Optimizing Irrigation Efficiency using Deep Reinforcement Learning in the Field

XIANZHONG DING, UC Merced, Merced, United States
WAN DU, UC Merced, Merced, United States

Agricultural irrigation is a significant contributor to freshwater consumption. However, the current irrigation systems used in the field are not efficient. They rely mainly on soil moisture sensors and the experience of growers but do not account for future soil moisture loss. Predicting soil moisture loss is challenging because it is influenced by numerous factors, including soil texture, weather conditions, and plant characteristics. This article proposes a solution to improve irrigation efficiency, which is called *DRLIC* (deep reinforcement learning for irrigation control). *DRLIC* is a sophisticated irrigation system that uses deep reinforcement learning (DRL) to optimize its performance. The system employs a neural network, known as the DRL control agent, which learns an optimal control policy that considers both the current soil moisture measurement and the future soil moisture loss. We introduce an irrigation reward function that enables our control agent to learn from previous experiences. However, there may be instances in which the output of our DRL control agent is unsafe, such as irrigating too much or too little. To avoid damaging the health of the plants, we implement a safety mechanism that employs a soil moisture predictor to estimate the performance of each action. If the predicted outcome is deemed unsafe, we perform a relatively conservative action instead. To demonstrate the real-world application of our approach, we develop an irrigation system that comprises sprinklers, sensing and control nodes, and a wireless network. We evaluate the performance of *DRLIC* by deploying it in a testbed consisting of six almond trees. During a 15-day in-field experiment, we compare the water consumption of *DRLIC* with a widely used irrigation scheme. Our results indicate that *DRLIC* outperforms the traditional irrigation method by achieving water savings of up to 9.52%.

CCS Concepts: • **Computing methodologies → Planning and scheduling**; **Machine learning;** • **Applied computing → Agriculture;** • **Computer systems organization → Sensors and actuators**;

Additional Key Words and Phrases: Irrigation control, reinforcement learning, monitoring, water resource optimization, experimentation, performance

ACM Reference Format:
Xianzhong Ding and Wan Du. 2024. Optimizing Irrigation Efficiency Using Deep Reinforcement Learning in the Field. *ACM Trans. Sensor Netw.* 20, 4, Article 99 (July 2024), 34 pages. https://doi.org/10.1145/3662182

This research is partially supported by the National Science Foundation under grant nos. #2008837 and 2239458, a UC Merced Fall 2023 Climate Action Seed Competition grant, a 2020 Seed Fund Award from CITRIS and the Banatao Institute at UC, and a 2022 Faculty Research Award through the Academic Senate Faculty Research Program at UC Merced.

Authors' Contact Information: Xianzhong Ding, UC Merced, Merced, California, United States; e-mails: xding5@ucmerced.edu; Wan Du (Corresponding author), UC Merced, Merced, California, United States; e-mails: wdu3@ucmerced.edu.

ACM 1550-4859/2024/07-ART99
https://doi.org/10.1145/3662182

1 INTRODUCTION

Agriculture is a major contributor to the consumption of ground and surface water in the United States, with estimates suggesting that it accounts for approximately 80% of the nation's water use and over 90% in many Western states.[1] Specifically, California's almond acreage in 2019 was estimated at 1,530,000 acres, and almond irrigation alone is estimated to consume roughly 195.26 billion gallons of water annually [24, 42]. Given the current drought affecting many Western states, it is critical to improve irrigation efficiency to conserve our limited freshwater reserves. This study focuses on enhancing the irrigation efficiency of almond orchards.

The primary objective of agricultural irrigation is to maintain the health of trees and maximize crop production. Achieving this goal requires maintaining the soil moisture of the trees within a specific range, typically between the **Field Capacity (FC)** level and the **Management Allowable Depletion (MAD)** level. If the soil moisture falls below the MAD level, almond trees may experience discoloration or even die. Conversely, if the soil moisture exceeds the FC level, it can lead to reduced oxygen movement in the soil, negatively impacting the tree's ability to absorb water and nutrients. Both FC and MAD levels are dependent on the soil type and plant species. Therefore, to determine the appropriate FC and MAD levels for a specific orchard, it is essential to identify the soil type and refer to a manual that outlines the corresponding FC and MAD levels for that soil type [41].

To ensure that the soil moisture remains within the MAD and FC range, the sprinklers must be activated every day or every few days, depending on the soil moisture level. Given the high evaporation rate in California, daily irrigation is recommended by the Almond Board of California [41] and is used in many existing irrigation systems [18, 25]. In most micro-sprinkler irrigation systems, irrigation is performed at night to reduce water loss due to evaporation, which can be as high as 14% to 19% during the day [53]. The irrigation control problem involves determining the appropriate amount of water to be applied to each sprinkler to ensure that the soil moisture remains within the MAD and FC range until the next irrigation cycle. This decision is based on the current soil moisture level and the predicted soil moisture loss for the following day, which is influenced by factors such as soil type, local weather conditions, and plant properties (e.g., root length and leaf number). The objective of irrigation is to provide trees with an appropriate amount of water such that the soil moisture remains above the MAD level until the next irrigation cycle.

Developing optimal irrigation control strategies requires accurate soil moisture loss prediction models. Traditional **Model Predictive Control (MPC)** methods can be utilized for optimal irrigation control if such a prediction model exists, but the accuracy of the model can have a significant impact on the performance of these methods [15, 37]. Obtaining an accurate soil moisture prediction model for an almond orchard is challenging, as soil moisture is influenced by multiple factors, including soil type, topography, ambient temperature, humidity, solar radiation intensity, and plant transpiration [50]. Additionally, customized soil moisture models must be developed for each orchard, limiting the scalability of MPC-based approaches. These two limitations have prevented the use of MPC-based methods in orchards.

Orchard irrigation systems currently employ two primary control methods: evapotranspiration (ET)-based and sensor-based approaches. The ET-based method relies on estimating ET, which accounts for soil moisture loss due to various weather factors such as wind, temperature, humidity, and solar irradiance. These weather parameters are measured by weather stations, with local ET values being publicly available [4] and updated hourly. ET-based irrigation controllers utilize the ET values since the previous irrigation cycle to determine when to activate sprinklers and

[1] Irrigation and Water Use: https://www.ers.usda.gov/

compensate for soil moisture depletion. However, this approach fails to consider the anticipated soil moisture loss for the following day before the subsequent irrigation cycle. Consequently, ET-based irrigation may result in either under-irrigation or over-irrigation if the soil moisture loss of the preceding day differs from that expected for the next day. To mitigate this issue, ET-based methods usually incorporate a safety margin of water [56], leading to over-irrigation in most instances [23, 25].

By utilizing precise soil moisture sensors, irrigation controllers have the capability to respond directly to the current soil moisture levels [25]. These controllers typically operate on a "rule-based" approach, in which they supply a specific amount of water when a deficiency in soil moisture is detected. However, the parameters governing the timing and quantity of irrigation are generally fine-tuned by growers based on their experience. Unfortunately, sensor-based irrigation often lacks the ability to systematically consider future weather conditions, such as rainfall and wind, for the following day. This absence of predictive capability regarding water loss hinders the effectiveness of sensor-based irrigation systems.

In order to address the limitations of the existing irrigation schemes mentioned above, we develop *DRLIC***(deep reinforcement learning for irrigation control)**, a practical irrigation system based on **Deep Reinforcement Learning (DRL)**. *DRLIC* is capable of autonomously acquiring an optimal irrigation control policy by exploring various control actions. Within the *DRLIC* framework, a control agent carefully observes the environmental state and selects an action based on a predefined control policy. Upon implementing the chosen action, the environment transitions to a subsequent state, and the agent receives a reward associated with its action. The primary objective of the learning process is to maximize the anticipated cumulative discounted reward. *DRLIC* employs a neural network as the control agent, utilizing it to learn and establish the control policy. The neural network effectively maps "raw" observations to the irrigation decision for the subsequent day. The state includes relevant weather information, such as ET and precipitation, for both the current day and the following day.

To minimize water consumption in irrigation while simultaneously safeguarding the health of the trees, we design a reward function that accounts for three distinct scenarios. If the soil moisture level surpasses the FC threshold or falls below the MAD threshold, a negative reward is assigned to the control agent. Conversely, if the soil moisture level remains within the MAD and FC range, a positive reward is granted, with its magnitude inversely proportional to the amount of water consumed.

To achieve optimal training results for *DRLIC*'s control agent, it would be ideal to conduct the training in a real almond orchard. However, due to the long irrigation interval of 1 day in our specific case, the control agent would only have the opportunity to explore 365 control actions per year. This limited exploration would significantly prolong the training process, requiring approximately 384 years to train a control agent that reaches convergence. To overcome this challenge and expedite the training process, we build a customized soil-water simulator. This simulator is calibrated using the 2-month soil moisture data collected from 6 almond trees. By utilizing this simulator, we can generate a sufficient amount of training data for *DRLIC* using 10-year weather data.

When deployed as an irrigation controller in the field, the control agent may encounter states that were not encountered during training, particularly in the case of agents trained in a simulated environment. In such scenarios, the control agent may make ill-advised decisions that pose risks to the plants, such as driving the soil moisture level below the MAD threshold or exceeding the FC threshold. To bridge the gap between the simulated environment and the real orchard, we design a robust irrigation mechanism. Instead of executing unwise actions generated by *DRLIC*'s control agent, we opt to utilize the ET-based method to generate alternative actions. We then leverage the soil moisture model from our soil-water simulator to validate the safety of the chosen actions.

In order to assess the effectiveness of *DRLIC*, we construct an irrigation experimental setup utilizing micro-sprinklers that are presently employed in almond orchards. Two raised beds accommodate a total of six almond trees. Every individual tree is equipped with a sensing and control node, which consists of a micro-sprinkler capable of independent control and a soil moisture measurement system comprising two sensors deployed at varying depths in the soil. Each node has the capability to transmit its sensing data to our server using IEEE 802.15.4 wireless transmission, and it can receive irrigation commands from the server.

Our testbed has been successfully deployed in the field, where we have collected soil moisture data from six sensing and control nodes for over 3 months. Using 2 months of this data, we train our soil moisture simulator, whereas 2 weeks of data were used to validate its accuracy. Once we train the control agent of *DRLIC*, we deploy it in our testbed for 15 days. The results of the experiment show that *DRLIC* can reduce water usage by 9.52% compared with the ET-based control method, without causing any harm to the health of almond trees.

We summarize the main contributions of this article as follows:

- We design *DRLIC*, an irrigation method that utilizes DRL to save water usage in agriculture.
- A variety of approaches have been proposed to convert *DRLIC* into a practical irrigation system, featuring our tailor-made design of DRL states and rewards to optimize irrigation, a validated soil moisture simulator for expedited DRL training, and a reliable irrigation module that prioritizes safety.
- We build an irrigation testbed with customized sensing and actuation nodes, and six almond trees.
- Extensive experiments in our testbed show the effectiveness of *DRLIC*.

2 IRRIGATION PROBLEM

Soil Water Content Parameters. Soil plays a critical role in the water supply for plants, serving as a reservoir for their hydration needs. Typically, up to 35% of the space in soil can be filled with water. Soil water content is a measure of the amount of water present in the soil, often expressed as a percentage of water by volume (%) or as inches of water per foot of root (in/ft). Soil moisture sensors are commonly employed to measure the soil water content at a specific location in the soil. For trees with roots that extend several feet, multiple soil moisture sensors may be utilized at various depths along the root structure. The root is typically divided into a defined number of sections, with a soil moisture sensor positioned at the midpoint of each section to enable accurate monitoring of the plant's hydration levels.

To calculate the soil water content of a tree, we use the equation $V = \sum_{j=1}^{M} \varphi_j * d_j$, where M represents the number of moisture sensors installed at different depths (in our experiments, M is 2). Each φ_j refers to the reading recorded by a specific soil moisture sensor, whereas d_j indicates the depth covered by that sensor. When these soil moisture sensors are employed to measure the soil water content within a region, they will be placed beneath a typical tree that shares a similar soil water content with most of the other trees in that area.

To ensure the health of a plant, it is essential to ensure that its roots have access to a sufficient supply of water. Figure 1 provides an illustration of two critical soil water content levels that are crucial for plant health [1]. Firstly, the **Permanent Wilting Point (PWP)** represents the minimum threshold of soil water content below which plants are unable to draw sufficient moisture from the soil. Prolonged periods of soil moisture levels below the PWP can lead to plant wilting or death. Secondly, if the soil water content exceeds the FC level, there is an excess of water in the

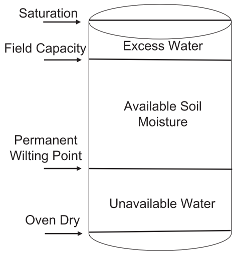


Fig. 1. The various levels of the soil water content [1].

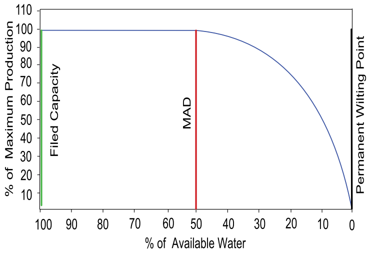


Fig. 2. How plant production (growth) is affected by soil water content [45].

Table 1. Suggested MAD For Different Crops [45]

Crop	MAD (%)	Crop	MAD (%)
Beans	40	Potatoes	30
Blueberries	50	Raspberries	50
Corn	50	Strawberries	50
Alfalfa	55	Sweet Corn	40
Mint	35	Tree Fruit	50

soil, which can lead to water wastage and rotting of the roots over time, ultimately compromising the plant's health. Therefore, the goal of irrigation systems is to ensure that the soil water content remains within the desired range, specifically between the PWP and the FC levels.

The primary objective of irrigation for fruit trees like almonds is to achieve maximum production. To achieve this goal, it is crucial to maintain soil moisture content above the MAD level instead of the PWP level. As shown in Table 1, the MAD level can vary for different types of crops (e.g., 40% for beans, 50% for fruit trees). Figure 2 illustrates the relationship between soil moisture content and almond tree production [45]. From Figure 2, it can be observed that the MAD level for almond trees is the median value (50%) between the FC level and the PWP level. Therefore, To ensure the optimum production of almond trees, it is crucial to maintain the soil water content above the threshold of the MAD level.

How to Determine These Parameters in an Orchard. The **Available Water-holding Capacity (AWC)** of the soil is defined as the soil water content range between the FC level and the PWP level. Figure 3 shows that the AWC varies for different soil types [3]. The texture, presence, and abundance of rock fragments, as well as the depth and layers of soil, can affect the AWC of the soil. Finer-textured soils, such as loam, have a higher AWC than sandier soils [3]. On the other hand, soils with more clay, such as clay loam, have a lower AWC than loamy soils [3].

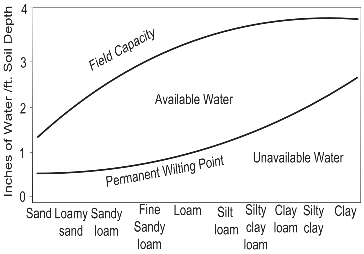


Fig. 3. Relationship between available water capacity and soil texture [3].

The tree's AWC, represented by V_{awc}, can be determined using the equation $V_{awc} = \sigma_{awc} * D_{foot}$. Here, σ_{awc} denotes the soil's AWC and D_{foot} refers to the tree's root depth in feet. Reference values for the AWC of various soil types, denoted as σ_{awc}, can be found in [3]. Similarly, the PWP level for a given soil type, denoted as V_{pwp}, can be calculated using the equation $V_{pwp} = \varphi_{pwp} * D_{inch}$. Here, φ_{pwp} represents the soil moisture content at the wilting point of the specific soil type, whereas D_{inch} corresponds to the root depth of the plant in inches. Specific values of φ_{pwp} for different soil types can be found in [3]. Using the above parameters (V_{awc} and V_{pwp}), we can determine the FC level as $V_{fc} = V_{awc} + V_{pwp}$ and the MAD level as $V_{mad} = \alpha * V_{awc} + V_{pwp}$. Here, α is set to 50% for almond trees.

How to Use These Parameters for Irrigation. The primary goal of irrigation is to effectively manage the soil water content of plants, ensuring that it remains within the desired range defined by the FC level and the MAD level. To achieve this goal, it is crucial to determine the soil's AWC and the permanent wilting point level (V_{awc} and V_{pwp}), which depend on the soil type, texture, and layers. Once we identify the soil type, we can calculate these parameters using the methods discussed earlier. However, in large orchards with varying soil types and changing parameters, it is essential to adjust the irrigation system's setting accordingly.

How Many Valves to Control in an Orchard. In an ideal scenario, achieving optimal irrigation in an orchard necessitates individual control of sprinklers for each tree. This stems from the fact that the ET rates vary between 0.12 to 0.20 inches among different trees within the orchard [40]. Furthermore, the soil type exhibits spatial variations within the orchard [41]. For instance, in a 60-acre California[2] orchard, there are 10 soil types with clay loam constituting 45.6% to 54.7% and slopes ranging from 0% to 8%. However, considering the high density of almond trees, with approximately 75 to 125 trees per acre, the deployment of soil moisture sensors under each tree becomes cost-prohibitive. Thus, orchards are typically divided into multiple irrigation regions based on similarities in soil texture. Each irrigation region is equipped with a valve that controls the sprinklers within that specific region. The irrigation management of a large orchard revolves around controlling a significant number of valves. It is important to note that the focus of this article is primarily on irrigation control rather than field partitioning. One practical approach to partitioning an orchard into distinct irrigation regions involves conducting soil sampling across the entire orchard using an auger. This soil sampling process serves multiple purposes for growers, including tree density planning and optimizing fertilization strategies.

3 *DRLIC* System Design

In this section, we provide a comprehensive overview of *DRLIC*, starting with its fundamental principles. We approach the irrigation problem by modeling it as a Markov Decision Process,

[2] Soil Map: https://casoilresource.lawr.ucdavis.edu/gmap/

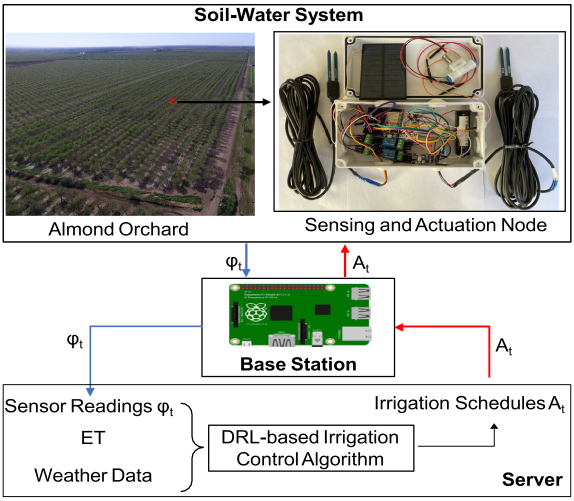

Soil-Water System
Almond Orchard
Sensing and Actuation Node
φ_t
A_t
Base Station
φ_t
A_t
Sensor Readings φ_t
ET
Weather Data
DRL-based Irrigation Control Algorithm
Irrigation Schedules A_t
Server

Fig. 4. *DRLIC* system architecture.

allowing us to analyze it within a well-defined framework. We present our innovative approach to address this challenge, which involves the design of a DRL-based irrigation scheme. In addition, we introduce a safe irrigation module designed to ensure the protection of the irrigation process.

3.1 Overview

Figure 4 illustrates the comprehensive system architecture of *DRLIC*, comprising two pivotal components: (1) a wireless network consisting of sensing and actuation sprinkler nodes and (2) a cutting-edge DRL-based control algorithm. For almond orchards, we strategically deploy a sensing and actuation node in each irrigation region. These nodes are equipped with an array of soil moisture sensors placed at various soil depths. To facilitate data transmission, sensing data are wirelessly conveyed to the base station through an IEEE 802.15.4 network. The *Base Station* gathers data from all *DRLIC* nodes and transmits them to a local server via Wi-Fi. This collective dataset of sensing data from *DRLIC* nodes provides a comprehensive “snapshot” of soil moisture readings φ_t encompassing the entire orchard.

On the server side, the DRL-based irrigation control agent leverages the soil moisture sensor readings, ET data, and weather information obtained from local weather stations to make informed irrigation decisions. It diligently generates optimal irrigation schedules for all *DRLIC* nodes, with the overarching goal of minimizing overall irrigation water consumption while ensuring the well-being of almond trees. Subsequently, the server transmits the generated irrigation schedules A_t to all *DRLIC* nodes. Upon receiving these commands, each node promptly initiates the opening of its sprinkler through a latching solenoid equipped with two relays. Further details pertaining to the implementation specifics of the nodes will be provided in Section 4.

3.2 MDP and DRL for Irrigation

We adopt a daily irrigation scheme wherein irrigation initiates at 11 PM every day to ensure optimal water management in the orchard. The controller’s primary objective is to determine the duration for which each sprinkler should be open during each irrigation cycle. This decision is crucial to maintain the soil water content within the desired range of MAD and FC by the following night. To make informed irrigation decisions, the controller takes into account various factors that influence the future soil water content. These factors include the current soil water content, the volume of water applied during irrigation, the trees' water absorption capacity, and the soil water losses caused by factors such as runoff, percolation, and ET.

To model this irrigation problem mathematically, we formulate it as a **Markov Decision Process (MDP)**, denoted by the tuple $<S, A, T, R>$:

- S represents a finite set of states that encompasses the sensed moisture levels obtained from the orchard and the weather data collected from the local weather station.
- A corresponds to a finite set of irrigation actions available for controlling the valves.
- T defines the state transition function $T: S \times A \longrightarrow S$, which determines the soil water content at the next timestep based on the current soil water content and the chosen irrigation action.
- R represents the reward function $R: S \times A \longrightarrow R$, which quantifies the performance of a specific control action in terms of achieving water conservation goals and maintaining optimal almond tree health.

Given the MDP formulation, our objective is to find an optimal control policy $\pi(s)^* : S \longrightarrow A$ that maximizes the accumulative reward R over time. However, due to the complex nature of the state transition function, conventional techniques such as dynamic programming are not suitable for finding the optimal control policy. Therefore, in this study, we adopt a DRL-based approach to develop irrigation control algorithms.

Unlike traditional approaches that rely on predefined rules in heuristic algorithms, our DRL-based approach learns the irrigation policy from observations. By leveraging machine learning techniques, the irrigation control algorithm becomes adaptive, continually refining its decision-making process based on real-time data and environmental conditions. This data-driven approach enables us to optimize water usage and ensure the health and productivity of almond trees in the orchard.

DRL is a powerful data-driven learning method that has gained widespread recognition and has been successfully employed in numerous control applications [9, 10, 17, 34, 36, 63]. DRL learns an optimal control policy by iteratively interacting with the environment, enabling agents to make informed decisions.

At each timestep t, the control agent, based on its policy π_θ, selects an action $A_t = a$ given the current state $S_t = s$:

$$a \sim \pi_\theta(a|s) = \mathbb{P}(A_t | S_t = s; \theta). \tag{1}$$

In DRL, the control policy is approximated using a neural network that is parameterized by θ. When the control agent takes action a, a state transition $S_{t+1} = s'$ occurs according to the system dynamics f_θ (Equation (2)), and the agent receives a reward $R_{t+1} = r$.

$$s' \sim f_\theta(s, a) = \mathbb{P}(S_{t+1} | S_t = s, A_t = a). \tag{2}$$

The objective of DRL is to determine the optimal set of parameters θ^* that maximizes the expected reward:

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{\pi_\theta}[r]. \tag{3}$$

One advantageous characteristic of DRL is that, due to the Markov property, both the reward and state transition solely depend on the previous state. Consequently, DRL seeks to discover a policy π_θ that maximizes the expected reward (Equation (3)). By iteratively updating the parameters θ based on observed experiences, the DRL agent progressively learns an effective control

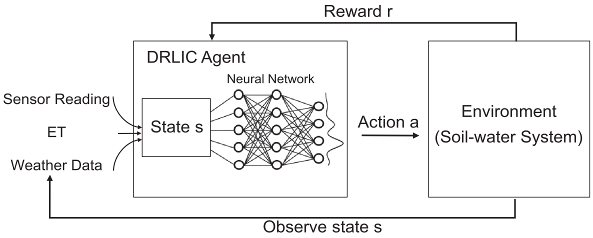


Fig. 5. Deep reinforcement learning in *DRLIC*.

policy, which can be specifically tailored for tasks such as irrigation control, enabling efficient and optimized decision-making.

There are several compelling reasons for employing DRL in irrigation control, as follows.

(1) **Data-Driven Optimal Control:** DRL offers a powerful approach for learning an optimal irrigation control policy directly from data, eliminating the need for pre-programmed control rules [20, 21] or explicit assumptions about the soil-water environment.
(2) **Utilizing Domain Knowledge:** DRL has the ability to leverage domain knowledge in training the irrigation control agent, which is implemented as a neural network. By incorporating domain knowledge into the training process, we can guide the agent's learning without relying on labeled data.
(3) **Generalization Capability:** The neural network used in DRL exhibits strong generalization abilities. This is particularly advantageous in irrigation control, as it enables the control agent to handle dynamically varying weather conditions and ET data. The agent can learn patterns and correlations from past experiences and apply them to new situations, improving its adaptability to changing environmental factors.

3.3 Deep Reinforcement Learning in *DRLIC*

The DRL architecture of *DRLIC* is summarized in Figure 5. The core component of this architecture is the *DRLIC* agent, which derives the irrigation control policy through training a neural network. The agent receives a comprehensive set of information as input, including the current soil water content, today's weather data (such as ET and precipitation), and the predicted weather data for tomorrow. Leveraging this input, the agent generates the optimal action, determining the appropriate amount of water to be irrigated.

Following the agent's decision, the resulting soil water content is observed and fed back to the agent the next day at 11 PM. This observed soil water content is then utilized to calculate a reward, which serves as a feedback signal for the agent's performance evaluation. The agent leverages this reward to update the parameters of the neural network, continuously improving its ability to control irrigation effectively.

In summary, the *DRLIC* architecture operates in a cyclical process, where the agent receives input, makes irrigation decisions, observes the outcomes, and updates its neural network parameters based on the received reward. This iterative learning process enables the agent to adapt and optimize its irrigation control policy over time. In the subsequent sections, we will provide further insights into the design and functionality of each component within the *DRLIC* framework.

3.3.1 State in DRLIC. State in *DRLIC* plays a crucial role in our irrigation MDP model, encompassing three key aspects: sensed state, weather-related state, and time-related state.

Sensed State: The sensed state represents the soil water content measured by *DRLIC* nodes. This information is obtained through Equation (6), which incorporates the sensor readings (φ) obtained from the *DRLIC* nodes. By monitoring the soil moisture levels, we can effectively assess the irrigation needs of each irrigation region.

Weather-Related State: The weather-related state is a vector that comprises both current and predicted weather variables obtained from the local weather station. This vector includes parameters such as ET and precipitation (in inches); maximum, average, and minimum temperature (in degrees Fahrenheit); maximum, average, and minimum humidity (in percentage); average solar radiation (in Ly/day); average wind speed (in mph); predicted ET using Equation (16) (in inches); and forecasted precipitation (in inches). By considering these weather variables, we can account for the dynamic nature of weather conditions in our irrigation control decisions.

Time-Related State: The time-related state focuses on the date and includes information about the month. Certain aspects of the state can change over time, such as plant water requirements and the weather-related state mentioned earlier. Plant water requirements vary depending on the growth stage of the plants, whereas weather conditions exhibit seasonal variations.

3.3.2 *Action in* DRLIC.

The main goal of our irrigation control is to optimize plant health or maximize production with minimum water consumption. To achieve this, the agent outputs a vector action that contains the amount of water to irrigate for each irrigation region in an orchard. This action is typically continuous and can take any value between 0 and the maximum water capacity of the irrigation system. To ensure that the action is feasible, we clip it to be within this range. Once we have the irrigation amount, we convert it to the open **time duration (td)** td_i for each ith micro-sprinkler. The duration is calculated as $td_i = a_i/I$, where I is the irrigation rate. For our testbed, we set I to 0.018 inch/min, which is the rate of the micro-sprinklers we used. The duration is then rounded up to the nearest multiple of the irrigation interval (15 minutes), which is the shortest time interval that can be controlled by the irrigation system.

3.3.3 *Reward in* DRLIC.

In order to express our objective of achieving good plant health with minimum water consumption, we define a reward function that incorporates both factors. As discussed in Section 2, maintaining the soil water content between the MAD and FC levels is crucial for maximizing almond tree production. Therefore, we use the deviation of the soil water content from these levels as a proxy for plant health. Additionally, we consider the control action as a proxy for water consumption.

To balance the relative importance of plant health and water consumption in the reward function, we introduce the hyperparameters λ and μ. These hyperparameters allow us to adjust the contribution of each factor and fine-tune the trade-off between achieving good plant health and minimizing water consumption. By carefully selecting the values of λ and μ, we can emphasize one factor over the other based on our specific objectives and priorities.

To ensure minimal water consumption without compromising plant health, our reward function accounts for three scenarios, as depicted in Equation (5). First, when the soil water content (V_i) of the ith irrigation region exceeds the FC (V_{fc}) level, it signifies that the plants are receiving more water than necessary. Over-irrigation can adversely affect plant health and result in excessive water consumption. Thus, in this situation, penalties are imposed on both plant health and water consumption. Second, when V_i falls within the range of V_{fc} and V_{mad}, the plants are deemed to be in good health. To conserve water, our objective is to maintain V_i close to V_{mad}. As a result, the reward is inversely proportional to the amount of water consumed. Third, when V_i is below V_{mad}, the plants experience water stress, severely impacting their health. The extent of this impact is proportional to the difference between V_i and V_{mad}.

Table 2. Parameter Setting in Reward

Parameter	Value	Parameter	Value
λ_1	3	α	50 (%)
μ_1	8	D_{inch}, D_{foot}	23.62 inches, 1.97 (ft)
μ_2	3	d	11.81 (in)
λ_3	10	φ_{pwp}	10 (%)
μ_3	1	σ_{awc}	2.4 (in/ft)

By considering these three scenarios, our reward function incorporates the trade-off between plant health and water consumption, enabling the DRL agent to strike an optimal balance for efficient irrigation management. Our reward function is defined as follows:

$$R = -\sum_{i=1}^{N} R_i \tag{4}$$

$$R_i = \begin{cases} \lambda_1 * (V_i - V_{fc}) + \mu_1 * a_i, & V_i > V_{fc} \\ \mu_2 * a_i, & V_{fc} > V_i > V_{mad} \\ \lambda_3 * (V_{mad} - V_i) + \mu_3 * a_i, & V_i < V_{mad} \\ \end{cases} \tag{5}$$

$$V = \sum_{j=1}^{M} \varphi_j * d_j \tag{6}$$

$$V_{mad} = \alpha * V_{awc} + V_{pwp} \tag{7}$$

$$V_{fc} = V_{awc} + V_{pwp} \tag{8}$$

$$V_{pwp} = \varphi_{pwp} * D_{inch} \tag{9}$$

$$V_{awc} = \sigma_{awc} * D_{foot}, \tag{10}$$

where N represents the number of irrigation regions within a single orchard and a denotes the amount of water recommended by the RL agent. The parameters σ_{awc} and φ_{pwp} are determined by referencing the guidelines provided by the Almond Board of California [41] and tailored to our specific soil type in the testbed. Equations (6), (7), (8), (9), and (10) are previously introduced in Section 2 and outline the calculations for the reward components. These equations capture the relationship between plant health, water consumption, and the reward values utilized in our implementation.

In our implementation, the parameters of the reward function are specifically configured based on the specifications of our testbed, as indicated in Table 2. The values assigned to the parameters in Equation (5) (λ_1, μ_1, μ_2, λ_3, and μ_3) are determined through a grid search approach, aiming to identify the values that yield the most favorable rewards during the training process. A detailed explanation of the grid search methodology will be provided in Section 5. The parameter values listed in Table 2 align with the intended design objective of the reward function. Specifically, when the soil water content V_i exceeds the FC (V_{fc}) level, we impose penalties to account for the impact on plant health and excessive water consumption ($\lambda_1 = 3$, but $\mu_1 = 8$). If V_i falls below the V_{mad} threshold, we assign a higher penalty to reflect the adverse effect on plant health ($\lambda_3 = 10$, but $\mu_3 = 1$).

3.4 *DRLIC* Training

3.4.1 Policy Gradient Optimization. Within the DRL framework outlined above, there exists a range of policy gradient algorithms that can be employed to train the irrigation control agent. These algorithms are designed to maximize the objective function stated in Equation (3). To accomplish this, policy gradient algorithms estimate the policy gradient, which indicates the direction of policy improvement, and optimize the objective by employing stochastic gradient ascent, as described in Equation (11).

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \mathbb{E}_{\pi_{\theta}}[r] \tag{11}$$

In this study, we utilize the **Proximal Policy Optimization (PPO)** algorithm [48] for training the irrigation control agent. PPO has demonstrated successful applications in various domains, including navigation [38] and games [11]. One advantage of PPO is its stability and robustness to both hyperparameters and network architectures [48]. Moreover, PPO has shown superior performance compared to other policy gradient algorithms such as **Natural Policy Gradients (NPGs)** [32] and **Trust Region Policy Optimization (TRPO)** [47]. Additionally, PPO exhibits less bias when compared with Q-learning [54].

Although PPO has a relatively high sample complexity, requiring a large number of environment interactions, this concern is mitigated in our case due to the use of a cost-effective simulator, as explained in detail in Section 3.5. The efficiency of the simulator helps alleviate the potential challenges associated with the sample complexity of PPO, making it a suitable choice for our irrigation control problem.

The loss function in Equation (12) captures the objective of PPO by maximizing the expected cumulative rewards while considering a regularization term. This encourages exploration and prevents the policy from deviating too far from the previous policy during training, promoting stability and controlled policy updates.

The advantage function $\hat{A}_t$, computed using Equation (13), provides an estimate of the advantage or relative benefit of selecting a particular action at a given state. It represents the difference between the expected cumulative rewards obtained by following the current policy and the baseline value, which is typically the expected value of rewards under the current policy. By utilizing the advantage function, PPO can effectively estimate the advantages of different actions and update the policy accordingly to maximize the expected rewards.

$$L_{PPO}(\theta) = -\hat{\mathbb{E}}_t[min(w_t(\theta)\hat{A}_t, clip(w_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{12}$$

$$\hat{A}_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i} \tag{13}$$

$$w_t(\theta) = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{14}$$

In Equation (14), $\pi_{\theta}(a_t|s_t)$ represents the policy that is being updated through the loss function, whereas $\pi_{\theta_{old}}(a_t|s_t)$ refers to the policy that was used to collect data during interaction with the environment. Since the data collection policy is different from the policy being updated, it introduces a distribution shift, which can affect the stability of training.

To address this distribution shift, the importance sampling ratio $w_t(\theta)$ is introduced. This ratio corrects for the shift by measuring the relative likelihood of actions under the updated policy and the old policy. However, the ratio of probabilities can sometimes become extremely large, leading to unstable training. To mitigate this issue, the importance sampling ratio is clipped using a parameter ϵ. Clipping the ratio prevents it from exceeding a certain threshold and helps maintain stability during the policy update process. By controlling the magnitude of the importance sampling ratio, PPO ensures a balance between exploration and exploitation, leading to more reliable and effective training.

3.4.2 Data Collection and Preprocessing. On day t, the *DRLIC* agent observes the current state s, which represents a particular aspect of the environment, such as the moisture level in the soil. Based on this state, the agent selects an action a, which determines the amount of water to be applied. After the action is implemented, the soil-water environment transitions to a new state s_{t+1} the next day, reflecting the changes caused by the water application. Additionally, the agent receives a reward r that provides feedback on the quality of the action taken.

To generate training data for our *DRLIC* agent, we collect data pairs (s_t, a_t, r_t, s_{t+1}) by repeatedly observing the state, taking an action, observing the resulting next state, and receiving a reward. It is worth noting that to ensure stability during training, we normalize the collected data by subtracting the mean of the states and actions and dividing by their standard deviation. This normalization process helps to standardize the data and ensure that the agent can learn effectively across different scales of state and action values.

For our dataset, we utilize 10-year weather data spanning from 2010 to 2020. This extensive weather data provides a rich and diverse set of environmental conditions that the *DRLIC* agent can learn from. By leveraging this dataset during the training process, the agent can acquire a comprehensive understanding of how different states and actions relate to rewards under various weather patterns and environmental dynamics.

3.4.3 Training Process. Ideally, the control agent of *DRLIC* should be trained directly in an almond orchard. However, due to the long control interval of irrigation systems, it would take an impractical amount of time, approximately 384 years, for a DRL agent to converge through direct training in a real orchard. To overcome this challenge, we adopt a feasible solution by utilizing a high-fidelity simulator as a surrogate for the real-world environment.

In the domain of soil-water management, there are currently no readily available high-fidelity simulators. Hence, we employ a data-driven simulator to accelerate the training process of *DRLIC*. As described in Section 3.5, we utilize the soil water content predictor as our surrogate simulator. This predictor leverages historical weather data spanning 10 years to approximate the soil-water dynamics. By using this data-driven simulator, the *DRLIC* agent can "experience" the weather patterns and corresponding soil-water conditions of 10 years within a matter of minutes, significantly speeding up the training process.

The training procedure of *DRLIC* is outlined in Algorithm 1. We train the *DRLIC* agent over 1,000 episodes, with each episode representing a span of 30 days. During each episode, we collect 30 data pairs (s_t, a_t, r_t, s_{t+1}) under different weather conditions. The objective defined in Equation (12) is optimized through stochastic gradient ascent using Equation (3). The training process continues until convergence, which is determined by comparing the total reward obtained at the end of each episode with the previous total. If the current episode's reward does not change by more than ±3%, we consider the policy to have converged. If the policy does not converge within 100 training iterations (100 episodes), the training is halted.

After the training phase, the well-trained *DRLIC* agent is ready to be deployed in a real almond orchard. When faced with a new environment, such as a different orchard, we first collect real-world irrigation data using existing controllers, such as ET-based control. This data is used to build a soil water content predictor specific to the new environment, which characterizes the water balance in the root zone soil. Subsequently, we leverage this soil water content predictor to expedite the training process for the new orchard. Once trained, the *DRLIC* agent can be deployed and employed to optimize irrigation control in the new almond orchard.

ALGORITHM 1: *DRLIC* Training Algorithm.

```
Input: State s, Action a, Reward r,an initialized policy, $\pi_\theta$;
Output: A trained irrigation control agent;
for $i = 0, \ldots, \# Episodes$ do
    State ← Soil-water environment;
    $\theta_{old} \leftarrow \theta$ ;
    for $t = 0, \ldots, Steps$ do
        $\hat{a}_t = \pi_\theta(s_t)$;
        $s_{t+1}, r_{t+1} = env.step(\hat{a}_t)$;
    Compute $\hat{A}_t$ ;
    With minibatch of size M;
    $\theta \leftarrow \theta - \alpha \nabla_\theta L_{PPO}(\theta)$;
```

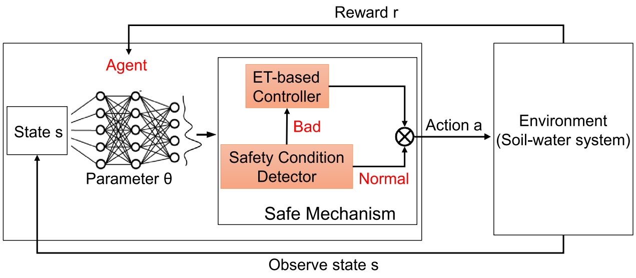


Fig. 6. Reinforcement learning with safe mechanism.

3.5 Safe Mechanism for Irrigation

We propose a safe mechanism that integrates the DRL agent and the ET-based controller into a tightly coupled closed-loop system. Figure 6 provides a visual representation of the workflow of this safe mechanism, which comprises several key elements designed to ensure safe irrigation control. (i) In contrast to a pure DRL framework, we introduce a safety moisture condition detector. This detector evaluates whether the action suggested by the DRL algorithm is considered safe in terms of maintaining appropriate soil moisture levels. (ii) If the safety moisture condition detector deems the DRL algorithm's action to be safe, the action is forwarded to the DRL agent, which takes charge of the irrigation control for that particular cycle. (iii) If the action is deemed unsafe by the detector, an alternative course of action is taken. In such cases, we rely on an ET-based controller to generate an action for the irrigation control cycle. (iv) The *DRLIC* agent acts as the DRL agent for future control cycles, continually learning and refining its decision-making process based on feedback and environmental conditions. To facilitate the safe mechanism, we introduce two important components: the soil water content predictor and the safety condition detector.

Soil Water Content Predictor. To enhance the safety of our system and enable early detection of potentially unsafe actions, we develop a soil water content predictor. This predictor is designed to forecast the moisture trend following the implementation of an action, enabling us to estimate the potential impact on soil moisture levels.

Furthermore, we design a safe condition detector that detects the almond health penalty. The objective is to determine whether the damage metric for an almond tree exceeds a predefined threshold. If the detector detects that the health penalty is higher than the threshold, it commands *DRLIC* to switch from the DRL-based controller to the ET-based controller.

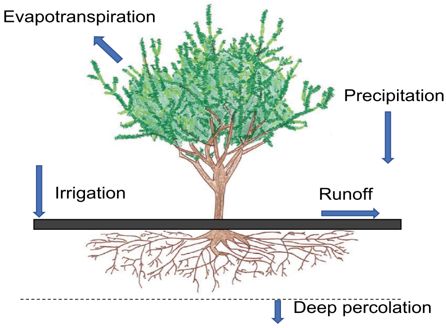


Fig. 7. Water balance in the root zone soil.

We develop a soil water content predictor that accurately describes the water balance within the root zone soil. As illustrated in Figure 7, the variations in water storage in the soil are attributed to both inflows, such as irrigation and precipitation, and outflows, primarily through evapotranspiration. This leads us to the following mathematical expression:

$$V_{i,t+1} = c_1 * V_{i,t} + c_2 * (A_{i,t} + P_t) + c_3 * E_t + b. \tag{15}$$

Here, $V_{i,t+1}$ represents the predicted moisture level in the root zone for the ith irrigation region after applying the action determined by the DRL algorithm. E_t and P_t denote the plants' evapotranspiration and the measured rainfall, respectively, during the time period t. The irrigation amount for the ith irrigation region is denoted as $A_{i,t}$. The coefficients c_1, c_2, and c_3 are included in Equation (15) to account for the proportional relationship assumed between soil moisture level and runoff as well as water percolation [12, 13, 43]. These coefficients can be accurately determined using system identification techniques [16]. It is worth noting that all variables in the equation are typically expressed in inches.

To calculate the ET, we adopt a simplified calculation model established in [26], represented by Equation (16):

$$E_t = \Gamma_c * RA * TD^{(1/2)} * (T_t + 17.8°C) \tag{16}$$

In Equation (16), Γ_c is a crop-specific parameter [44] and RA stands for extraterrestrial radiation, which is expressed in the same unit as E_t. TD represents the annual average daily temperature difference, which can be derived from local meteorological data, whereas T_t refers to the average outdoor temperature during the tth time period.

To acquire the weather data, we obtain it from a local weather station. The calculation of ET is based on the simple model established in [26], which incorporates the parameters mentioned above to estimate evapotranspiration accurately.

Safety Condition Detector. We utilize the discrepancy between the predicted moisture level and the lower bound as an indicator to estimate the potential damage to almond trees. As outlined in Section 2, the lower bound is represented by MAD. To serve as a safety condition detector, we calculate the sum of differences, $\sum_{i=1}^{N}(V_{mad} - V_{i,t+1})$, where $V_{i,t+1}$ represents the predicted moisture level for the t timestep from the ith irrigation region, and V_{mad} denotes the lower bound for water content. Whenever the safety condition detector detects a hazardous irrigation action, *DRLIC* triggers the use of the ET-based controller as a precautionary measure.

Parameter Learning of Our Soil Water Content Predictor. We leverage the designed testbed to collect the irrigation amount of almond trees over a period of 2 months. Each day, we

Table 3. Coefficients of Predictor for Each Tree

	c1	c2	c3	b	R^2	NRMSE
Tree1	0.973	0.288	−0.103	0.003	0.982	0.062
Tree2	0.937	0.325	−0.121	0.013	0.985	0.071

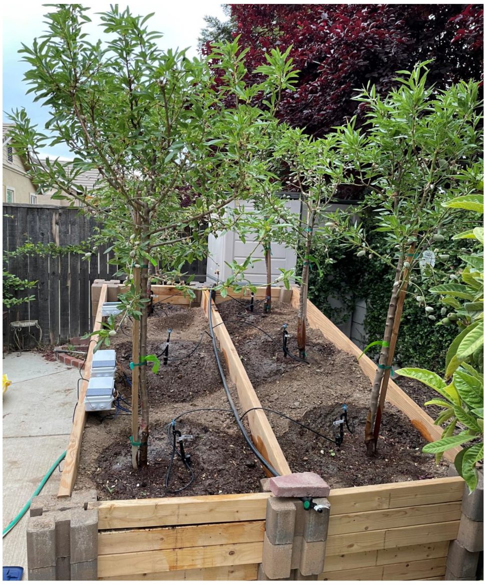


Fig. 8. Testbed and micro-sprinkler irrigation system.

collect the ET value from a local weather station [4], and the moisture level for each tree is being collected by the designed *DRLIC* node. Subsequently, we apply the linear least square method to estimate the coefficients, which helps us understand the relationship between the moisture level and the related factors.

To assess the strength of this relationship, we utilize R^2 as a measure. By examining the results in Table 3, we can observe that R^2 is close to 1, indicating a strong correlation between the irrigation practices, ET, precipitation, and soil water content for the trees. This suggests that irrigation, together with ET and precipitation, plays a key role in determining the moisture level in the soil.

Furthermore, we employ the **normalized root-mean-square error (NRMSE)** as a measure of goodness-of-fit for our predictors. Notably, the NRMSE value is less than 0.1, demonstrating that our predictor achieves a high level of accuracy in predicting soil water content.

4 TESTBED AND HARDWARE

4.1 Testbed and Microsprinkler Description

In Figure 8, we present our micro-sprinkler irrigation testbed, which is specifically designed for this study. The micro-sprinkler irrigation system installed in the testbed is identical in terms of hardware and micro-sprinkler coverage. The dimensions of the testbed measure 290 cm × 160 cm, and it features a grid arrangement of micro-sprinklers in a 3 × 2 layout, with each micro-sprinkler positioned 97 cm apart from the next.

For this testbed, we select state-of-the-art micro-sprinklers by Rainbird, specifically the 1/4′ with a 360° pattern. These micro-sprinklers are recognized for their advanced technology in

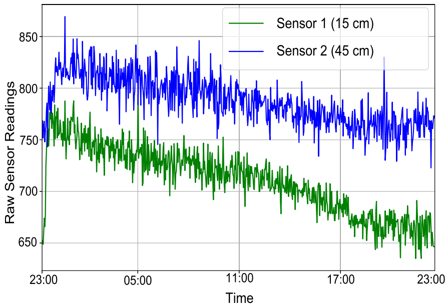


Fig. 9. Daily soil moisture readings.

the field of micro-sprinkler irrigation. Within the testbed, we plant a total of six young almond trees, with three trees allocated to each section. These almond trees have an average height of 2 meters, and their growth and development are closely monitored throughout the study.

To simulate real-world conditions, we have collected soil from a local orchard, representative of typical loam soil. The soil used in the testbed has a volume of 2.7 m^3, and its plant-available water-holding capacity is measured at 2.4 inches of water per foot.

This meticulously designed micro-sprinkler irrigation testbed provides us with a controlled environment to conduct accurate and reliable experiments, enabling us to investigate the effectiveness of different irrigation strategies and their impact on almond tree growth and water management.

4.2 *DRLIC* Node Development.

The *DRLIC* node, as depicted in Figure 4, is composed of four primary components: sensors, actuator, power supply, and transmission module.

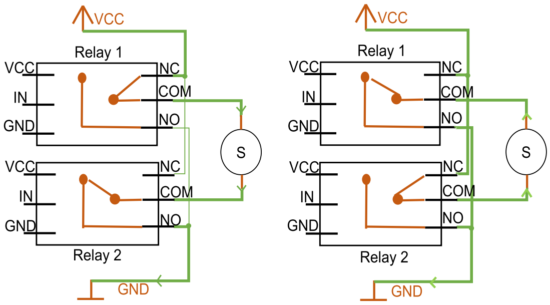
Sensors: The *DRLIC* node incorporates multiple moisture sensors that are strategically placed at different depths to accurately monitor soil moisture. These sensors exhibit varying levels of sensitivity and have different soil volume coverage capacities. We install moisture sensors designed for the 12-inch depth, following the guidelines of the Almond Board Irrigation Improvement Continuum [41]. Since the root zone depth of the almond trees in our testbed is 24 inches, we assign two moisture sensors to each *DRLIC* node to ensure comprehensive coverage and precise soil moisture assessment.

A crucial capability of the *DRLIC* node is its ability to measure the **volumetric water content (VWC)** in the soil surrounding the almond trees. To achieve this, we have procured high-quality Decagon EC-5 sensors[3] known for their research-grade performance and reported accuracy of ±3%. Figure 9 displays the raw sensor readings collected over a day, capturing the dynamic changes in soil moisture with a high sampling frequency. These sensors report the soil's dielectric constant, an electrical property that strongly correlates with the VWC.

To convert the raw sensor readings into meaningful VWC values, we utilize a linear calibration function provided by the sensor manufacturer shown in Equation (17) for this purpose. The resulting φ values range between 0% and 100%, providing a comprehensive measure of the soil's moisture content. It is important to note that the φ values of saturated soils typically fall within the range of 40% to 60%, which may vary depending on the soil type under consideration.

$$\varphi(m^3/m^3) = 9.92 * 10^{-4} * raw\_reading - 0.45 \tag{17}$$

[3] Decagon devices. http://www.decagon.com/products/soils/



(a) Positive Current Pulse.

(b) Negative Current Pulse.

Fig. 10. On and off circuit diagram for latching solenoid.

Actuator: The actuator component of the *DRLIC* node utilizes a latching solenoid equipped with two relays. Unlike a standard solenoid that requires a constant power supply to enable water flow, a latching solenoid offers a more suitable solution for a battery-powered system. This choice was made to extend the operational lifetime of the *DRLIC* node. In contrast to the standard solenoid, which would deplete the power of the 9-volt performance all-purpose alkaline batteries from Amazon within 8 hours when supplying power to a 12V DC solenoid continuously, the latching solenoid requires only a brief pulse of positive or negative voltage (25 ms) to open or close, respectively.

To control the latching solenoid, an H-bridge configuration is commonly employed to generate a bi-directional current [58]. However, since the ESP32 and latching solenoid have different voltage requirements, a special design is necessary to accommodate these distinct voltage specifications. By addressing the voltage requirements of both the ESP32 and the latching solenoid, we ensure compatibility and efficient operation of the actuator within the *DRLIC* node.

To effectively control the latching solenoid, we design a circuit diagram that utilizes two relays, minimizing the overall connection complexity. A relay functions as an electrically operated switch, facilitating the control of current flow. Figure 10 illustrates the circuit diagram for turning the latching solenoid on and off. Initially, both relays are in a **normally closed (NC)** position, ensuring no current passes through the solenoid (S).

To activate the solenoid, Relay 1 is switched from NC to **normally open (NO)** for a duration of 25 ms, generating a positive current pulse that energizes the solenoid. The current path in Figure 10(a) is as follows: VCC (power source) -> NC_1 (normally closed contact of Relay 1) -> COM_1 (common contact of Relay 1) -> S (solenoid) -> COM_2 (common contact of Relay 2) -> NO_2 (normally open contact of Relay 2) -> GND (ground). This configuration allows the current to flow through the solenoid, activating it.

Conversely, to turn the solenoid off, Relay 2 is switched from NC to NO for 25 ms, causing the solenoid to return to the closed position. The current path depicted in Figure 10(b) is as follows: VCC -> NC_2 (normally closed contact of Relay 2) -> COM_2 -> S -> COM_1 -> NO_1 (normally open contact of Relay 1) -> GND. This path interrupts the current flow through the solenoid, de-energizing it and returning it to the closed state.

To prevent over-irrigation in the event of a power failure, a power supply module is incorporated into the system, ensuring a continuous and uninterrupted power source for the *DRLIC* node.

Power Supply: The power supply for the system comprises a 5-V, 1.2-W solar panel for energy harvesting and a 18650 lithium-ion battery with a capacity of 3.7 V 3000 mAh for energy storage. To facilitate the charging of the lithium battery, we utilize the TP4056 lithium battery charger module, which includes built-in circuit protection to prevent battery over-voltage and reverse

Table 4. Hyperparameters

RL Parameters	Value	General Parameters	Value
Learning_rate	0.01	Iteration	1000
Discount_factor	0.99	Number_neurons	256
Number_layers	2	Clip_parameter	0.3
Minibatches	128	Number_workers	2

polarity connection. This power supply setup effectively powers all the sensors, including 1 ESP32, 2 moisture sensors, 2 relays, and 1 latching solenoid.

By harnessing energy from the solar panel and storing it in the lithium battery, the power supply module ensures a continuous and reliable power source for the entire system. This capability is particularly crucial for the actuator module, as it prevents over-irrigation in the event of a power failure.

Transmission Module: The transmission process in the system consists of two components: the uplink and the downlink [60, 61]. In the uplink path, the ESP32, a low-cost and low-power **system on a chip (SoC)** with built-in Wi-Fi capability, samples the moisture sensor readings from the field. These readings are then transmitted from the ESP32 to the base station, where they serve as input for the optimal control algorithm.

In the downlink path, the control command calculated by the DRL agent is sent from the base station to all ESP32 modules. This command is responsible for activating or deactivating the solenoids connected to each ESP32 node. By coordinating the control commands, the downlink communication enables the precise control of the irrigation system, ensuring efficient water distribution to the almond trees.

5 IMPLEMENTATION

In this section, we provide a detailed explanation of the implementation of the *DRLIC*, along with the process of tuning its hyperparameters.

DRLIC **Implementation Details.** Our *DRLIC* system is implemented in Python, utilizing various open-source frameworks such as Pandas, Scikit-learn, and Numpy. To facilitate the control scheme for *DRLIC*, we leverage RLlib [35], a scalable reinforcement learning framework that supports TensorFlow, TensorFlow Eager, and PyTorch. RLlib offers extensive customization options for training the *DRLIC* system, including target environment modeling, neural network modeling, action set building and distribution, and optimal policy learning.

For our specific implementation of *DRLIC*, we gather 10 years of weather data spanning from 2010 to 2020. We divide this dataset into 9 years for training and 1 year for testing purposes. To optimize the training process, we employ the Adam optimizer with a learning rate set to 0.01 for gradient-based optimization. Additionally, the discount factor is set to 0.99 to account for future rewards. The neural network model utilized in *DRLIC* consists of two hidden layers, each comprising 256 neurons. The detailed hyperparameters are shown in Table 4.

The *DRLIC* agent is trained and executed on a local server with a 64-bit quad-core Intel Core i5-7400 CPU operating at 3.00 GHz. The server is running Ubuntu 18.04 as the operating system. These specifications ensure the computational resources necessary for efficient training and deployment of *DRLIC*.

Training Details and Tuning Hyperparameters. The performance of the *DRLIC* agent relies heavily on the values chosen for its hyperparameters. However, finding the ideal values that guarantee improved total reward for the system is not a straightforward task. To enhance the performance of the *DRLIC* agent and optimize its hyperparameters, we utilize a tuning approach. This approach involves optimizing parameters such as λ, μ (associated with rewards and penalties) and learning rates.

Specifically, we employ a grid search approach, which enables us to define a range of values for each hyperparameter to be considered. The grid search process systematically constructs and evaluates the model using every possible combination of hyperparameters. To assess each learned model, we utilize cross-validation. This tuning approach enables us to identify the best hyperparameter configuration for the *DRLIC* agent, ensuring optimal performance of the *DRLIC* system.

6 EVALUATION

In this section, we assess the performance of the *DRLIC* system in real-world conditions. The evaluation involves a 15-day field trial of the *DRLIC* system, in which we closely observe its functionality and effectiveness. This real-world evaluation provides valuable insights into how well the *DRLIC* system performs in practical scenarios.

6.1 Experiment Setting

6.1.1 Baseline Strategy:. We evaluate the performance of *DRLIC* by comparing it to two state-of-the-art irrigation control schemes that will be introduced in Section 7. These baseline strategies serve as benchmarks to assess the effectiveness and superiority of the *DRLIC* system. The baseline irrigation systems and *DRLIC* system operate on a daily schedule.

ET-Based Irrigation Control [41]. To implement an ET-based controller, we rely on data from a local weather station to obtain the ET loss for the previous day. This information allows us to estimate the amount of water that needs to be replenished in the soil. To compensate for this loss, we utilize the sprinkler's irrigation rate, which is provided by its dataset. By combining the ET loss and the sprinkler's irrigation rate, we calculate the duration for which the irrigation system should be activated to adequately replenish the water in the soil. This approach ensures that the irrigation is aligned with the water needs of the plants based on the local weather conditions.

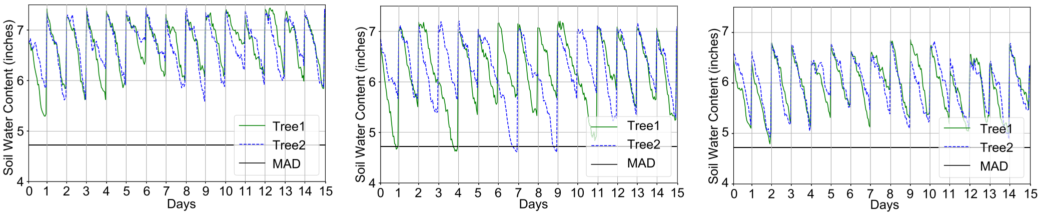
Sensor-Based Irrigation Control [25]. The sensor-based controller employs two specific thresholds to determine the irrigation needs based on the soil water content. The first threshold is set at 4.96 inches, which is 10% higher than the MAD level. This threshold ensures that irrigation is initiated before the soil experiences a significant water deficit, preventing under-irrigation prior to the wetting front reaching the depth of the sensor.

The second threshold is set at 6.97 inches, which is 5% below the FC level. This threshold allows for some storage capacity for rainfall and prevents over-irrigation. The careful selection of these thresholds is based on a thorough understanding of the soil environment in our specific testbed. By setting these thresholds appropriately, we can effectively manage irrigation to maintain optimal soil moisture levels for the health and growth of the plants.

6.1.2 Performance Metrics. In our evaluation, we assess the performance of the *DRLIC* system and compare it to two baseline systems using two key performance metrics.

Quality of Service. The first metric we consider is the quality of service, which focuses on the irrigation system's ability to maintain soil moisture above the MAD threshold consistently across all measured locations. While the irrigation system may not have control over factors such as solar exposure and soil nutrients, it directly influences soil moisture levels. Ensuring that the system maintains moisture above the MAD threshold guarantees healthy plant growth and prevents any production loss. In this article, we refer to this as the quality of service metric for the irrigation system.

Water Consumption. The second metric we evaluate is water consumption. Since we have direct control over the activation times of each micro-sprinkler and each utilizes a water supply, we



(a) ET-based Method (b) Sensor-based Method (c) *DRLIC*

Fig. 11. Daily soil water content of different irrigation methods (15 days).

can monitor the amount of water consumed by all three systems at any given time. Minimizing water consumption while satisfying the quality of service constraints is crucial for achieving efficient irrigation practices. Therefore, water consumption is another important metric we consider.

6.1.3 Experiments in our Testbed. In order to assess the performance of the *DRLIC* system, we conduct a real-world deployment validation study focusing on plant health and water consumption over a 15-day period. The study is conducted using a testbed consisting of six almond trees, as depicted in Figure 9.

To evaluate the *DRLIC* system alongside the baseline control schemes, we allocate the upper, middle, and lower two trees separately for irrigation by *DRLIC*, sensor-based control, and ET-based control. As there is no runoff between the trees in our testbed, we ensure the independent operation of the three irrigation systems. Each micro-sprinkler is controlled by a dedicated *DRLIC* node, allowing us to vary only the schedules sent to the nodes while keeping the other aspects consistent among the three systems.

6.2 Experiment Results

6.2.1 Quality of Service. The primary objective of installing irrigation systems is to ensure the health of almond trees without incurring any production loss. Figures 11(a), 11(b), and 11(c) illustrate the daily soil water content in the field for the ET-based control, sensor-based control, and *DRLIC* systems, respectively. The black horizontal line represents the MAD level below which the tree health is impacted.

Upon examining the figures, we observe that both *DRLIC* and the ET-based system effectively maintain the soil water content above the MAD threshold throughout the 15-day deployment, meeting the requirements for almond tree health. However, the trees irrigated by the sensor-based method experience periods of under-irrigation lasting 18 hours for four days (days 1, 4, 7, and 9) as the soil water content falls below the MAD level. This occurs because the moisture level of the previous day is close to but not reaching the MAD, causing the sensor-based method to refrain from irrigation despite the soil moisture trending towards under-irrigation. In contrast, the *DRLIC* system dynamically adjusts irrigation based on learned models of soil water dynamics, effectively maintaining the soil water content close to the MAD level.

At the start of the experiment, all three irrigation systems had sufficient water content. Notably, in the ET-based control system (Figure 11(a)), the soil water contents of the two trees were significantly above the FC threshold. This highlights the limitations of ET-based control and underscores the core contribution of our work. The irrigated regions do not receive moisture uniformly, and in most cases, the ET-based controller applies more water than necessary for optimal plant growth.

Overall, the evaluation demonstrates that the *DRLIC* system effectively manages irrigation by adapting to the specific water requirements of almond trees, maintaining soil moisture levels near the MAD threshold and outperforming both the sensor-based and ET-based control strategies.

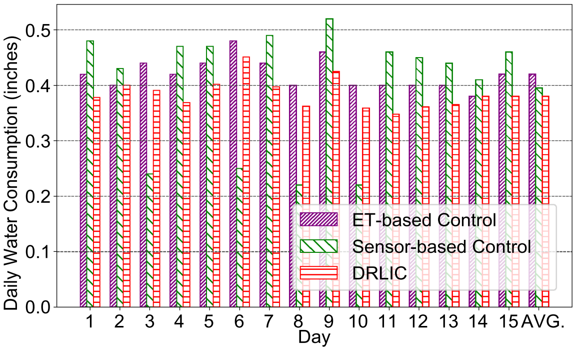


Fig. 12. Daily water consumption.

6.2.2 Water Consumption. When considering the adoption of a new almond irrigation control system, one of the primary concerns is its efficiency. The system's ability to provide a return on investment based on increased efficiency often plays a significant role in its acceptance. Furthermore, the environmental benefits associated with reduced freshwater consumption are evident and contribute to the promotion and adoption of such systems.

In our experimental setup, each micro-sprinkler's water source is regulated to the industry standard pressure of 30 psi. Each micro-sprinkler has a predefined water distribution rate per unit time, as outlined in the almond irrigation manual [41]. By precisely tracking the activation of each micro-sprinkler by the system, we can accurately determine the amount of water consumed.

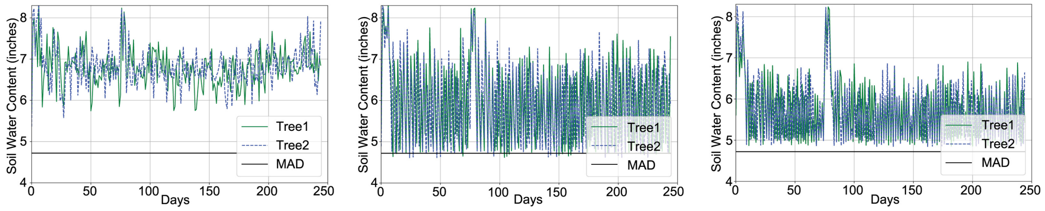
Figure 12 illustrates the daily irrigation amounts for two trees controlled by the ET-based, sensor-based, and *DRLIC* systems during a 15-day deployment experiment. From the figure, we observe that *DRLIC* achieves water savings of approximately 9.52% and 3.79% on average compared with the ET-based and sensor-based control, respectively, over the 15-day period. The ET-based control method employs centralized control to irrigate all almond trees without considering their specific needs. In contrast, the sensor-based control method conserves water by monitoring moisture levels and irrigating when the moisture falls below the MAD threshold. However, the specific thresholds used in the sensor-based control are site specific and may not be optimal. In contrast, the *DRLIC* system can learn and adapt to the optimal irrigation control by interacting with the local weather and soil water dynamics.

In summary, the evaluation demonstrates that *DRLIC* offers improved water efficiency compared with both the ET-based and sensor-based control methods. By optimizing irrigation based on learned models, *DRLIC* achieves significant water savings while meeting the specific water requirements of almond trees. These findings underscore the potential economic benefits and environmental advantages associated with adopting the *DRLIC* system for almond irrigation.

6.3 Simulation Results

In this section, we present the simulation results of *DRLIC* and two baseline systems for an entire growing season. By evaluating their performance over a more extended period, we gain a comprehensive understanding of their effectiveness and suitability for long-term irrigation control.

6.3.1 Quality of Service. Figure 13 depicts the daily soil water content for the ET-based control, sensor-based control, and *DRLIC* during the simulation. The MAD level is indicated by the black horizontal line, representing the threshold for maintaining almond health. Remarkably, all four control methods effectively sustain the soil water content above the MAD level, ensuring the well-being of the almond trees.



(a) ET-based Method (b) Sensor-based Method (c) *DRLIC*

Fig. 13. Daily soil water content of different irrigation methods.

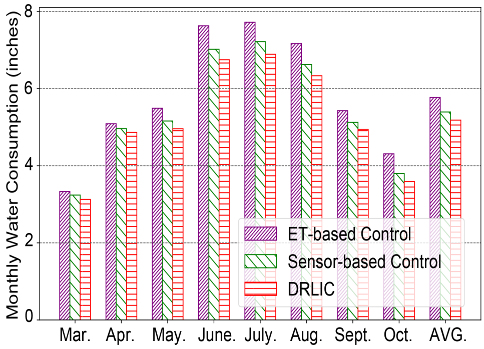


Fig. 14. Monthly water consumption.

Notably, on specific dates such as March 2, March 5, and March 24, as well as May 16 and May 18, all control systems demonstrate identical soil water content across all trees. This alignment can be attributed to substantial rainfall events during those days, which saturated the soil and reached FC.

When comparing *DRLIC* against the ET-based control strategy, as shown in Figure 13(a), we observe that the soil water content for the two trees under the *DRLIC* system differs significantly and consistently exceeds the MAD level. In contrast, the sensor-based control maintains two distinct moisture levels, initiating irrigation when reaching the lower level and stopping irrigation at the higher level. However, even with careful calibration of these thresholds for each tree, Figure 13(b) reveals 43 instances in which the soil water content slightly falls below the MAD level, potentially affecting almond production.

Overall, both $DRLIC_{MAD}$ and the *DRLIC* system (Figure 13(c)) successfully ensure uniform irrigation for both trees throughout the growing season, effectively maintaining their health and minimizing the risk of under-irrigation.

6.3.2 *Water Consumption.*

Figure 14 illustrates the monthly water consumption of the ET-based system, sensor-based system, and *DRLIC* system from March to October. It is evident that the *DRLIC* system consistently consumes less water than the ET-based and sensor-based systems throughout each month. Furthermore, the irrigation amounts for all systems exhibit an initial increase followed by a decrease. This pattern can be attributed to two factors.

Firstly, as the spring rains cease and the weather becomes hotter, the demand for irrigation increases. Secondly, almond trees undergo distinct stages in their annual lifecycle, including winter dormancy, the vibrant bloom in March, growth throughout spring, and nut development during summer. The year concludes with the harvest period spanning from mid-August to October.

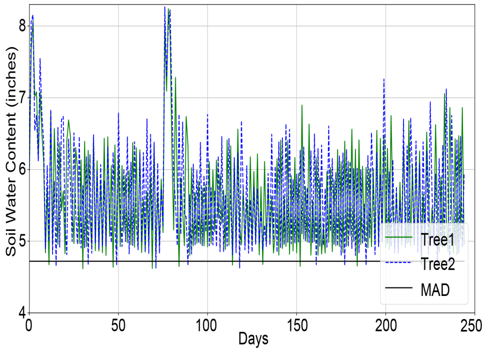


Fig. 15. Daily soil water content (w/o safe mechanism).

Consequently, water consumption tends to be higher in the summer months compared with other seasons.

Overall, the *DRLIC* system demonstrates water-saving benefits, achieving a 10.21% and 3.93% reduction in water usage compared with the ET-based system and sensor-based system, respectively, over the entire growing season. Considering that California's 2019 almond acreage was estimated at 1,530,000 acres, with almond irrigation consuming approximately 195.26 billion gallons per year, the implementation of the *DRLIC* system could potentially save 19.94 and 7.67 billion gallons of water annually compared with the ET-based system and sensor-based system, respectively[24]. This significant water conservation potential further highlights the environmental advantages of adopting the *DRLIC* system.

6.4 Effect of Our Safe Irrigation Mechanism

During the 15-day deployment, we observe that on two occasions (Days 2 and 14 in Figure 11(c)), the *DRLIC* system triggered the ET-control method. This observation is further supported by Figure 12, where the water consumption of the ET method and *DRLIC* on those two days are identical. To understand the reason behind this behavior, we examine the weather data and discover that the wind speeds on Days 2 and 14 were 7.2 and 11.9 mph, respectively, significantly higher than the average wind speed of 2.8 mph during the other 13 days.

To gain further insights, we conduct simulations of the *DRLIC* system with and without a safety mechanism for the entire growing season. These simulations, labeled as Robust-RL and RL-only, respectively, are depicted in Figures 13(c) and 15, illustrating the daily soil water content. From the perspective of almond health, the Robust-RL method effectively maintains the health of the trees, with 0 days below the MAD level. In contrast, the RL-only irrigation method exhibits 21 days below the MAD level. This disparity arises due to the fact that the RL models trained on past weather data may not perform optimally when faced with test weather data that deviates significantly. While it might be feasible to train a model on changing weather conditions to obtain a more robust policy, offline training alone cannot account for all potential weather changes.

However, the RL agent equipped with a safety mechanism in the *DRLIC* system proves to be robust against weather variations. The safety condition detector within the system identifies hazardous actions from the RL agent, enabling the ET system to assume control and mitigate any potential risks. This integration of safety mechanisms ensures the *DRLIC* system's ability to handle weather changes effectively and maintain the desired irrigation levels for optimal almond health.

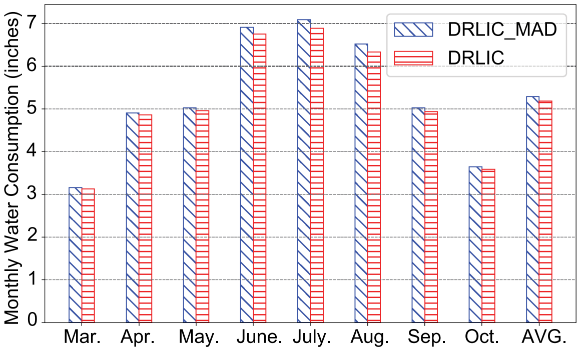


Fig. 16. Water consumption for *DRLIC* with different rewards.

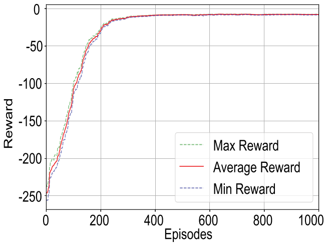


Fig. 17. Reinforcement learning policy convergence.

6.5 Effect of Proposed Reward

In this section, we present the simulation results of *DRLIC* with different reward functions for the entire growing season (March 1 to October 31, 246 days).

To balance the objectives of minimizing water consumption while ensuring plant health, we consider three situations in the reward formulation: (1) when the soil water content (V_i) is higher than the field capacity (V_{fc}) level, (2) when V_i is between V_{fc} and the maximum allowable depletion (V_{mad}), and (3) when V_i is lower than V_{mad}. Only in the second situation are the plants in good health. To evaluate the effectiveness of our reward function, we compare it with a simple reward (*DRLIC*$_{MAD}$) commonly used in sensor-based methods [25], which only focuses on maintaining V_i above V_{mad}. The reward is defined as $R = -\sum_{i=1}^{N} \lambda_3 * (V_{mad} - V_i) + \mu_3 * a_i, V_i < V_{mad}$. This reward function assigns higher penalties to plant health when V_i is lower than V_{mad}, as lower V_i significantly impacts plant health. The parameters used are the same as in Section 3.3.3.

Figure 16 illustrates the water consumption of *DRLIC* with our proposed reward (*DRLIC*) and the simple reward (*DRLIC* _MAD). *DRLIC* saves an additional 2.04% of water compared with *DRLIC* _MAD since the latter does not account for the case in which V_i is higher than V_{mad}. *DRLIC* considers two additional scenarios by applying different penalties for plant health and water consumption. In the case of over-irrigation, in which water consumption is excessively high, the penalty for water consumption is increased relative to plant health. Conversely, in situations in which the plants are in good health, *DRLIC* aims to maintain V_i close to V_{mad} to achieve water savings while ensuring plant health.

6.6 *DRLIC* Policy Convergence

Figure 17 depicts the DRL training process, demonstrating that the policy converges around the 500th training iteration. In our training setup, we define the length of an episode as 30 days. To

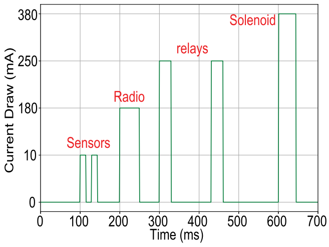


Fig. 18. Energy profile for different kinds of sensors.

expose the policy to different soil water content conditions and encourage it to avoid depleting water below the MAD level, we randomly vary the initial soil water content for each tree between the FC of 7.08 inches and the MAD of 4.72 inches at the start of each episode. Initially, the RL policy receives a substantial negative reward as it lacks knowledge of a valid sequence of actions that maximizes the reward. However, as training progresses, the policy gradually learns optimal actions, leading to convergence at the 500th training iteration. The entire training process, comprising 1,000 training iterations, takes approximately 4 hours using a 64-bit quad-core Intel Core i5-7400 CPU operating at 3.00 GHz.

6.7 Energy Consumption of Sensor Nodes

From a wireless sensor network perspective, it is crucial for a system to operate autonomously for extended periods without requiring user intervention. The same applies to *DRLIC* nodes, especially when they are deployed on the ground. To address this, our hardware and software are meticulously designed to minimize energy consumption.

DRLIC nodes are equipped with a latching solenoid that enables water flow to be controlled through short power pulses rather than a continuous supply. Additionally, to further conserve energy, the radio in each node adopts duty cycling, activating for only a 10-second period every minute. This higher data frequency is necessary because the base station can send an “off” command to *DRLIC* with minute-level granularity. Among the various peripherals in our devices, including the two moisture sensors, solenoid, two relays, and radio, significant energy consumption is observed.

To meet these energy demands, we implement an energy-harvesting mechanism that leverages a 5/6 V 1.2 W solar panel. Figure 18 illustrates the energy consumption of different sensors. Each moisture sensor sample requires 10 mA of power for 10 ms, while each activation of the latching solenoid necessitates 380 mA of power for 30 ms. The ESP32 radio consumes 180 mA of power for 50 ms in transmitting mode, and the relay requires 250 mA for 20 ms when switching on or off. To avoid premature power cutoff, we introduced a safety band of 50% in the timing for both devices, resulting in 15-ms and 45-ms trigger durations for the sensor and solenoid, respectively. Overall, the solar-harvesting mechanism adequately meets the daily energy requirements of all the sensors in the *DRLIC* nodes.

Figure 19 showcases the energy-charging and energy-discharging process over 2 days. After discharging during the night, the battery level of the 18650 battery begins to increase at 9:15 am on May 3. It typically takes 2 hours to fully charge the battery (from 9:15 am to 11:35 am). The battery remains at 100% charge from 11:35 am to 6:45 pm, during which time the harvested solar energy fulfills the energy demands of all the sensors in the *DRLIC* nodes. The battery gradually discharges from 100% at 6:45 pm on May 3 to 90.7% at 8:45 am on May 4th. This energy-charging and energy-discharging cycle repeats. On average, the lowest battery level observed is 90%. Throughout the

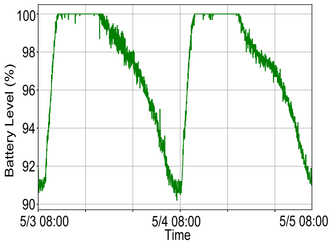


Fig. 19. Battery charging and discharging cycle.

Table 5. Micro-sprinkler Node Manufacture Cost

Component	Price	Component	Price
Moisture Sensor x 2	$250.00	ESP32	$6.50
18650 Li-Ion Battery	$3.00	Solar Panel	$4.30
Latching Solenoid	$4.00	Switch Relay x 2	$5.00
Waterproof Enclosure	$12.00	Maintenance Fee	$10.00
		Total	$294.80

2-week deployment, we discover that even on cloudy days, the battery can still be charged, albeit requiring an additional hour to reach full capacity.

6.8 Return on Investment

When considering the purchase or upgrade of an irrigation control system, a primary concern is the return on **investment (ROI)**, which refers to the duration required to save enough money from water consumption to cover the cost of the new system. To calculate the ROI of the *DRLIC* system, we consider the initial investment cost of the *DRLIC* system and the savings achieved through reduced water consumption resulting from increased irrigation efficiency.

First, we calculate the cost of developing a single *DRLIC* node. All the components required for a *DRLIC* node can be found in consumer electronics and home improvement stores. The cost of each component is listed in Table 5. The total cost for a *DRLIC* sensing and actuation node amounts to $294.80, with a significant portion of the budget allocated to two high-quality soil moisture sensors known for their accuracy and long lifespan.

The factors that primarily influence the payback period of our system are the water price and the volume of water saved by *DRLIC*. Water prices can vary significantly across different irrigation districts and over time. For this study, we assume 100% groundwater usage and availability. The monthly cost of irrigation water for each tree is estimated to be $11.30. Based on our experimental results, *DRLIC* can achieve a 9.52% reduction in water expenses per month, equivalent to $1.08. Considering that almond orchards typically have 100 trees per acre, *DRLIC* can save $108.00 per month. Let's take the example of a 60-acre almond orchard with 10 irrigation regions, where each region covers six acres. In each irrigation region, *DRLIC* can save $648.00 per month.

To deploy the *DRLIC* system, one *DRLIC* node is needed per irrigation region, with a cost of $294.80 per node. The existing infrastructure, such as pipelines and micro-sprinklers under each tree, can be used for the other irrigation components. Hence, the cost of upgrading the existing irrigation system with our control system amounts to $294.80 for each irrigation region in an orchard. With our system's monthly savings of $648.00 per irrigation region, the investment can be recouped in less than half a month. The *DRLIC* system can be easily upgraded to control micro-sprinklers of any type, delivering site-specific actuation for systems at any scale.

As shown in Table 5, the unit cost of a *DRLIC* node is dominated by the soil moisture sensor, which was chosen in our experiments due to its very high accuracy (±3% [5]). As the size of the system scales up, the initial cost of the system may become impractically high. To scale to very large systems, then, we must consider the use of significantly cheaper soil moisture sensors at the price of slightly less accurate measurements [51]. Imaging techniques will help the sensor cost if they are accurate enough.

7 RELATED WORK

ET-Based Irrigation Control. In order to utilize weather data for irrigation control, various systems have been developed. These systems recognize the importance of weather as a primary source or sink of water in irrigated areas. The simplest among these systems employ standard fixed-schedule irrigation but incorporate a precipitation sensor to override the control and conserve water during rainfall [29]. However, the industry standard has shifted towards more sophisticated systems that rely on ET — an estimate of water lost through evaporation and plant transpiration — to efficiently replace water loss [8, 30].

To mimic the functionality of an ET controller, farmers can query a local weather station [4] to obtain the previous day's ET losses, provided in units of surface water height. With this information, they can use the sprinkler datasheet, which specifies the surface application rate, to calculate the precise duration for which the irrigation system should be activated to replenish the previous day's losses. In a commercial ET controller, this calculated amount represents the actual irrigation applied. Some providers claim an average reduction of 30% in water consumption using such systems.

However, despite their effectiveness, ET-based systems are constrained by centralized control, similar to other industry irrigation systems. Consequently, they are unable to offer site-specific irrigation, limiting their potential for optimizing system efficiency and achieving precise control quality. This lack of site specificity hampers their ability to adapt to the unique characteristics and requirements of individual areas.

Sensor-Based Irrigation Control. Advancements in soil moisture sensing technology have paved the way for the development of irrigation controllers that respond directly to soil moisture levels, leading to more accurate and efficient irrigation practices [7, 14, 25, 33]. These controllers utilize moisture sensors that are installed in the root zone of plants and provide real-time data on soil moisture to the controller. Based on this information, the controller can dynamically adjust the pre-programmed watering schedule to ensure optimal irrigation.

There are two main types of soil moisture sensor–based irrigation systems. (1) Suspended cycle irrigation systems: These systems utilize traditional timed controllers and automated watering schedules with predefined start times and durations. The key difference is that when there is sufficient moisture in the soil, the system will suspend or skip the next scheduled irrigation cycle. This approach helps prevent overwatering and conserves water resources. (2) Water on-demand irrigation: This type of system eliminates the need for programming irrigation durations and focuses on specifying only the start times for watering. It relies on two soil moisture thresholds — a lower threshold to initiate watering and an upper threshold to terminate watering [25]. By maintaining moisture levels between these thresholds, the system ensures efficient irrigation practices.

It is important to note that in the absence of a comprehensive model that accounts for water loss mechanisms, these moisture thresholds are typically set based on experiential knowledge rather than optimization techniques.

Model-Based Irrigation Control. In studies [58, 59], researchers develop a mechanistic **partial differential equation (PDE)** model to simulate moisture movement in irrigated areas, enabling the determination of an optimal watering schedule to maintain appropriate moisture levels. However, this PDE model has limitations, as it is not updated over time and does not account for future weather predictions. To address these limitations, the same authors further enhance the control system in subsequent works [56, 57]. In the improved system described in [56, 57], the PDE model is replaced with an adaptive approach that involves models trained from sensor data. Both long-term and short-term models are developed to describe the relationship between runoff from sprinklers and the movement of water through the soil.

It is important to note that the system [56, 57] is specifically designed for turf irrigation and may not provide significant benefits in shrubbery or tree irrigation. This is because turf soil moisture is influenced by factors such as water runoff on the soil surface and the overlapping coverage of sprinklers, which the models in [14, 43] focused on capturing. In the case of tree irrigation, there is minimal runoff due to the spacing between trees, and the soil moisture model needs to consider the soil–water relationship at different depths to be effective. Moreover, for almond orchards, which typically have thousands of trees, it is not feasible to install controllers for each sprinkler due to the high cost. Instead, our irrigation system operates on a region-based approach, where there are no sprinkler overlap relationships between different regions.

Additionally, a study [39] demonstrates that the decay of volumetric water content derived from the long-term model proposed in [14] was faster than what occurs in real-world scenarios. This discrepancy would result in light and frequent irrigation, which has been found to be inefficient [2]. Thus, there is a need to refine and improve the modeling techniques to better align with the requirements and characteristics of tree irrigation, especially in almond orchards.

DRL-Based Control. DRL has found applications in various domains, including autonomous mobility-on-demand systems [27, 28], sensor energy management [22], mobile app prediction [49], and building energy optimization [17, 19]. These applications demonstrate the versatility and effectiveness of DRL in solving complex problems and optimizing resource allocation in diverse fields. In particular, DRL techniques have demonstrated the potential optimal irrigation controls. The authors of [52] proposed an RL-based irrigation control system. The basic idea is to use a reinforcement learning algorithm to perform both irrigation planning and control. Two **neural networks (NNs)** are also introduced to predict **DSSAT (Decision Support System for Agrotechnology Transfer)** [31] simulation results. DSSAT is the defacto standard model for crop growth. One NN inputs irrigation and weather information and predicts total soil water content; the other NN predicts crop yield given the daily total soil water content for an entire crop season. The prediction of crop yield is then used as the training data to train the RL model. This approach can achieve relatively precise irrigation and allows full automation of the irrigation process. However, it is restricted to a small state space and is difficult to scale to large problems. Therefore, accurate representation of the actual irrigation context is difficult, leading to loss of important information that is needed for optimizing irrigation decisions. To solve this small state space problem, a DRL-based irrigation control approach [62] is introduced for optimizing irrigation applications in terms of net return. This approach determines the amount of irrigation for each zone at each timestep, taking soil moisture, ET, precipitation probability, and crop growth stage into consideration. Compared with the previous approach using traditional RL, it can handle a much greater state space and a greater number of irrigation choices. However, this control method is still central-valve control and thus affects the production performance.

Testbed. Testbeds are widely used to study precision irrigation. There are several existing platforms with features required for irrigation control [6, 46, 55]. The Farmbot is a CNC-style mechanism that consists of a plot for vegetables and a modular set of interchangeable tools [6]. The tools can execute a variety of tasks, including soil-moisture sensing, RGB imaging, planting, weeding, and irrigating. The basic assembly kit is priced at USD $2595.00. The authors of [46, 55] presented

RAPIDMOLT, a modular, open-source testbed that enables real-time, fine-grained data collection and irrigation actuation. RAPIDMOLT costs USD $600.00 and has floor space of 0.37 m^2. The functionality of the platform is evaluated by measuring the correlation between plant growth (Leaf Area Index) and water stress (Crop Water Stress Index) with irrigation volume. Both of these two testbeds are used for vegetables, not for trees with much deep soil and large scale.

8 DISCUSSION

System Adaptability and Generalization: The varying soil moisture requirements and root absorption capacities across different plant types necessitate a robust system capable of adapting to a wide range of agricultural contexts. Our system demonstrated significant water-saving effectiveness in almond orchards. This success showcases the potential for DRL in optimizing irrigation strategies. To extend our model to other crops, the following strategies can be employed.

Model Retraining: To retrain the model for different crops using the framework described in this article, we may follow these steps.

(1) Data Collection: Collect data specific to the new crop type. This includes soil moisture levels, irrigation patterns, growth metrics, and environmental conditions. Ensure that the data collected covers a wide range of scenarios and growth stages for comprehensive learning.
(2) Preprocessing: Process the collected data to fit the model's input format. This includes normalizing values, handling missing data, and segmenting data into training and testing sets.
(3) Initialization: Initialize the DRL model with the parameters specified in this article. This setup includes defining the reward function to focus on water conservation and crop yield optimization for the specific crop.
(4) Training: Feed the new crop data into the model and begin the training process. Monitor the model's performance and adjust hyperparameters as necessary to improve learning efficiency and accuracy.
(5) Evaluation and Tuning: After training, evaluate the model using a separate set of test data. Assess its performance in terms of water savings and crop yield. Based on the results, fine-tune the model by adjusting its parameters or architecture.
(6) Deployment and Continuous Learning: Deploy the retrained model in a real-world setting. Continuously collect data from this deployment to facilitate ongoing learning and improvement of the model.

Transfer Learning: To utilize transfer learning for adapting the DRL model to different crops, consider the following steps, drawing from the methodology outlined in this article.

(1) Source Model Selection: Select a well-performing model trained on a crop with similar characteristics or environmental conditions as the target crop. This model serves as the source from which knowledge will be transferred.
(2) Feature Alignment: Ensure that the features relevant to the source crop are compatible with the target crop. Adjustments may involve aligning soil types, weather conditions, and crop physiology parameters.
(3) Transfer and Freeze Layers: Transfer the layers of the source model to the new model intended for the target crop. Typically, early layers capturing general features are kept frozen, whereas later layers are fine-tuned to adapt to the specific characteristics of the target crop.

(4) Fine-Tuning: With the new crop data, fine-tune the adjustable layers of the model. This process involves a smaller set of data and focuses on refining the model's predictions for the target crop's specific irrigation needs.
(5) Validation and Performance Assessment: Validate the fine-tuned model on a separate dataset of the target crop. Evaluate its performance in terms of irrigation efficiency, water conservation, and crop health, ensuring that the transfer has been successful.
(6) Iterative Improvement: Continuously improve the model through iterative fine-tuning and validation, adjusting as necessary to ensure optimal performance across different crop types.

By following these steps and leveraging the base knowledge from a related crop, transfer learning can effectively adapt the DRL model for diverse agricultural settings, enhancing its utility and efficiency in optimizing irrigation strategies. We plan to further investigate and elaborate on these adaptation strategies in future work.

Limited Testing Results of *DRLIC*: The 15-day period was crucial for demonstrating the proof of concept and the immediate effects of *DRLIC* on water efficiency. These results, while preliminary, are promising indicators of the system's potential. To enhance the robustness of our findings, we have explored the integration of historical weather and irrigation data into our analysis. This strategic inclusion allows us to simulate and assess the performance of the *DRLIC* under diverse hypothetical scenarios, including those involving extreme weather conditions. Our dedicated efforts are captured in the newly introduced Section 6.3, Simulation Results. In this section, we meticulously present the outcomes of *DRLIC* alongside two baseline systems, scrutinizing their performance across an entire growing season. This extended evaluation period provides us with valuable insights, fostering a comprehensive understanding of the effectiveness and suitability of these systems for long-term irrigation control.

Building upon the above analysis, we recognize the necessity for a more extensive evaluation that spans the entire lifecycle of almond trees. Almonds and the trees that grow them go through many stages in their annual lifecycle including the stunning bloom in March, "growing up" through the spring, and "cracking open" in summer. The year finishes with harvest spanning from mid-August to October, followed by shelling and sizing. Our future work involves extending the experimental period to cover a full growth cycle of almond trees, commencing from March to October. This holistic approach will enable us to capture a diverse range of environmental conditions and their nuanced effects on the performance of *DRLIC*.

9 CONCLUSIONS

We introduce *DRLIC*, an innovative irrigation system that leverages DRL to generate optimal control commands based on real-time soil water content, current weather conditions, and weather forecasts. Our approach encompasses several key techniques, including the customization of DRL states and rewards for achieving optimal irrigation, the development of a validated soil moisture simulator for efficient DRL training, and the implementation of a reliable irrigation module. To evaluate the effectiveness of the *DRLIC* system, we design specialized irrigation nodes and construct a testbed featuring six almond trees. Through extensive experiments conducted in both real-world and simulated environments, we demonstrate the remarkable efficiency and performance of the *DRLIC* system.

REFERENCES

[1] 2010. Field Capacity. Retrieved from https://nrcca.cals.cornell.edu/soil/CA2/CA0212.1-3.php
[2] 2014. Light and Frequent Irrigation. Retrieved from https://www.usga.org/course-care/water-resource-center/our-experts-explain--water/is-it-better-to-irrigate-light-and-frequent-or-deep-and-infreque.html

[3] 2022. Soil Quality Indicators. Retrieved from https://www.nrcs.usda.gov/Internet/FSE_DOCUMENTS/nrcs142p2_053288.pdf

[4] 2023. California Department of Water Resources. Retrieved from https://www.cimis.water.ca.gov/

[5] 2023. Decagon Devices. Retrieved from http://www.decagon.com/products/soils/.

[6] 2023. Open-Source CNC Farming. Retrieved from https://farm.bot/

[7] 2023. UGMO Irrigation. Retrieved from http://www.ugmo.com/

[8] Richard G. Allen, Luis S. Pereira, Dirk Raes, and Martin Smith. 1998. Crop evapotranspiration—Guidelines for computing crop water requirements—FAO Irrigation and drainage paper 56. *Fao, Rome* 300, 9 (1998), D05109.

[9] Zhiyu An, Xianzhong Ding, and Wan Du. 2024. Go beyond black-box policies: Rethinking the design of learning agent for interpretable and verifiable HVAC control. *arXiv preprint arXiv:2403.00172* (2024).

[10] Zhiyu An, Xianzhong Ding, Arya Rathee, and Wan Du. 2023. CLUE: Safe model-based RL HVAC control using epistemic uncertainty estimation. In *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 149–158.

[11] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, and Chris Hesse. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680* (2019).

[12] Guotao Cui and Jianting Zhu. 2018. Infiltration model based on traveling characteristics of wetting front. *Soil Science Society of America Journal* 82, 1 (2018), 45–55.

[13] Guotao Cui and Jianting Zhu. 2018. Prediction of unsaturated flow and water backfill during infiltration in layered soils. *Journal of Hydrology* 557 (2018), 509–521.

[14] Sumon Datta and Saleh Taghvaeian. 2023. Soil water sensors for irrigation scheduling in the United States: A systematic review of literature. *Agricultural Water Management* 278 (2023), 108148.

[15] Dilini Delgoda, Hector Malano, Syed K. Saleem, and Malka N. Halgamuge. 2016. Irrigation control based on model predictive control (MPC): Formulation of theory and validation using weather forecast data and AQUACROP model. *Environmental Modelling & Software* (2016).

[16] Dilini Delgoda, Syed K. Saleem, Hector Malano, and Malka N. Halgamuge. 2016. Root zone soil moisture prediction models based on system identification: Formulation of the theory and validation using field and AQUACROP data. *Agricultural Water Management* (2016).

[17] Xianzhong Ding, Alberto Cerpa, and Wan Du. 2023. Exploring deep reinforcement learning for holistic smart building control. *ACM Transactions on Sensor Networks* (2023).

[18] Xianzhong Ding and Wan Du. 2022. DRLIC: Deep reinforcement learning for irrigation control. In *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 41–53.

[19] Xianzhong Ding, Wan Du, and Alberto E. Cerpa. 2020. Mb2c: Model-based deep reinforcement learning for multi-zone building control. In *Proceedings of the 7th ACM International Conference on Systems for Energy-efficient Buildings, Cities, and Transportation*.

[20] Bing Dong, Yapan Liu, Hannah Fontenot, Mohamed Ouf, Mohamed Osman, Adrian Chong, Shuxu Qin, Flora Salim, Hao Xue, and Da Yan. 2021. Occupant behavior modeling methods for resilient building design, operation and policy at urban scale: A review. *Applied Energy* 293 (2021), 116856.

[21] Bing Dong, Romana Markovic, Salvatore Carlucci, Yapan Liu, Andreas Wagner, Antonio Liguori, Christoph van Treeck, Dmitry Oleynikov, Elie Azar, and Gianmarco Fajilla. 2022. A guideline to document occupant behavior models for advanced building controls. *Building and Environment* 219 (2022), 109195.

[22] Francesco Fraternali, Bharathan Balaji, Dhiman Sengupta, Dezhi Hong, and Rajesh K. Gupta. 2020. Ember: Energy management of batteryless event detection sensors with deep reinforcement learning. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 503–516.

[23] Shmulik P. Friedman. 2023. Is the crop evapotranspiration rate a good surrogate for the recommended irrigation rate?*Irrigation and Drainage* (2023).

[24] Julian Fulton, Michael Norton, and Fraser Shilling. 2019. Water-indexed benefits and impacts of California almonds. *Ecological Indicators* 96 (2019), 711–717.

[25] G. L. Grabow, I. E. Ghali, R. L. Huffman, G. L. Miller, D. Bowman, and A. Vasanth. 2013. Water application efficiency and adequacy of ET-based and soil moisture–based irrigation controllers for turfgrass irrigation. *Journal of Irrigation and Drainage Engineering* 139, 2 (2013), 113–123.

[26] George H. Hargreaves and Zohrab A. Samani. 1985. Reference crop evapotranspiration from temperature. *Applied Engineering in Agriculture* 1, 2 (1985), 96–99.

[27] Sihong He, Shuo Han, and Fei Miao. 2023. Robust electric vehicle balancing of autonomous mobility-on-demand system: A multi-agent reinforcement learning approach. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 5471–5478.

[28] Sihong He, Yue Wang, Shuo Han, Shaofeng Zou, and Fei Miao. 2023. A robust and constrained multi-agent reinforcement learning electric vehicle rebalancing method in AMoD systems. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 5637–5644.

[29] Hunter. 2023. Hunter Rain-Clik rain detection. Retrieved from hunterindustries.com/irrigation-product/sensors/rain-clikr

[30] Marvin Eli Jensen, Robert D. Burman, and Rick G. Allen. 1990. Evapotranspiration and irrigation water requirements. ASCE.

[31] James W. Jones, Gerrit Hoogenboom, Cheryl H. Porter, Ken J. Boote, William D. Batchelor, L. A. Hunt, Paul W. Wilkens, Upendra Singh, Arjan J. Gijsman, and Joe T. Ritchie. 2003. The DSSAT cropping system model. *European Journal of Agronomy* (2003).

[32] Sham M. Kakade. 2001. A natural policy gradient. *Advances in Neural Information Processing Systems* 14 (2001).

[33] Yunseop Kim, Robert G. Evans, and William M. Iversen. 2008. Remote sensing and control of an irrigation system using a distributed wireless sensor network. *IEEE TIM* (2008).

[34] Devanshu Kumar, Xianzhong Ding, Wan Du, and Alberto Cerpa. 2021. Building sensor fault detection and diagnostic system. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 357–360.

[35] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. 2018. RLlib: Abstractions for distributed reinforcement learning. In *International Conference on Machine Learning*. PMLR, 3053–3062.

[36] Miaomiao Liu, Xianzhong Ding, and Wan Du. 2020. Continuous, real-time object detection on mobile devices without offloading. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 976–986.

[37] Camilo Lozoya, Carlos Mendoza, Leonardo Mejía, Jesús Quintana, Gilberto Mendoza, Manuel Bustillos, Octavio Arras, and Luis Solís. 2014. Model predictive control for closed-loop irrigation. *IFAC Proceedings Volumes* (2014).

[38] Artem Molchanov, Tao Chen, Wolfgang Hönig, James A. Preiss, Nora Ayanian, and Gaurav S. Sukhatme. 2019. Sim-to-(multi)-real: Transfer of low-level robust control policies to multiple quadrotors. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 59–66.

[39] Akshay Murthy, Curtis Green, Radu Stoleru, Suman Bhunia, Charles Swanson, and Theodora Chaspari. 2019. Machine learning-based irrigation control optimization. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 213–222.

[40] Haoyu Niu, Dong Wang, and YangQuan Chen. 2020. Estimating actual crop evapotranspiration using deep stochastic configuration networks model and UAV-based crop coefficients in a pomegranate orchard. In *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping V*. International Society for Optics and Photonics.

[41] Almond Board of California. 2020. Almond Irrigation Improvement Continuum. Retrieved from https://www.almonds.com/sites/default/files/2020-02/Almond-Irrigation-Improvement-Continuum.pdf

[42] Almond Board of California. 2020. Water Footprint for Almonds. Retrieved from https://almonds.com/sites/default/files/2020-05/Water_footprint_plus_almonds.pdf

[43] Su Ki Ooi, Iven Mareels, Nicola Cooley, Greg Dunn, and Gavin Thoms. 2008. A systems engineering approach to viticulture on-farm irrigation. *IFAC Proceedings Volumes* (2008).

[44] Luis S. Pereira, Paula Paredes, D. J. López-Urrea, and N. Jovanovic. 2021. Updates and advances to the FAO56 crop water requirements method. *Agricultural Water Management* (2021).

[45] R. Troy Peters, Kefyalew G. Desta, and Leigh Nelson. 2013. Practical use of soil moisture sensors and their data for irrigation scheduling. (2013).

[46] Mark Presten and Ken Goldberg. 2022. Design and Implementation of Physical Experiments for Evaluation of the AlphaGarden: An Autonomous Polyculture Garden. (2022).

[47] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *ICML*. PMLR.

[48] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[49] Zhihao Shen, Kang Yang, Wan Du, Xi Zhao, and Jianhua Zou. 2019. DeepAPP: A deep reinforcement learning framework for mobile application usage prediction. In *Proceedings of the 17th ACM Conference on Embedded Networked Sensor Systems (SenSys'19)*.

[50] Bruno Silva Ursulino, Suzana Maria Gico Lima Montenegro, Artur Paiva Coutinho, Victor Hugo Rabelo Coelho, Diego Cezar dos Santos Araújo, Ana Cláudia Villar Gusmão, Severino Martins dos Santos Neto, Laurent Lassabatere, and Rafael Angulo-Jaramillo. 2019. Modelling soil water dynamics from soil hydraulic parameters estimated by an alternative method in a tropical experimental basin. *Water* 11, 5 (2019), 1007.

[51] Egbert J. A. Spaans and John M. Baker. 1992. Calibration of watermark soil moisture sensors for soil matric potential and temperature. *Plant and Soil* (1992).

[52] Lijia Sun, Yanxiang Yang, Jiang Hu, Dana Porter, Thomas Marek, and Charles Hillyer. 2017. Reinforcement learning control for water-efficient agricultural irrigation. In *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*. IEEE.

[53] Yenny Fernanda Urrego-Pereira, Antonio Martínez-Cob, and Jose Cavero. 2013. Relevance of sprinkler irrigation time and water losses on maize yield. *Agronomy Journal* (2013).

[54] Kyriakos G. Vamvoudakis and Frank L. Lewis. 2010. Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica* 46, 5 (2010), 878–888.

[55] Marius Wiggert, Leela Amladi, Ron Berenstein, Stefano Carpin, Joshua Viers, Stavros Vougioukas, and Ken Goldberg. 2019. RAPID-MOLT: A Meso-scale, Open-source, Low-cost Testbed for Robot Assisted Precision Irrigation and Delivery. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE.

[56] Daniel A. Winkler, Miguel Á Carreira-Perpiñán, and Alberto E. Cerpa. 2018. Plug-and-play irrigation control at scale. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 1–12.

[57] Daniel A. Winkler, Miguel Á Carreira-Perpiñán, and Alberto E. Cerpa. 2020. OPTICS: OPTimizing Irrigation Control at Scale. *ACM Transactions on Sensor Networks (TOSN)* 16, 3 (2020), 1–38.

[58] Daniel A. Winkler, Robert Wang, Francois Blanchette, Miguel Carreira-Perpinán, and Alberto E. Cerpa. 2016. MAGIC: Model-based actuation for ground irrigation control. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 1–12.

[59] Daniel A. Winkler, Robert Wang, Francois Blanchette, Miguel A. Carreira-Perpinan, and Alberto E. Cerpa. 2019. DICTUM: Distributed Irrigation aCtuation with Turf hUmidity Modeling. *ACM Transactions on Sensor Networks (TOSN)* 15, 4 (2019), 1–33.

[60] Kang Yang, Yuning Chen, Xuanren Chen, and Wan Du. 2023. Link quality modeling for LoRa networks in orchards. In *Proceedings of the 22nd International Conference on Information Processing in Sensor Networks*. 27–39.

[61] Kang Yang, Yuning Chen, and Wan Du. 2024. OrchLoc: In-orchard localization via a single LoRa gateway and generative diffusion model-based fingerprinting. In *Proceedings of the 22nd ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'24)*.

[62] Yanxiang Yang, Jiang Hu, Dana Porter, Thomas Marek, Kevin Heflin, and Hongxin Kong. 2020. Deep reinforcement learning-based irrigation scheduling. *Transactions of the ASABE* (2020).

[63] Hang Zhu, Varun Gupta, Satyajeet Singh Ahuja, Yuandong Tian, Ying Zhang, and Xin Jin. 2021. Network planning with deep reinforcement learning. In *Proceedings of the ACM SIGCOMM 2021 Conference*.

Received 7 July 2023; revised 17 January 2024; accepted 21 April 2024