Deep Reinforcement Learning Based Automatic Control in Semi-Closed Greenhouse Systems

Akshay Ajagekar. Fengqi You

Cornell University, Ithaca, New York, USA 14853

Abstract: This work proposes a novel deep reinforcement learning (DRL) based control framework for greenhouse climate control. This framework utilizes a neural network to approximate state-action value estimation. The neural network is trained by adopting a Q-learning based approach for experience collection and parameter updates. Continuous action spaces are effectively handled by the proposed approach by extracting optimal actions for a given greenhouse state from the neural network approximator through stochastic gradient ascent. Analytical gradients of the state-action value estimate are not required but can be computed effectively through backpropagation. We evaluate the performance of our DRL algorithm on a semi-closed greenhouse simulation located in New York City. The obtained computational results indicate that the proposed Q-learning based DRL framework yields higher cumulative returns. They also demonstrate that the proposed control technique consumes 61% lesser energy than deep deterministic policy gradient (DDPG) method.

Keywords: Deep reinforcement learning, Greenhouse, Climate control

1. INTRODUCTION

Growing demand for healthy and fresh food with the increasing global population is one of the pressing issues faced today. Reduction of agricultural land per capita by effects of climate change, reduced fresh water supply, and energy crisis can further exacerbate the shortage of fresh food (FAO et al., 2020). To this end, advantages offered by greenhouses like year-round vegetable and fruit crops production, minimal transportation costs, food safety control, and reduced water supply must be augmented. Controlled environment in greenhouses can yield a higher plant production (Tantau, 1990). Apart from higher investment and labour costs, expensive energy requirements and uncertainty in crop yield are some of the few challenges associated with indoor farming in greenhouses. Due to growth of worldwide greenhouse cultivation (McNutty, 2017), research and development of better controlled greenhouse environment with high energy efficiency for high utilization of plant yield is necessary. Better control of the greenhouse environment is required to ensure higher plant productivity and quality. Temperature, carbon dioxide, and humidity are some of the most important factors of greenhouse's indoor climate and should be carefully monitored and regulated in a controlled environment. To prevent damage to plants caused by undue heat stress or cold damage, indoor air temperature should be controlled within certain ranges. Extreme temperatures beyond plant's tolerance may inhibit growth spurt in plants leading to a significant decrease in crop yield (Ahamed et al., 2019).

Conventional greenhouse climate control techniques include nonlinear control based on feedback received from the greenhouse environment (Pasgianos et al., 2003). Uncertainty associated with temperature control in greenhouses can be effectively tackled with adaptive control (Sigrimis et al., 1999) and robust control (Linker et al., 1999) based techniques. Among these optimal control approaches for greenhouse control, model predictive control is a lucrative choice that utilizes future disturbance predictions to optimize future system behaviour under certain constraints (Blasco et al., 2007). However, such temperature control techniques make use of system model incorporating environmental disturbances and constraints that can be derived from first principle models like (Serale et al., 2018). Such first principle models cannot always accurately model complex phenomenon like crop growth and heat flow in complicated greenhouse structures.

In the absence of first principle models, artificial intelligence (AI) powered agricultural techniques are excellent candidates for realizing autonomous greenhouses (Hemming et al., 2019). Recently, utilizing machine learning for several agricultural practices has been on the rise (Kamilaris & Prenafeta-Boldú, 2018; Shang et al., 2020; Chen et al. 2021). Machine learning has been proved to be an effective tool for augment existing optimization and data analytics methods for various applications (Ning and You, 2019; Shang et al., 2019; Ajagekar, 2020; Sun, 2021) Reinforcement learning (RL) based control optimization has also been adopted in agriculture for tasks like irrigation and water management (Bhattacharya et al., 2003) and does not assume an explicit model of the system dynamics. Leveraging the autonomous decision making ability of RL and hierarchical feature learning offered by deep learning, AI-based control strategies for greenhouse climate control have been proposed (Wang et al., 2020). There are several research challenges associated with developing such deep reinforcement learning (DRL) based automatic control strategies. Greenhouse temperatures may often exhibit random behaviour owing to external environmental disturbances like outdoor air temperature, wind, snow, and others. The first challenge is to develop a DRL framework for greenhouse climate control that effectively adapts to time-varying outdoor weather conditions. Typically, complex control problems like greenhouse control, consist of continuous spaces making it difficult for conventional RL techniques to scale efficiently. A further challenge lies in ensuring that the DRL-based control technique can efficiently handle large and continuous state and action spaces, thereby overcoming the limitations of conventional RL. In addition, interacting with the greenhouse environment to learn an autonomous control strategy is expensive. Therefore, the final challenge lies in developing a climate control technique that learns quickly and is capable of yielding better returns within shorter times.

The objective of this work is to develop DRL-based control strategies for greenhouse climate control that effectively handles continuous action spaces. We perform state-action value function approximation using a neural network. Continuous actions serve as input to this neural network along with the state inputs. Training of this neural network approximator is performed by adopting a Q-learning based algorithm. Optimal actions for a particular state are extracted from the trained neural network by means of gradient ascent being performed on the neural network output or the value estimate. Gradients required for the value function maximization are obtained by backpropagation through the neural network. The applicability and efficiency of the proposed DRL-based technique is demonstrated with a simulation of a real-world greenhouse. The obtained control strategy is also compared with conventionally used DRL algorithm for continuous action spaces.

2. PRELIMINARIES

RL is a machine learning paradigm that deals with intelligent agents maximizing cumulative reward by taking corrective actions in an environment (Sutton & Barto, 2018). From an optimal control perspective, RL provides a model-free framework for solving problems stated as Markov decision processes (MDP). A general RL problem described as a MDP consists of a set of states S, set of actions A, reward function r, discount factor $\gamma \in [0,1]$, and transition dynamics. This problem can be formalized as a discrete time stochastic control process where an RL agent interacts with its environment at any timestep t by selecting an action $u_t \in A$ after receiving a state $x_t \in S$. This causes the agent to receive a reward r_t and environment state transitions to $x_{t+1} \in S$. Given a state, the RL agent selects an action to perform that is dictated by the control strategy termed as policy π. The policy provides a mapping from states to actions $\pi: S \times A \rightarrow [0,1]$. The goal of RL is to learn a optimal policy π^* that maximizes the expected return defined as cumulative discounted reward in Eq. (1). The cumulative discounted reward denoted as state-value function $V^\pi(s)$ is the expected return when starting in state x and following policy π subsequently and is also commonly referred to as the V-value function. The optimal expected return is governed by the optimal policy π^* and can be defined as shown in Eq. (2). Unavailability of transition dynamics in a typical RL problem allows us to construct state-action value $Q^\pi(x,u)$ defined in Eq. (3). The optimal policy in Eq. (4) can be obtained by optimizing state-action value or Q-value greedily at every state.

$$V^\pi(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid x, \pi\right] \tag{1}$$

$$V^*(x) = \max_{\pi} V^\pi(x) \quad \forall x \in S \tag{2}$$

$$Q^\pi(x,u) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid x, u, \pi\right] \tag{3}$$

$$\pi^*(x) = \arg\max_{u \in A} Q^\pi(x,u) \tag{4}$$

Use of neural networks to approximate either state value function $V^\pi(s)$, state-action value $Q^\pi(x,u)$, policy π, or the transition dynamics is referred to as DRL (Li, 2017). They can be parameterized by the weights of the deep neural network and is particularly important for scaling up prior state-of-the-art algorithms in RL to higher dimensional problems. The curse of dimensionality in complex RL problems can be efficiently dealt with DRL through representation learning (Bengio et al., 2013), unlike traditional tabular and nonparametric RL methods. The key differences between DRL and "shallow" RL is the choice of function approximators. Deep neural networks allow learning of low-dimensional feature representations and can act as powerful nonlinear function approximators (Arulkumaran et al., 2017). For example, convolutional neural networks can be used to learn representations from visual inputs like images or videos. Deep neural networks are well-suited for high dimensional inputs and do not require exponential amount of data during scale-up of state or action space. Such scaled up RL techniques based on deep neural networks allow for learning a wide variety of complex sequential decision-making tasks directly from high-dimensional inputs.

Some of the most popular value-based methods for DRL that aim to build a value function are based on Q-learning (Watkins & Dayan, 1992). Q-learning based methods like deep Q-network (Mnih et al., 2015) and double Q-learning (Hasselt et al., 2016) demonstrate exceptional performance with high dimensional sensory states and actions. An alternate class of DRL methods termed as policy gradient methods optimize a performance objective by finding a good policy. Deep deterministic policy gradient (DDPG) is one such DRL technique that extends the deep Q-network to continuous spaces (Lillicrap et al., 2015) and is commonly used to handle continuous action spaces.

3. MDP FORMULATION

Microclimate control system in greenhouse is operated to maintain a desired temperature, based on current indoor temperatures and outdoor environmental disturbances. The greenhouse climate control problem can be formulated as a MDP. Greenhouse indoor temperatures at next time step are predicted by the current states and environmental disturbances with a control input and are independent of the previous greenhouse states. Furthermore, we define the greenhouse climate control as a RL problem described as MDP with its five essential components.

In this work, greenhouse air temperature, wall temperature, ceiling temperature, and floor temperature are considered as states, following the physics-based model for greenhouse control (Chen and You, 2021, 2022). Maintaining the

$x_t \in \mathbb{R}$ at timestep t. The structure of the greenhouse climate control model is shown in Figure. 1.

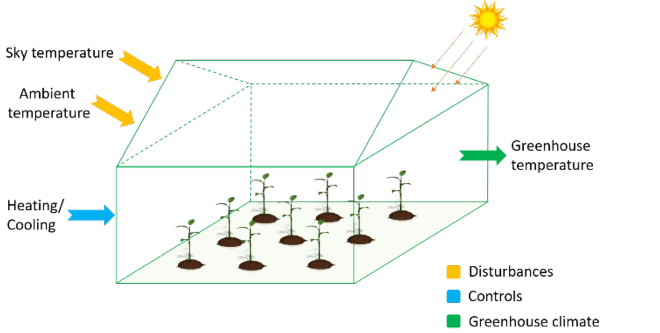

Figure. 1. Greenhouse model that shows disturbances, control actuators, and climate states

We consider a control system that provides or draws heat from the greenhouse. The control system can provide heating or cooling power to the greenhouse under certain operational constraints. Control inputs $u_t \in \mathbb{R}$ should be between the minimum and maximum allowable power values, which are $u_{\min} \le u_t \le u_{\max}$. For this particular RL problem, we consider a continuous action space so as to obtain better quality of optimal controls as compared to a discretized action space. The goal of the agent is to operate this control system to maintain the greenhouse temperature within a desired range, while minimizing the total energy cost or total energy use. The agent receives a reward $r_t \in \mathbb{R}$ from the greenhouse environment after taking an action u_t at state x_t, evolving the greenhouse states into a new state x_{t+1}. The reward r_t is calculated as shown in Eq. (5), where x_{t+1}^0 indicates the first element of the state vector that represents the greenhouse indoor temperature. The specified reward function includes the energy provided or drawn from the greenhouse and the penalty of temperature violation. This penalty is dependent on the deviation from the desired temperature range [T_{min}, T_{max}]. This reward function tries to balance the energy use minimization goal with that of maintaining the desired indoor greenhouse temperature. The discount factor γ that describes the importance of future rewards over immediate rewards is required to calculate cumulative returns over a fixed horizon. The transition dynamics in greenhouse is stochastic in nature because the greenhouse temperature is directly affected by environmental time-varying disturbances that may not be accurately forecasted.

$$r_t = -|u_t| - |x_{t+1}^0 - T_{\min}| - |T_{\max} - x_{t+1}^0| \qquad (5)$$

A greenhouse environment is simulated for climate control with Building Resistance-Capacitance Modelling (BRCM) MATLAB toolbox (Sturzenegger et al., 2014). This toolbox is built on previously validated modelling principles. Heat flow by conduction, convection and radiation is defined by the laws of thermodynamics with the greenhouse materials acting as resistances and capacitors affecting rate of heat flow. In this work, we simulate semi-closed greenhouse environment located in Brooklyn, New York, USA of dimensions 40m x 13m x 4m. Historical forecast data and measured weather data recorded in 2019 is used to train the RL agent. Validation of the RL agent's performance is conducted using forecast and measured temperature data for the year 2020. The sampling time of the greenhouse simulation is set to one hour. We also consider a crop growth scenario for year-round production of tomatoes. To this end, the greenhouse climate should be maintained within a specified range to ensure optimal crop growth and prevent damage caused by adverse climates. For an optimal growth rate of tomatoes, an indoor air temperature of 22-25°C should be maintained (Adams et al., 2001).

4. GREENHOUSE CLIMATE CONTROL

Real-world RL problems like greenhouse climate control comprise of large state and continuous action spaces. Value-based methods like Q-learning are typically adopted for such complex problems owing to their ability to solve harder problems. However, this involves discretizing the action space since such methods are not suitable for continuous action spaces. Previous works in literature for greenhouse control make use of policy gradient methods to tackle continuous action spaces (Wang et al., 2020). To overcome difficulties brought forth by policy gradient methods, we propose a DRL algorithm based on Q-learning for greenhouse control that can effectively tackle continuous action spaces by leveraging state-of-the-art optimization strategies based on gradient ascent.

4.1 Model Architecture

The first step towards constructing a DRL agent is to choose a nonlinear function approximator for the state-action value function $Q(x,u)$

$Q_\theta(x,u)$ where θ are the weights and biases of the neural network. The architecture of the neural network approximator is shown in Figure. 2. As seen in the figure, the greenhouse states and actions form the input to the neural network. Two fully connected hidden layers are used in this network to extract multiple layers of non-linear features from the time-varying states and actions. A rectified linear unit

(ReLU) is used as an activation function for the first fully connected layer. A linear activation is used for the remaining layers of the network. The obtained reward at each timestep cannot be greater than zero, as a result, linear activation is used specifically at the output layer. The output layer consists of a single neuron and approximates $Q_\theta(x,u)$. An estimate of the Q-value for a state and action pair is obtained by performing a forward pass through the neural network.

For a set of state and action inputs (x_k,u_k) and their respective optimal Q-value targets $Q^*(x_k,u_k)$, the neural network is trained to minimize the mean squared error between the target values and the predicted output as shown in Eq. (6). The neural network parameters θ are updated by performing gradient descent to minimize the loss function through backpropagation.

$$\min \frac{1}{K}\sum_{k=1}^{K}\left(Q_\theta(x_k,u_k)-Q^*(x_k,u_k)\right)^2 \tag{6}$$

Input data pre-processing is an important step of training such neural networks. As the range of values of greenhouse states and actions can vary substantially, we scale the state action inputs accordingly. For example, greenhouse temperatures can realistically vary from -10 to 30ºC, but heating or cooling power may vary from -10,000 to 10,000 Joules. To this end, we scale the greenhouse temperature states to the range [0, 1] with the maximum and minimum values determined from the weather forecasts. Similarly, we also scale the action input between [-1, 1]. Generating the training dataset with state action inputs along with computation of target optimal Q-values is described in the following section.

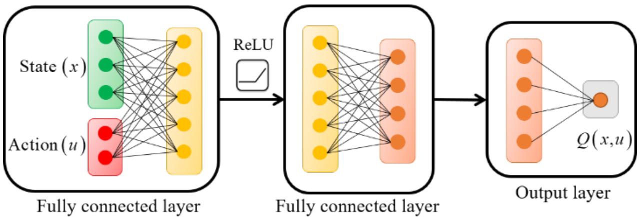

Figure. 2. Architecture of the Q-value function approximator

4.2 Q-learning based DRL Algorithm

The goal of the agent is to maximize the cumulative reward obtained by performing optimal actions. We can obtain optimal estimates of the Q-value by solving the Bellman equation written recursively in Eq. (7). Optimal Q-value estimates are obtained by following the Q-learning technique (Watkins & Dayan, 1992). The overall Q-learning based DRL algorithm for greenhouse climate control is presented in Figure. 3.

$$Q^*_\theta(x_t,u_t)=\mathbb{E}\left[r_{t+1}+\gamma\cdot Q^*_\theta(x_{t+1},u_{t+1})\right] \tag{7}$$

The agent learns by collecting experience in its memory *M*. The parametric Q-value function is initialized along with the exploration rate ε, discount factor γ, and a training batch size. The exploration rate defines the probability with which the agent randomly explores the greenhouse environment rather than exploiting the learned Q-value function. Initially, ε can be set to one and can be reduced gradually as the training progresses. Since, the greenhouse control task is not episodic we set a fixed length for each training episode. During the first few training episodes, the agent explores the greenhouse environment by taking random actions and observing the obtained reward and the next greenhouse states. A tuple (x,u,r,x') containing the current state, taken action, obtained reward, and observed next state are recorded in memory *M*. Cumulative returns for each episode being of fixed length are also recorded as episodic history. Once the agent's memory size exceeds the training batch size, the Q-value function network is trained as follows.

```
Q-learning based DRL Algorithm
Initialize
    Initialize Agent : $M$, $\theta$, $\varepsilon$, $\gamma$, batch size
Repeat (for each episode)
    Reset greenhouse environment states and cumulative returns $R$
    Repeat (for each step of episode)
        Choose action $u$ for current state $x$ using policy derived from $Q_\theta(x,u)$
        $\Pr\{u=\text{Gradient-ascent}(Q_\theta(x,u))\}=1-\varepsilon$   (Epsilon greedy approach)
        Apply optimal action $u^*$ and observe next state $x'$ and reward $r$
        Record $(x, u, r, x')$ in memory $M$
        $R \leftarrow R+r$
        if size($M$) exceeds batch size then
            $u'=\text{Gradient-ascent}(Q_\theta(x',u))$
            $Q^*(x,u)=r+\gamma\cdot Q_\theta(x',u')$
            $\theta \leftarrow \arg\min_\theta(Q_\theta(x,u)-Q^*(x,u))$
        end
    end
end
Define Gradient-Ascent$(Q_\theta(x,u))$
    Compute gradients : $\nabla_u Q_\theta(x,u)$
    Apply gradients to action $u$ to maximize Q-value: $u \leftarrow u+\eta\cdot\nabla_u Q_\theta(x,u)$
    Return $u^*=\arg\max_u Q_\theta(x,u)$
end
```

Figure. 3. Q-learning based DRL algorithm for greenhouse control

The training process primarily involves updating the Q-value network to yield better estimates of optimal Q-values for a state-action input pair. The Q-learning approach is used to update the Q-value estimates. For the new greenhouse state x', the optimal Q-value is computed and is used to update the Q-value estimate of the previous state as shown in the general Q-learning update rule in Eq. (8). However, for the neural network approximated Q-value function, the update rules involve training the neural network with updated state-action inputs and their corresponding target Q-values. The update rules for the neural network are given in Eq. (9). This training process is repeated for each new greenhouse state observed and the Q-value network is updated accordingly until the episode ends. During the next training episodes, the optimal actions are either randomly selected or extracted from the Q-value network using an epsilon-greedy strategy. As the

ε is linearly reduced by a decay factor with each episode.

$$Q(x,u) \leftarrow Q(x,u) + \alpha\left[r + \gamma \cdot \max_{u'} Q(x',u') - Q(x,u)\right] \qquad (8)$$

$$\theta \leftarrow \arg\min_{\theta}\left\{Q_\theta(x,u) - r - \gamma \cdot \max_{u'} Q_\theta(x',u')\right\} \qquad (9)$$

Typical Q-learning approaches are applied to environments with discrete action spaces making it easier to extract the optimal actions and Q-values for a particular state. However, for continuous action spaces computing $\max_{u} Q_\theta(x,u)$ and $\arg\max_{u} Q_\theta(x,u)$ for a parameterized Q-value neural network is a non-trivial task. As a result, we adopt a gradient ascent technique to maximize the Q-value for a fixed state. To achieve this, the action values can be driven towards an optimum by means of gradient ascent. Output gradients with respect to action inputs $\nabla_u Q_\theta(x,u)$ necessary to perform gradient ascent can be computed by performing a forward pass through the Q-value neural network and backpropagating through the network. Update rule for driving the variable action input towards an optimum is given in Eq. (10). The optimized actions are finally returned after a convergence criterion is met. The gradient update rules and convergence criterion may vary with the choice of optimization algorithms used like stochastic gradient descent, Adam, etc. However, the underlying idea behind optimizing the action inputs for a fixed state with the Q-value function neural network remains the same.

$$u \leftarrow u + \eta \cdot \nabla_u Q(x,u) \qquad (10)$$

Throughout the training process, the cumulative returns recorded at each episode can be used to track the progress of the training algorithm. Finally, the trained Q-value network can be used to extract optimal controls for any state by turning off agent's exploration and following the above gradient ascent based optimization process.

5. COMPUTATIONAL RESULTS

We evaluate the performance of the proposed Q-learning based DRL algorithm for greenhouse climate control with the simulated greenhouse located in Brooklyn, NY, USA. The DRL agent is initialized by specifying the state dimensionality as six corresponding to the greenhouse indoor temperature states, outdoor air temperature and temperature forecast. The discount factor γ and decay factor for exploration rate ε are set to 0.95 and 0.995 respectively. An empty set of maximum size 32 is set as the agent's memory *M.* Since the sampling time of the greenhouse simulation is one hour, we set the control horizon or the episode length to be 12 hours. Model architecture of the agent's Q-network neural network is set as follows. The input size coincides with the state and action space dimensions. The first and second fully connected layers in the model comprise of 32 and 16 neurons, respectively. Adam optimizer is used to train this neural network with the samples recorded in the DRL agent's memory. The value function network is constructed and trained with the Tensorflow deep learning library. In order to extract optimal actions from the neural network, maximization of the Q-value for a fixed state is performed by gradient ascent with the stochastic gradient ascent optimization technique.

For comparison purposes, we also implement a DRL control strategy based on DDPG (Wang et al., 2020). DDPG is an actor-critic algorithm and is an extension of deep Q-learning for continuous action spaces. In this technique we approximate both Q-value function $Q(x,u)$ and policy $\pi(x)$ with a neural network. For consistency sake, identical neural network architecture is used as nonlinear approximations of the Q-value function and the policy. Other parameters for this DRL technique are also kept consistent with those of the Q-learning based DRL algorithm for greenhouse control.

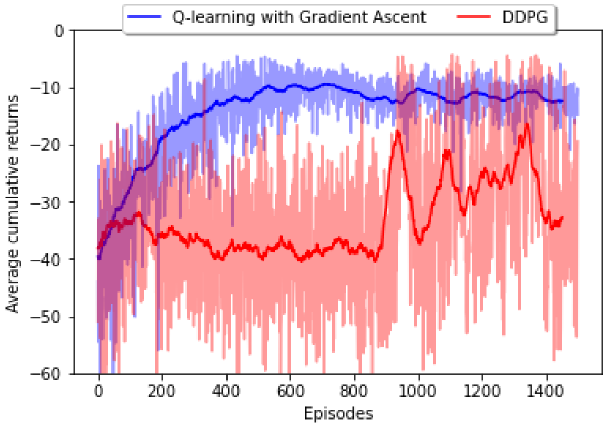

Figure. 4. Average cumulative returns obtained during training process for proposed Q-learning and the DDPG method.

Table 1. Computational results for the greenhouse climate control

	DDPG	Q-learning with gradient ascent
Training time (s)	2031.1 ± 33.5	2016.3 ± 381.1
Final returns	-21.06 ± 1.58	-12.52 ± 2.33
Energy use (kJ)	9800.26 ± 1160	3801.64 ± 950

DRL agents for both Q-learning and DDPG are trained for a series of 1500 episodes. Average cumulative returns per episodic length or horizon are recorded to track the training process. The cumulative returns obtained during their training for both DRL techniques are plotted in Figure. 4. Smoothed reward curves are generated to efficiently visualize the training progress. Based on the training plots it can be seen that the proposed Q-learning based DRL algorithm achieves high cumulative returns as compared to DDPG. To validate the performance of the trained Q-learning and DDPG agents, we simulate the greenhouse environment with weather and forecast data from January 2020 to June 2020. The computational results obtained for the training and evaluation phase of the DRL-based greenhouse control strategies are presented in Table. 1. Computational time required to train both DRL agents remain comparable. On the other hand, a significant difference is observed between obtained cumulative returns at each step of the training process. Similarly, final returns obtained at the end of 1500 episodes is much higher for the Q-learning based DRL algorithm as compared to DDPG. We also allow the trained agents to perform greenhouse climate control for six months in 2020 in order to evaluate energy usage. As seen in Table. 1, energy usage with Q-learning based DRL method is 61% lesser than that of energy use incurred with DDPG. Although a clear computational advantage in terms of training time cannot be seen for the Q-learning based DRL technique, the proposed DRL-based control strategy yields significantly higher returns accompanied by lower energy use for greenhouse control.

6. CONCLUSION

We proposed a Q-learning based DRL control framework for greenhouse climate control. We approximated the state-value function with a neural network which was trained with experience collected by the DRL agent. Extracting optimal actions from the trained neural network was performed with stochastic gradient ascent in order to handle continuous action spaces. Integration of Q-learning based training methodology with stochastic gradient ascent was achieved through the proposed DRL framework. The obtained computational results showed that the proposed Q-learning based DRL framework yields higher returns than DDPG. Evaluation of the trained agents on a simulation of a real-world greenhouse also resulted in substantial energy savings with the proposed Q-learning based DRL framework for greenhouse control.

REFERENCES

Adams, S. R., Cockshull, K. E., & Cave, C. R. J. (2001). Effect of Temperature on the Growth and Development of Tomato Fruits. *Annals of Botany*, *88*(5), 869-877.

Ahamed, M. S., Guo, H., Taylor, L., & Tanino, K. (2019). Heating demand and economic feasibility analysis for year-round vegetable production in Canadian Prairies greenhouses. *Information Processing in Agriculture*, *6*(1), 81-90.

Ajagekar, A., & You, F. (2020). Quantum computing assisted deep learning for fault detection and diagnosis in industrial process systems. *Computers & Chemical Engineering*, 143, 107119.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, *34*(6), 26-38.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis*, *35*(8), 1798-1828.

Bhattacharya, B., Lobbrecht, A. H., & Solomatine, D. P. (2003). Neural Networks and Reinforcement Learning in Control of Water Systems. *129*(6), 458-465.

Blasco, X., Martínez, M., Herrero, J. M., Ramos, C., & Sanchis, J. (2007). Model-based predictive control of greenhouse climate for reducing energy and water consumption. *Computers and Electronics in Agriculture*, *55*(1), 49-70.

Chen, W.-H., Shang, C., Zhu, S., et al. (2021). Data-driven robust model predictive control framework for stem water potential regulation and irrigation in water management. *Control Engineering Practice*, 113, 104841.

Chen, W.-H., & You, F. (2021). Smart greenhouse control under harsh climate conditions based on data-driven robust model predictive control with principal component analysis and kernel density estimation. *Journal of Process Control*, 107, 103-113.

Chen, W.-H., & You, F. (2022). Semiclosed Greenhouse Climate Control Under Uncertainty via Machine Learning and Data-Driven Robust Model Predictive Control. *IEEE Transactions on Control Systems Technology*, 30, 1186-1197.

FAO, IFAD, UNICEF, WFP, & WHO. (2020). The State of Food Security and Nutrition in the World (SOFI).

Hasselt, H. v., Guez, A., & Silver, D. (2016). *Deep reinforcement learning with double Q-Learning* Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona.

Hemming, S., de Zwart, F., Elings, A., Righini, I., & Petropoulou, A. (2019). Remote Control of Greenhouse Vegetable Production with Artificial Intelligence-Greenhouse Climate, Irrigation, and Crop Production. *Sensors (Basel, Switzerland)*, *19*(8), 1807.

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, *147*, 70-90.

Li, Y. (2017). Deep reinforcement learning: An overview.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Linker, R., Gutman, P. O., & Seginer, I. (1999). Robust controllers for simultaneous control of temperature and CO2 concentration in greenhouses. *Control Engineering Practice*, *7*(7), 851-862.

McNutty, J. (2017). Solar greenhouses generate electricity and grow crops at the same time, UC Santa Cruz study reveals. *USC Newscenter. Santa Cruz: University of California*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529-533.

Ning, C., & You, F. (2019). Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming. *Computers & Chemical Engineering*, 125, 434-448.

Ning, C., & You, F. (2021). Online learning based risk-averse stochastic MPC of constrained linear uncertain systems. *Automatica*, 125, 109402.

Pasgianos, G. D., Arvanitis, K. G., Polycarpou, P., & Sigrimis, N. (2003). A nonlinear feedback technique for greenhouse environmental control. *Computers and Electronics in Agriculture*, *40*(1), 153-177.

Serale, G., Fiorentini, M., Capozzoli, A., Bernardini, D., & Bemporad, A. J. E. (2018). Model predictive control (MPC) for enhancing building and HVAC system energy efficiency: Problem formulation, applications and opportunities. *11*(3), 631.

Shang, C., Chen, W.-H., Stroock, A. D., et al. (2020). Robust Model Predictive Control of Irrigation Systems With Active Uncertainty Learning and Data Analytics. *IEEE Transactions on Control Systems Technology*, 28, 1493-1504.

Shang, C., & You, F. (2019). Data Analytics and Machine Learning for Smart Process Manufacturing: Recent Advances and Perspectives in the Big Data Era. *Engineering*, 5, 1010-1016.

Sigrimis, N., Paraskevopoulos, P. N., Arvanitis, K. G., & Rerras, N. (1999). Adaptive temperature control in greenhouses based on multirate-output controllers. *IFAC Proceedings Volumes*, *32*(2), 3760-3765.

Sturzenegger, D., Gyalistras, D., Semeraro, V., Morari, M., & Smith, R. S. (2014, 4-6 June 2014). BRCM Matlab Toolbox: Model generation for model predictive building control. 2014 American Control Conference,

Sun, L., & You, F. (2021). Machine Learning and Data-Driven Techniques for the Control of Smart Power Generation Systems. *Engineering*, 7, 1239-1247.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Tantau, H. J. (1990). Automatic Control Application in Greenhouse. *IFAC Proceedings Volumes*, *23*(8, Part 6), 277-280.

Wang, L., He, X., & Luo, D. (2020, 9-11 Aug. 2020). Deep Reinforcement Learning for Greenhouse Climate Control. 2020 IEEE International Conference on Knowledge Graph (ICKG),

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3), 279-292.