently under higher relative humidity.

The ablation study, discussed in the previous section, showed that, on the specialist-policy training trajectory, the selection mechanism was the primary contributor to controller performance. However, under unseen weather conditions, the predicted MPC trajectory was more likely to be selected, as illustrated in Fig. 6(g-i). This observation suggests that, in such settings, the terminal ingredients play a more important role in controller performance than in the specialist case. Previous work has also shown that policy-guided MPC with terminal constraints and terminal costs improves controller performance (Msaad et al., 2025; Van Laatum et al., 2026). Hence, these findings support the need for terminal ingredients, in addition to the selection mechanism, to improve closed-loop controller performance under unseen weather conditions.

The normalization vector $\bar{x}_{\text{ref}}$ defined in (19) was computed separately for each closed-loop simulation from an RL rollout over the complete simulation period. Since GreenLight's state variables have very different numerical magnitudes, normalization was required to

prevent individual states from dominating the terminal region constraint and terminal cost. This work computed the normalization vector using the weather trajectory over the simulation period spanning a single day. During online implementation, in production cycles that extend a single day, this dependence on future weather can be avoided by computing a fixed normalization vector offline from representative historical weather trajectories or the RL training dataset and subsequently using this vector during deployment. The Appendix A includes a supplementary experiment that compares trajectory-selection RL–MPC using a fixed normalization vector with the normalization vector defined in (19). These results indicate that the choice of a fixed normalization vector or a simulation-specific normalization vector does not alter the main findings of this study. As RL–MPC improved the cumulative objective $\mathcal{J}$ by 48.9% relative to RL and by 76.8% relative to MPC, compared to 52.2% and 79.9% when using the simulation-specific normalization vector.

Overall, these results suggest that when the test set differs from the training set, trajectory-selection RL–MPC effectively leverages the complementary strengths of its underlying methods. The controller retained the economic performance of the guiding policy through high revenue due to crop growth, while the MPC optimization improved short-term constraint handling and decreased operating costs. The trajectory-selection RL–MPC controller was evaluated on weather trajectories not included in the RL policy's training set. When the RL policy was trained on a broader weather dataset and faced unseen weather conditions at evaluation, its closed-loop performance may still decrease. Trajectory-selection RL–MPC's receding-horizon approach can refine control input trajectories, reducing operating costs and constraint violations relative to the RL policy. This mechanism is not limited to weather variability and may extend to other external disturbances, such as changes in crop prices and resource costs. However, trajectory-selection RL–MPC does require a sufficiently accurate prediction model. Modeling errors or uncertainty in the state dynamics can reduce, or even reverse, the benefit of the proposed framework. When such uncertainties affect controller performance, uncertainty-aware approaches that combine RL with scenario-based stochastic MPC (SMPC) offer a feasible alternative. For instance, RL-SMPC learns an RL policy in a stochastic setting and uses it to construct terminal ingredients and provide nonlinear feedback within scenario-based SMPC under parametric uncertainty (Van Laatum et al., 2026).

5. Conclusion

The main contribution of this paper is *trajectory-selection RL–MPC*, a framework designed to improve short-horizon MPC for greenhouse fruit production systems. It was shown that short-horizon MPC with a one-hour prediction horizon can incorporate economic information learned by an RL policy over a substantially longer horizon (approximately 16 h) by adding a terminal region constraint and terminal cost, derived from an RL rollout trajectory, and an online trajectory-selection mechanism. In addition, the finite-horizon MPC optimization refined the control input trajectories using short-term weather disturbances, improving performance relative to the guiding policy by reducing constraint violations and operating costs when evaluated on weather trajectories not encountered during policy training. The framework was evaluated using GreenLight, a nonlinear greenhouse tomato production model with stiff dynamics that is substantially larger than models used in previous greenhouse control studies.

The simulation results demonstrate that trajectory-selection RL–MPC with a one-hour prediction horizon matched the closed-loop performance of a specialist policy trained on a single weather trajectory. Under this setting, trajectory-selection RL–MPC outperformed standalone MPC with the same prediction horizon, indicating that the framework can mitigate the myopic behavior of short-horizon MPC. When training a generalist policy on a broader set of weather disturbances and evaluating controller performance on weather trajectories held out during training, trajectory-selection RL–MPC improved over both standalone RL (by $\sim$ 52%) and MPC (by $\sim$ 80%). Trajectory-selection RL–MPC achieved similar revenue from crop growth as the generalist policy, while reducing operating costs and constraint violations.

Future work should evaluate trajectory-selection RL–MPC over full production cycles, using crop models that represent fruit-development stages, to assess whether this method can balance current control decisions against longer-delayed economic returns. In addition, future studies should benchmark trajectory-selection RL–MPC against control methods with longer prediction horizons that capture slow crop yield dynamics, such as the discussed timescale decomposition approach, to quantify the trade-off between computational demand and closed-loop economic performance. Finally, practical greenhouse experiments should assess the robustness of this control method under real-world uncertain conditions, such as model mismatch, state estimation errors, and unseen weather and market-price disturbances.

CRediT authorship contribution statement

B. van Laatum: Writing – review & editing, Writing – original draft, Visualization, Software, Project administration, Methodology, Formal analysis, Conceptualization. **S. Msaad:** Writing – review & editing, Methodology, Conceptualization. **E.J. van Henten:** Writing – review & editing, Supervision. **R.D. McAllister:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **S. Boersma:** Writing – review & editing, Methodology, Conceptualization.

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.

The author is an Editorial Board Member/Editor-in-Chief/Associate Editor/Guest Editor for this journal and was not involved in the editorial review or the decision to publish this article.

Acknowledgments

This research received funding from the Sector Plan for Science and Technology (project number 3183019014), established by the Dutch Ministry of Education, Culture and Science.

Appendix A. Sensitivity to normalization vector

This simulation experiment evaluates the sensitivity of trajectory-selection RL–MPC to the choice of the normalization vector. The closed-loop performance of the presented framework using the normalization vector described in Section 2.6 was compared against RL–MPC using a single fixed normalization vector for each simulation run. For each weather trajectory in the generalist training dataset $\mathbf{d} \in D^{\text{train}}_{\text{generalist}}$, the generalist policy performed a rollout over that trajectory from the initial state using the greenhouse prediction model. The resulting state trajectory was then used to calculate the mean state vector:

$$\bar{x}^{(\mathbf{d})}_{\text{ref}} = \frac{1}{T+1} \sum_{k=0}^{T} x^{(j)}_{\text{ref}}(k). \tag{30}$$

Then the normalization vector was computed by taking the average over the resulting mean state vectors:

$$\bar{x}^{\text{fixed}}_{\text{ref}} = \frac{1}{|D^{\text{train}}_{\text{generalist}}|} \sum_{\mathbf{d} \in D^{\text{train}}_{\text{generalist}}} \bar{x}^{(\mathbf{d})}_{\text{ref}}. \tag{31}$$

Next, trajectory-selection RL–MPC was evaluated using this fixed normalization vector on the generalist test dataset $D^{\text{test}}_{\text{generalist}}$. The results for this simulation experiment are presented in Table 8.

Table 8
Sensitivity of trajectory-selection RL–MPC performance to choice of the normalization vector. The simulation-specific normalization vector is computed separately for each evaluation trajectory using (19), as described in Section 2.6. The fixed normalization vector is computed offline from the generalist training dataset using (31). At evaluation, trajectory-selection RL–MPC uses this fixed normalization vector during all simulation runs. The final row reports the paired difference between the simulation-specific and fixed normalization approaches, computed using (26).

Normalization	$\mathcal{J}$	$\mathcal{E}$	$\mathcal{P}$
Simulation-specific $\bar{x}_{\text{ref}}$	−0.041	0.104	0.144
Fixed normalization vector $\bar{x}_{\text{ref}}^{\text{fixed}}$	−0.046	0.106	0.152
Paired difference Δ	+0.005	−0.003	−0.007

Table 9
GreenLight model state variables with symbols and units.

State (description)	Symbol	Units
CO_2 variables		
CO_2 concentration of the main compartment	$x_{\text{CO}_2,\text{air}}$	mg m^{-3}
CO_2 concentration of the top compartment	$x_{\text{CO}_2,\text{top}}$	mg m^{-3}
Indoor temperature variables		
Greenhouse air temperature	$x_{\text{T,air}}$	°C
Air temperature above the screen (top air)	$x_{\text{T,top}}$	°C
Canopy temperature	$x_{\text{T,can}}$	°C
Internal cover temperature	$x_{\text{T,cov,in}}$	°C
External cover temperature	$x_{\text{T,cov,out}}$	°C
Thermal screen temperature	$x_{\text{T,thScr}}$	°C
Greenhouse floor temperature	$x_{\text{T,flr}}$	°C
Pipe temperature	$x_{\text{T,pipe}}$	°C
First soil layer temperature	$x_{\text{T,soil}_1}$	°C
Second soil layer temperature	$x_{\text{T,soil}_2}$	°C
Third soil layer temperature	$x_{\text{T,soil}_3}$	°C
Fourth soil layer temperature	$x_{\text{T,soil}_4}$	°C
Fifth soil layer temperature	$x_{\text{T,soil}_5}$	°C
Top lamp temperature	$x_{\text{T,lamp,top}}$	°C
Inter-lamp temperature	$x_{\text{T,lamp,inter}}$	°C
Grow pipe temperature	$x_{\text{T,growPipe}}$	°C
Blackout screen temperature	$x_{\text{T,blScr}}$	°C
Vapour pressure variables		
Vapour pressure in the main compartment	$x_{\text{VP,air}}$	Pa
Vapour pressure in the top compartment	$x_{\text{VP,top}}$	Pa
Crop variables		
Average crop canopy temperature over the last 24 h	$x_{\text{T,can,24h}}$	°C
Carbohydrates in the crop's buffer	x_{buf}	mg CH_2O m^{-2}
Carbohydrates in the leaves	x_{leaf}	mg CH_2O m^{-2}
Carbohydrates in the stem	x_{stem}	mg CH_2O m^{-2}
Carbohydrates in the fruits	x_{frt}	mg CH_2O m^{-2}
Crop temperature sum	$x_{\text{T,sum}}$	°C d^{-1}

The results show that trajectory-selection RL–MPC slightly benefits from using a simulation-specific normalization vector $+0.005 \pm 0.0056$. This difference is mainly due to a lower cumulative penalty for indoor climate constraint violations. When comparing trajectory-selection RL-MPC using the fixed normalization vector to standalone RL and MPC, the main results reported in Section 3.2 are preserved. RL–MPC improved the cumulative objective $\mathcal{J}$ by 48.9% relative to RL and by 76.8% relative to MPC, compared with 52.2% and 79.9% when using the simulation-specific normalization vector.

Appendix B. State variables Greenlight

See Table 9.

Data availability

The source code required to reproduce the results is publicly available at https://github.com/BartvLaatum/RL-MPC-GreenLight. Trained models and data for visualizations are available upon request.

References

Adams, S., Cave, C.R.J., Cockshul, K.E., 2001. Effect of temperature on the growth and development of tomato fruits. Ann. Botany 88 (5), 869–877. http://dx.doi.org/10.1006/anbo.2001.1524.

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M., 2019. CasADi: A software framework for nonlinear optimization and optimal control. Math. Program. Comput. 11 (1), 1–36. http://dx.doi.org/10.1007/s12532-018-0139-4.

Ariesen-Verschuur, N., Verdouw, C., Tekinerdogan, B., 2022. Digital Twins in greenhouse horticulture: A review. Comput. Electron. Agric. 199, 107183. http://dx.doi.org/10.1016/j.compag.2022.107183.

Baudoin, W., Nono-Womdim, R., Lutaladio, N., Hodder, A., Castilla, N., Leonardi, C., De Pascale, S., Qaryouti, M., Duffy, R., 2013. Good Agricultural Pratices for Greenhouse Vegetable Crops: Principles for Mediterranean Climate Areas 217. FAO, https://www.fao.org/4/i3284e/i3284e.pdf.

Bisbis, M.B., Gruda, N., Blanke, M., 2018. Potential impacts of climate change on vegetable production and product quality – A review. J. Clean. Prod. 170, 1602–1620. http://dx.doi.org/10.1016/j.jclepro.2017.09.224.

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

De Koning, A.N.M., 1994. Development and Dry Matter Distribution in Glasshouse Tomato: A Quantitative Approach (Ph.D. thesis). Agricultural University, Wageningen, The Netherlands, http://dx.doi.org/10.18174/205947.

Fanourakis, D., Tsaniklidis, G., Makraki, T., Nikoloudakis, N., Bartzanas, T., Sabatino, L., Fatnassi, H., Ntatsi, G., 2025. Climate change impacts on greenhouse horticulture in the mediterranean basin: Challenges and adaptation strategies. Plants 14 (21), 3390. http://dx.doi.org/10.3390/plants14213390.

Fink, M., Daniels, A., Qian, C., Velásquez, V.M., Salotra, S., Wollherr, D., 2023. Comparison of dynamic tomato growth models for optimal control in greenhouses. In: 2023 IEEE International Conference on Agrosystem Engineering, Technology & Applications. AGRETA, IEEE, Shah Alam, Malaysia, pp. 28–33. http://dx.doi.org/10.1109/AGRETA57740.2023.10262422.

Ghezzi, A., Reiter, R., Baumgärtner, K., Bemporad, A., Diehl, M., 2026. Rollout then optimize: A one-step Newton refinement of learned policies for nonlinear model predictive control. Preprint, URL arxiv.org/abs/2504.02710.

Katzin, D., Van Mourik, S., Kempkes, F., Van Henten, E.J., 2020. GreenLight – An open source model for greenhouses with supplemental lighting: Evaluation of heat requirements under LED and HPS lamps. Biosyst. Eng. 194, 61–81. http://dx.doi.org/10.1016/j.biosystemseng.2020.03.010.

Kearns, M., Singh, S., 2002. Near-optimal reinforcement learning in polynomial time. Mach. Learn. 49 (2), 209–232. http://dx.doi.org/10.1023/A:1017984413808.

Kirk, R., Zhang, A., Grefenstette, E., Rocktäschel, T., 2023. A survey of zero-shot generalisation in deep reinforcement learning. J. Artificial Intelligence Res. 76, 201–264. http://dx.doi.org/10.1613/jair.1.14174.

KNMI, 2020. Uurgegevens van het weer in nederland: station 240 (schiphol), 2011–2020. ZIP archive (ASCII). Open data provided under CC BY 4.0; (Accessed 2025). URL https://cdn.knmi.nl/knmi/map/page/klimatologie/gegevens/uurgegevens/uurgeg_240_2011-2020.zip.

Kuijpers, W.J.P., Antunes, D.J., Hemming, S., Van Henten, E.J., Van De Molengraft, M.J.G., 2021a. Fruit development modelling and performance analysis of automatic greenhouse control. Biosyst. Eng. 208, 300–318. http://dx.doi.org/10.1016/j.biosystemseng.2021.06.002.

Kuijpers, W.J.P., Katzin, D., Van Mourik, S., Antunes, D.J., Hemming, S., Van De Molengraft, M.J.G., 2021b. Lighting systems and strategies compared in an optimally controlled greenhouse. Biosyst. Eng. 202, 195–216. http://dx.doi.org/10.1016/j.biosystemseng.2020.12.006.

Mallick, S., Airaldi, F., Dabiri, A., Sun, C., De Schutter, B., 2025. Reinforcement learning-based model predictive control for greenhouse climate control. Smart Agric. Technol. 10, 100751. http://dx.doi.org/10.1016/j.atech.2024.100751.

Maree, S.C., Zhang, P., van Marrewijk, B.M., de Zwart, F., Bijlaard, M., Hemming, S., 2025. Autonomous greenhouse cultivation of dwarf tomato: Performance evaluation of intelligent algorithms for multiple-sensor feedback. Sensors 25 (14), http://dx.doi.org/10.3390/s25144321.

Mediratta, I., You, Q., Jiang, M., Raileanu, R., 2024. The generalization gap in offline reinforcement learning. Preprint, URL arXiv:2312.05742.

Morcego, B., Yin, W., Boersma, S., Van Henten, E., Puig, V., Sun, C., 2023. Reinforcement learning versus model predictive control on greenhouse climate control. Comput. Electron. Agric. 215, 108372. http://dx.doi.org/10.1016/j.compag.2023.108372.

Msaad, S., Harraway, M., McAllister, R.D., 2025. RL-guided MPC for autonomous greenhouse control. In: IFAC-PapersOnLine, vol. 59, pp. 449–454. http://dx.doi.org/10.1016/j.ifacol.2025.11.829.

Mulders, P.J.A.M., Van Den Heuvel, E.R., Van De Molengraft, M.J.G., Heemels, W.P.M.H., Reidsma, P., 2024. Extreme drought and rainfall had a large impact on potato production in the Netherlands between 2015 and 2020. Commun. Earth Environ. 5 (1), 496. http://dx.doi.org/10.1038/s43247-024-01658-3.

Petropoulou, A.S., van Marrewijk, B., de Zwart, F., Elings, A., Bijlaard, M., van Daalen, T., Jansen, G., Hemming, S., 2023. Lettuce production in intelligent greenhouses—3D imaging and computer vision for plant spacing decisions. Sensors 23 (6), http://dx.doi.org/10.3390/s23062929.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N., 2021. Stable-Baselines3: Reliable reinforcement learning implementations. J. Mach. Learn. Res. 22 (268), 1–8.

Ramírez-Arias, A., Rodríguez, F., Guzmán, J.L., Berenguel, M., 2012. Multiobjective hierarchical control architecture for greenhouse crop growth. Automatica 48 (3), 490–498. http://dx.doi.org/10.1016/j.automatica.2012.01.002.

Reiter, R., Ghezzi, A., Baumgärtner, K., Hoffmann, J., McAllister, R.D., Diehl, M., 2025. AC4MPC: Actor-critic reinforcement learning for guiding model predictive control. IEEE Trans. Control Syst. Technol..

Righini, I., van Marrewijk, B., Ntakos, G., Zhang, P., Jansen, G., Maree, S.C., Bijlaard, M., de Zwart, H.F., Hemming, S., 2024. 4Th autonomous greenhouse challenge: Pre-trial dwarf tomato measurements and images. http://dx.doi.org/10.4121/e1ee9de9-6ce9-4502-a37c-34b5b1372bed.v1, Open data provided under CC BY 4.0; (Accessed 2025).

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. Preprint, URL arXiv:1707.06347.

Seginer, I., Van Beveren, P.J.M., Van Straten, G., 2018. Day-to-night heat storage in greenhouses: 3 co-generation of heat and electricity (CHP). Biosyst. Eng. 172, 1–18. http://dx.doi.org/10.1016/j.biosystemseng.2018.05.006.

Serban, R., Hindmarsh, A.C., 2005. CVODES: the sensitivity-enabled ODE solver in SUNDIALS. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. 47438, pp. 257–269. http://dx.doi.org/10.1115/DETC2005-85597.

Silva, R.S., Kumar, L., Shabani, F., Picanço, M.C., 2017. Assessing the impact of global warming on worldwide open field tomato cultivation through CSIRO-Mk3·0 global climate model. J. Agric. Sci. 155 (3), 407–420. http://dx.doi.org/10.1017/S0021859616000654.

Sutton, R.S., Barto, A.G., 2018. Reinforcement learning: An introduction, second ed. In: Adaptive Computation and Machine Learning Series, The MIT Press, Cambridge, Massachusetts, USA.

Svensen, J.L., Cheng, X., Boersma, S., Sun, C., 2024. Chance-constrained stochastic MPC of greenhouse production systems with parametric uncertainty. Comput. Electron. Agric. 217, 108578. http://dx.doi.org/10.1016/j.compag.2023.108578.

Tap, F., 2000. Economics-Based Optimal Control of Greenhouse Tomato Crop Production (Ph.D. thesis). Wageningen University and Research, Wageningen, The Netherlands, http://dx.doi.org/10.18174/195235.

Van Henten, E.J., 1994. Greenhouse Climate Management : an Optimal Control Approach (Ph.D. thesis). Agricultural University, Wageningen, The Netherlands, http://dx.doi.org/10.18174/205106.

Van Henten, E.J., Bontsema, J., 2009. Time-scale decomposition of an optimal control problem in greenhouse climate management. Control Eng. Pract. 17 (1), 88–96. http://dx.doi.org/10.1016/j.conengprac.2008.05.008.

Van Laatum, B., Msaad, S., Van Henten, E.J., Mcallister, R.D., Boersma, S., 2026. Stochastic model predictive control with reinforcement learning for greenhouse production systems under parametric uncertainty. Control Eng. Pract. 169, 106787. http://dx.doi.org/10.1016/j.conengprac.2026.106787.

Van Laatum, B., Van Henten, E.J., Boersma, S., 2025. Greenlight-gym: Reinforcement learning benchmark environment for control of greenhouse production systems. In: 8th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture, AGRICONTROL 2025. IFAC, pp. 437–442. http://dx.doi.org/10.1016/j.ifacol.2025.11.827.

Van Ooteghem, R.J.C., 2007. Optimal Control Design for a Solar Greenhouse (Ph.D. thesis). Wageningen University, Wageningen, The Netherlands.

Van Straten, G., Van Henten, E.J., 2010. Optimal greenhouse cultivation control: Survey and perspectives. IFAC Proc. Vol. 43 (26), 18–33. http://dx.doi.org/10.3182/20101206-3-JP-3009.00004.

Van Straten, G., Van Willigenburg, G., Van Henten, E.J., Van Ooteghem, R., 2010. Optimal Control of Greenhouse Cultivation, zero ed. CRC Press, http://dx.doi.org/10.1201/b10321.

Vanthoor, B.H.E., De Visser, P.H.B., Stanghellini, C., Van Henten, E.J., 2011. A methodology for model-based greenhouse design: Part 2, description and validation of a tomato yield model. Biosyst. Eng. 110 (4), 378–395. http://dx.doi.org/10.1016/j.biosystemseng.2011.08.005.

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

Xu, D., Du, S., Van Willigenburg, G., 2018. Adaptive two time-scale receding horizon optimal control for greenhouse lettuce cultivation. Comput. Electron. Agric. 146, 93–103. http://dx.doi.org/10.1016/j.compag.2018.02.001.

Xu, D., Xu, L., Wang, S., Wang, M., Ma, J., Shi, C., 2025. Rule-based year-round model predictive control of greenhouse tomato cultivation: A simulation study. Inf. Process. Agric. 12 (3), 344–357. http://dx.doi.org/10.1016/j.inpa.2024.11.001.

Zhang, A., Ballas, N., Pineau, J., 2018. A dissection of overfitting and generalization in continuous reinforcement learning. Preprint, URL arXiv:1806.07937.

Zhang, C., Vinyals, O., Munos, R., Bengio, S., 2018. A study on overfitting in deep reinforcement learning. Preprint, URL arXiv:1804.06893.