OPTICS: OPTimizing Irrigation Control at Scale

DANIEL A. WINKLER, MIGUEL Á. CARREIRA-PERPIÑÁN, and ALBERTO E. CERPA,
University of California, Merced

Lawns, also known as turf, cover an estimated 128,000 km² in North America alone, with landscape requirements representing 30% of freshwater consumed in the residential domain. With this consumption comes a large amount of environmental, economic, and social incentive to make turf irrigation systems as efficient as possible. Recent work introduced the concept of distributed control in irrigation systems, but existing control strategies either do not take advantage of the distributed control, or do not revise the strategy over time in response to collected data. In this work, we introduce OPTICS, a data-driven control strategy that self-improves over time, adapts to the local specific conditions and weather changes, and requires virtually no human input in both setup and maintenance providing a *plug-and-play* system that requires minimal pre-deployment efforts. In addition to substantial improvements in ease-of-use, we find across 4 weeks of large-scale irrigation system deployment that OPTICS improves system efficiency by 12.0% in comparison to industry best and 3.3% in comparison to academic state of the art. Despite using less water, OPTICS also was found to improve quality of service by a factor of 4.0× compared to industry best and 2.5× compared to academic state of the art.

CCS Concepts: • **Applied computing** → *Agriculture*; Environmental sciences; • **Computer systems organization** → **Sensors and actuators**; *Sensor networks*; • **Information systems** → *Sensor networks*; Process control systems; • **Theory of computation** → *Linear programming*; • **Computing methodologies** → *Multiscale systems*; Discrete-event simulation;

Additional Key Words and Phrases: Irrigation, monitoring, decentralized control, experimentation, measurement, performance

ACM Reference format:

Daniel A. Winkler, Miguel Á. Carreira-Perpiñán, and Alberto E. Cerpa. 2020. OPTICS: OPTimizing Irrigation Control at Scale. *ACM Trans. Sen. Netw.* 16, 3, Article 22 (May 2020), 38 pages.
<https://doi.org/10.1145/3372024>

1 INTRODUCTION

Accessible fresh water is estimated to make up just 1% of all water on Earth's surface [4], and lawn irrigation is estimated to consume roughly 7 billion gallons per day [30]. Due to the scale of this usage, there is much economic, environmental, and social pressure to improve the efficiency of these systems as much as possible.

A previous version of this work titled "Plug-and-play Irrigation Control at Scale" [39] has appeared in the Proceedings of the 17th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN 2018).

Authors' addresses: D. A. Winkler, M. Á. Carreira-Perpiñán, and A. E. Cerpa, University of California, Merced, 5200 North Lake Road, Merced, CA 95344; emails: dwinkler2@andes.ucmerced.edu, mcarreiraperpinan@ucmerced.edu, acerpa@andes.ucmerced.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.

1550-4859/2020/05-ART22 \$15.00

<https://doi.org/10.1145/3372024>

Although system efficiency (minimizing costs) is a key selling point, the primary goal of these systems is to maintain healthy turf, and this must be done carefully. An under-watered plant will eventually wilt and die, making them aesthetically unpleasant, but over-watering can cause many issues as well. Consistently over-saturated soil can cause turf roots to rot and soil to erode, and in extreme cases excessive irrigation can carry fertilizer chemicals deep beyond the root zone into drinking water sources, as has occurred in California's Salinas Valley [27]. However, as aesthetic symptoms of under-watering are more pronounced, many irrigation systems over-water by design to ensure that wilting does not occur.

These issues can all be avoided if each location in the irrigated space receives just the water it requires. Complex models exist that predict the movement of moisture across and through the soil of an irrigated space, but each of these models must make assumptions. Most commonly, due to the difficulty of sampling soil type, soil depth, direct solar irradiance, and other key factors across an irrigated space, controllers are forced to assume that water must move in a uniform way until it settles. Coupled with the centrally located water valve that is industry standard, it is impossible to control in such a way that all locations in the space are adequately irrigated, while also minimizing water consumption.

Some of these limitations are addressed by the distributed sensing/actuation node proposed in Reference [38] and developed in Reference [40], which allows each sprinkler in the irrigation system to independently actuate based on a wirelessly transmitted schedule and monitor local soil moisture conditions in real time, allowing more efficient control routines to be developed. However, the proposed control framework requires manual model generation and offers no model correction, limiting its scalability. In considering the future of smart irrigation using wireless sensor networks, References [32, 36] concluded that key system goals are scalability, possible by minimizing system complexity with automation, reasonable costs, and the ability to close the loop by gathering data, modeling, and making/applying control decisions. Toward this end, the control system in this work is tailored to the space in a data-driven way that requires no human intervention. This allows us to not only deploy the system with ease and minimal configuration, but perhaps more importantly, it allows us to *learn and adapt* to the local conditions experienced in the field. OPTICS does not require cumbersome measurements of soil type, topography, direct solar irradiance, but rather adapts to the conditions measured from the moisture data, even if the conditions are heterogeneous across the field.

In this work, OPTICS uses this alternate approach to solve this very complex problem. We argue that a model adaptively trained from data will react to unforeseen conditions better than a system using a mechanistic model with approximated parameters and fixed assumptions. The contributions of this work are as follows: (1) As no manual input is required of the installer, OPTICS is a truly plug-and-play system, avoiding costly expertise to determine environmental characteristics (e.g., soil characteristics, topography, solar exposure), which can be difficult or infeasible to measure accurately at scale; (2) constant model re-training with fresh data allows OPTICS to automatically adapt to unforeseen/changing environmental conditions and seasonal variations, and weather forecasting allows OPTICS control to react to future weather conditions; (3) to improve system scalability, OPTICS decouples short- and long-term models, allowing the latter to become spatially independent; (4) as OPTICS closes the loop, the lightweight learning model reduces the computational complexity, allowing us to compute optimal schedules in a timely manner without significant computational resources while maintaining overall accuracy. In 4 weeks of deployment during the summer and fall, we demonstrate that in addition to these significant improvements to ease-of-use and scalability, OPTICS reduces water consumption against all baselines, while simultaneously improving quality of irrigation.

Fig. 1. Sample fluid curve across the 24-hour cycle.

2 SYSTEM OVERVIEW

Our system takes advantage of a distributed irrigation control system, with sensing/actuation nodes installed beneath each sprinkler. Each node is equipped with a wireless sensing mote [15] providing minor computational capability and wireless communications, a volumetric water content (VWC) sensor to sense local conditions, and a solenoid, allowing the opening and closing of water to the sprinkler on command. These devices form a mesh network and are accessible through a border router, a special node physically connected to a nearby internet-accessible computer. Sprinkler schedules are sent outbound along this link, and real-time data from the sensing nodes are sent inbound along this link, allowing us to automate the system with any strategy.

The goal of our irrigation system is to keep the turf healthy and to do so, a number of requirements must be satisfied. Adequate solar exposure must be provided, the soil must contain the correct types of nutrients in appropriate amounts, and an adequate amount of moisture must be provided to the soil to be absorbed by the plant roots. Although our irrigation system has no control over solar exposure and soil nutrient composition, it has direct control on the application of water onto the surface of the soil. In plant physiology, two primary thresholds of volumetric water content (VWC) will determine the happiness of the plant. The first represents the level where the plant will no longer be able to extract water from the soil and is known as the Permanent Wilting Point, or θ_{pwp} [15,31,33]; if a plant spends an extended period of time in soil beneath this moisture level, then it will begin to wilt and die. The second threshold more conservatively describes the level of moisture below which the plant *starts* feeling stressed and is a fraction p of the total available water in the soil, where p is tailored to the particular crop type in question [19]. While the more conservative no-stress threshold is certainly more applicable in the installation of a more decorative turfgrass, we chose the more restrictive θ_{pwp} moisture threshold in this work, as it will better demonstrate the OPTICS system's ability to maintain even the most challenging moisture levels. While this threshold can be easily modified in the system depending on needs, the goal of the OPTICS system is the same under any of them; to minimize the amount of time the system spends beneath the chosen minimum threshold, as verified in our system with the installation of VWC sensors to constantly monitor soil moisture levels. By doing so, we will be minimizing the plant's exposure to unhealthy moisture conditions, giving it the best possible chance to thrive. We note that θ_{pwp} is the *only* soil or topographic parameter that must be measured or estimated in order for OPTICS system to operate.

Figure 1 shows an example 24-hour irrigation cycle. After day 1's irrigation ends at $t_{f,1}$, the water begins discharging from the soil subject to the effects of diffusion, leaching, and weather effects, which occur very slowly relative to effects during irrigation. After hours of these losses, the soil VWC will reach its minimum value just before the start of irrigation at day 2, $t_{s,2}$. The

Fig. 2. OPTICS system architecture.

effects of diffusion, leaching, and weather are not homogeneous; differences in environmental factors such as soil type, soil depth, and solar irradiance change the rate at which moisture is lost across the space. Over time, we use the historical loss trends to build a *long-term model* (LTM) that characterizes the way water is lost *between* irrigation cycles for each individual sensing node. With knowledge of the desired minimum moisture threshold θ_{pwp} , we wish to maintain in the soil and our *long-term model* describing losses that are expected to occur, we can work backwards to intelligently choose a *Goal State* of moisture that we wish to reach by the end of irrigation. By reaching this *Goal State*, we ensure we will stay above minimum VWC θ_{pwp} across the full 24-hour cycle without wasting water.

Between day 1's start of irrigation $t_{s,1}$ and finish of irrigation $t_{f,1}$ as shown in Figure 1, the water applied to the space gradually increases VWC within the soil. $t_{f,1}$, the time that irrigation ends is not known beforehand, but occurs when VWC has reached the pre-computed *Goal State* for this sensor node. During this irrigation time, we utilize a data-driven *short-term model* (STM) describing how the actuation of one or more sprinklers affects the moisture in the soil at all sensor locations, and solve an optimization problem to find the most efficient irrigation schedule to reach the *Goal State* at each sensor location, allowing us to save water by utilizing sprinkler overlap, soil runoff, and schedule intermittency to our advantage. The de-coupling of these two models allows this technique to scale to control very large irrigation systems. The two are intertwined, as one produces the *Goal State* used by the other. In the short-term, the model and optimization takes into account the effects of all nodes' sprinkler coverage jointed spatially and temporally, but once irrigation ends, the models describing losses across the field become spatially independent.

Figure 2 shows the data processing required to achieve these goals at irrigation-time (daily irrigation at dusk, by request of campus groundskeepers). First occurs model generation. The freshest data from the irrigated space is used to build the long- and short-term models for use in loss prediction and irrigation schedule optimization. The short-term model describes the direct in-flow of moisture as sprinkler moisture lands above the sensor, and takes into account sprinkler overlap and water runoff effects. The long-term model shows how the moisture tends to move across the full 24-hour cycle due to soil transport effects such as diffusion and leaching [23], and a separate weather prediction module predicts future weather trends in the form of reference evapotranspiration [28].

Next, using the long-term model, the *Expected Losses* module computes a *Goal State* for each node in the space as shown in Equation (1). This state is computed by taking the minimum acceptable VWC and adding the node's expected moisture losses between irrigation cycles. As later explained in Section 3.2, future weather predictions are decoupled from the long-term model, so forecasted evapotranspiration losses are also added at this stage, computed as described in Section 3.1. In this way, the minimum moisture, experienced right before the next irrigation will begin, should

Fig. 3. Data communication across irrigation system with distributed actuation.

be at or just above the minimum VWC threshold. This *Goal State* is later used by the optimization module:

$$Goal\ State = \theta_{pwp} + Expected\ Losses + ET_{forecasted}. \quad (1)$$

Once these initial conditions are defined for irrigation, the control loop is entered, which will fetch the freshest data snapshot from all nodes across the space. This data is used as the initial moisture conditions for an optimization problem that computes the optimal actuation sequence for each individual node in the space, such that each node's *Goal State* VWC will be reached by the end of the irrigation period. If our short-term model used in optimization is accurate, then we can perform this control loop only once at the beginning of irrigation. However, for safety and for accuracy, it is advantageous to occasionally re-run this control loop during the irrigation period using the most current VWC data across the space to help correct for drift caused by errors in the *short-term model*. This prevents schedules from over- or under-watering due to stale starting conditions and will allow the optimizer to re-search for more efficient schedules from the freshest starting point.

In our deployment, the model generation and optimization takes place on a computationally weak Raspberry Pi [14] that is collocated with our irrigation system. Attached to the Raspberry Pi via USB is a TMote Sky [15], through which schedules can be sent and data can be received. Once optimization is complete, the schedules produced by the optimizer are sent over USB and through the wireless sensor network to their respective sensing/actuation node. The schedule is run, and if desired, a fresher data snapshot will be obtained and optimization will occur again.

To maintain a fresh snapshot of the soil moisture content at all times, the distributed sensor/actuator devices are constantly polling their attached sensors. During irrigation time, when fresh data is crucial, the sampling frequency is set to be equal to the *Control Timestep* of the actuator, chosen in our system to be 1 minute. When irrigation is complete, the sampling period is reduced to once every 10 minutes to maintain a long battery life. Sampling across the full 24-hour cycle is useful for us to evaluate the control strategies we are testing, but in a commercial system it may only be necessary to collect data during irrigation, consuming less sensing and radio transmission energy and extending system lifetime. When sensing is occurring at a 10-minute timestep, we later post-process by linear interpolation to ensure we have a consistent sampling period throughout the entire day. This data is routed from the sensing nodes across the wireless sensor network to the base station as shown in Figure 3 for use in modeling, visualization, and so on. Due to the relatively small distances of our test irrigation system, only a single-hop wireless network was required to communicate from sensing nodes to the base station, as later discussed in Sections 6.4 and 9.

Control on the distributed sensor/actuator devices also operates at the same *Control Timestep*, chosen in our system to be 1 minute. Individualized valve schedules are routed from the base station across the wireless sensor network as shown in Figure 3, each containing a control timestep and a binary sequence of actuations for a particular sensor/actuator node. Once received, the sensing/actuation node will then route a pulse of either forward or reverse current to enable or disable the connected solenoid, following the provided schedule. If a schedule is received while another is already running, then the first is cancelled and overwritten by the fresher one.

3 SYSTEM MODELING

The purpose of the control system is to decide how much moisture must be applied to the surface by the sprinklers. Whereas standard irrigation controllers use weather-only or rule-of-thumb techniques to make this decision, we hope to leverage the rich, spatially distributed data collected by our sensing/actuation nodes to make this process more efficient. Toward this goal, we generate data-driven models that help us understand how moisture tends to move in the short- and long-term.

The water used for irrigation moves through the space subject to many factors, which all tend to occur within two different time horizons. In the short term, *during* irrigation, factors such as sprinkler distribution and water runoff on the surface of the soil cause water movement that will occur for seconds or minutes, movement that tends to come to an end when irrigation is completed or soon thereafter. Once this moisture infiltrates fully into the soil, much slower effects continue to take place in the long term *between* irrigation periods. For instance, depending on the type of soil, leaching of water beyond the root zone and diffusion can occur on the order of 10^{-2} – 10^{-3} cm/s [19], and will continue to move for hours or days.

Previous work [40] combines all of these moisture movement factors together in one large mathematical model based on first principles. However, the size of the resulting models have performance repercussions, and the lack of model correction could potentially cause it to deviate from reality. In this work, we solve these problems by modeling water movement using a lighter data-driven approach based on machine learning techniques. The implicit advantage of this approach is that it is based on moisture data measured locally, being able to cope with heterogeneous conditions like soil type, topography, and solar exposures that vary across the field. To reduce the computational complexity of the model-based optimization problem to minimize water consumption subject to quality constraints, we chose to use a two-model approach, each of which represent one of the distinct time horizons. The long-term model, which learns how water tends to be lost between irrigation cycles, is used to compute a *Goal State*, the required moisture level that must be reached at each location in the field so that moisture levels will not be depleted below our minimum moisture threshold before the next irrigation period. The short-term model is used by the *Schedule Optimization Module* to determine the best schedules that will take advantage of runoff, overlapping sprinkler coverage, and other short-term effects to compute schedules that bring moisture to the *Goal State* while consuming minimal water. With a cleverly chosen *Goal State*, we will maintain adequate moisture levels across the full 24-hour cycle, while only requiring optimization during the irrigation period, allowing optimization to be run on very computationally weak machines.

3.1 Weather Forecasting

A large portion of losses on the 24-hour cycle is expected to be caused by the combined environmental effects of evaporation and plant transpiration, referred to together as evapotranspiration [28] (ET). The industry-standard ET estimation uses four environmental factors, temperature, humidity, wind, and solar irradiance to estimate ET losses, or the water that has been removed from the soil due to weather conditions. Please note that this is the ET_0 or reference crop ET.

Fig. 4. Water inputs and outputs to reactive irrigation system.

The reference surface used in the calculation of ET_0 is a hypothetical turf reference crop with an assumed crop height of 0.12 m, a fixed surface resistance of 70 sm^{-1} and an albedo of 0.23. The reference surface closely resembles an extensive surface of green, well-watered turf of uniform height, actively growing and completely shading the ground. The fixed surface resistance of sm^{-1} implies a moderately dry soil surface resulting from about a weekly irrigation frequency. This matches the conditions in our experiments. If we were to apply OPTICS system to a different crop, then a K_c cropping factor should be used to adjust the reference ET_0 to the crop ET .

Evapotranspiration-based irrigation controllers are becoming industry standard, and the control strategy follows intuition. Weather stations in the region take periodic measurements of the four environmental factors, and make a calculation of evapotranspiration (ET) losses. The ET irrigation controller communicates with the weather station for an estimate of total losses that have occurred since the last irrigation. With knowledge of the sprinklers' water application rates, these losses are then used to decide the irrigation schedule to replace the losses that have occurred. However, this technique has a fundamental limitation, as the control can only react to past weather losses, and cannot prepare for future weather losses. Figure 4 shows example water inputs and outputs in an irrigation system with a strategy that reacts to the previous day's weather trends, where irrigation each day is chosen to match the losses of the previous day. Here, the water applied and lost is displayed in inches of water, the standard unit used to describe irrigation and precipitation inputs, and evapotranspiration losses. This unit describes a column of water on the surface, and when describing irrigation amount is a function of the sprinkler's distribution pattern and on-time. The resulting net change in water can be seen in Figure 5, where it can be seen that performing a direct water replacement results in the soil in general receiving a net loss in water content due to the system's inability to prepare for future losses. In practice, to ensure the plant receives enough water, commercial ET irrigation controllers provide a direct replacement of past losses *plus* a safety margin of extra water to guide the soil moisture toward adequate levels over time. This inexact method of overwatering by design leads to unnecessary water waste or reduced quality of service.

In contrast, if we have the ability to *predict* the losses before they occur, then we can pre-irrigate to ensure net moisture change is always positive or zero, without requiring an arbitrary safety band of extra irrigation. The loss/irrigation inputs and outputs of such a system can be seen in Figure 6, and the resulting net moisture change in the soil can be seen in Figure 7. Each day, the amount of irrigation is chosen based on the next day's *predicted* losses rather than past losses, which allows us to allocate the water ahead of time to prepare for or take advantage of weather that is expected to occur. In addition to avoiding unhealthy moisture levels in the soil, prediction will allow the system

Fig. 5. Net change in soil moisture content under reactive system.

Fig. 6. Water inputs and outputs to predictive irrigation system.

to save water if rain is forecasted in the near future, whereas a reactive system will continue to irrigate as usual. If more rain occurs than was predicted, then the OPTICS system will sense this through soil conditions and prevent wasteful irrigation from being applied to the space. If less rain occurs than was predicted, then the system will recuperate these losses after-the-fact in the same way a standard ET controller would.

Although *prediction* of future ET losses will allow an irrigation system to provide better quality of control and system efficiency, this data is not available in practice. However, we found that 15 years of historical data trends [1] of the 4 ET factors (temperature, humidity, wind, and solar irradiance) were available at hourly intervals at our location, as well as the daily ET losses that occurred on those days. Additionally, we consider the fact that in most locations, it is possible to find *forecasted* hourly weather trends (for at least temperature) for at least 24 hours into the future. To use the data we have available to provide future ET prediction, we trained a k-nearest-neighbors regression model to accept as input the hourly vectors of all (or a subset, depending on forecast availability at the system's location) of the 4 ET factors, and as output the resulting ET losses for the entire day. For instance, in our locale, we found that the Wunderground API [17] has hourly forecasting of both temperature and humidity. If the deployed system is not located near a weather station, then the National Oceanic and Atmospheric Administration (NOAA) provides temperature, humidity, wind speed and cloud cover prediction at hourly intervals for any

Fig. 7. Net change in soil moisture content under predictive system.

point in the United States [11]. Using our 15 years of historical data with approximately 5,475 data samples, we step through each day and create a 48-element feature vector composed of 24 hourly temperature readings and 24 hourly humidity readings, and as output the cumulative ET losses from this 24-hour period as a single value. We train our KNN on these features and outputs with a k -value of 5, so in prediction the regressor will use the 5 historical days with the most similar hourly weather trends to choose the evapotranspiration value that is likely to occur. Finally, each day we query Wunderground to get *tomorrow's* forecasted hourly temperature and humidity readings, which are then passed to the KNN regressor to predict the ET losses that are expected to occur tomorrow based on those trends, which is then used by the OPTICS controller to optimize irrigation schedules. Evaluation of the accuracy of this technique can be found in Section 7.1.

3.2 Long-term Model

To provide the highest quality of control, we wish to maintain at least a minimum level of moisture in the soil, θ_{pwp} , as discussed in Section 2. Although industry-standard irrigation systems choose a target on-time for the irrigation system, our rich coverage of soil moisture sensors allow us to instead set a *Goal State* of soil moisture. This *Goal State* is unique for each node in the space, acting as a target volumetric water content (VWC) for the *Schedule Optimization Module* to reach as shown in Figure 1. To choose a useful *Goal State*, we must consider both future predicted losses due to weather patterns (evapotranspiration) as discussed in Section 3.1 and environmental loss data trends from previous data for each individual node.

Once irrigation is complete and the irrigated moisture has infiltrated into the soil, it begins to move more slowly through the processes of diffusion, leaching, and weather patterns as shown in Figure 8. Diffusion is the tendency of moisture in the soil to move from areas of higher to lower concentration due to differences in hydrostatic pressure caused by intermolecular forces in the soil, and leaching is the tendency of water to eventually move beyond the root zone of the plant due to the force of gravity. Movement under either of these forces tends to be very slow, as hydraulic conductivity of water through certain soils can be on the order of 10^{-2} – 10^{-3} cm/s [19]. Losses due to evapotranspiration (ET) occur slowly as well, as they are primarily caused by solar radiation and high air temperature, which occur during daylight hours, or windy conditions.

Figure 9 shows how moisture losses occur at one selected sensor site across three consecutive days. This example curve is unique at this location due to its specific soil characteristics. In this figure, $t = 0$ is time-aligned to the end of irrigation where moisture losses start to occur, and $t = 21$ hours corresponds to the beginning of irrigation on the following day. By tracking these losses

Fig. 8. Sources of water movement *between* daily irrigation (Long-term).

Fig. 9. Sample moisture decay fit between irrigation.

using our deployed VWC sensors, we found that the discharge of water from the soil medium tends to occur as an exponential decay when irrigation ends as shown in the figure. In our system, we use these historical loss trends to fit an exponential decay curve as our “LTM,” or Long-term model, as shown in Figure 9. In addition, for each day $d \in \{1, \dots, D\}$, we record the measured evapotranspiration that occurred, ET_d . With this fitted model, the expected loss of this node is computed by taking the difference of the curve at $t = 0$ and at $t = t_{s,d+1} - t_{f,d}$, the expected delay between irrigation of today and tomorrow, e.g., 21 hours. This computed loss is finally offset by the average of the daily evapotranspiration levels measured during the training period:

$$\text{Expected Losses} = (\text{LTM}(0) - \text{LTM}(t_{s,d+1} - t_{f,d})) - \frac{1}{D} \sum_{d=1}^D ET_d. \quad (2)$$

We subtract the weather to ensure that the weather experienced when the training trends were recorded do not impact the expected losses for future irrigation. Before use as the *Goal State* for optimization, the *predicted* ET for the following day will be re-added as shown in Equation (1). We decouple these weather effects to prevent learning past weather trends into our model. In a climate where the weather changes very little from day-to-day, it may be reasonable to assume that past weather trends will continue into the future, but to allow OPTICS to be more generalizable and reactive, we choose to utilize local weather forecasting separately.

Fig. 10. Sources of water movement *during* irrigation (Short-term).

Table 1. Short-term Model Variables

Variable	Description
$t \in \{0, \dots, T\}$	Temporal index
K	Number of sensing/actuating nodes in system
$\mathbf{s}_t \in \mathbb{R}^K$	Vector of moisture levels at time t , size K
$\mathbf{f}_t \in \{0, 1\}^K$	Vector of binary sprinkler actuation at time t , size K

When the system is first turned on, the model has no understanding of expected losses. It is not until after the first day of irrigation, in our case a fixed schedule to train the short-term model as discussed in Section 3.3, that the long-term model can be trained. In our experiments, the model is retrained each day before irrigation using the loss data from all of the preceding days. That is to say, on day d , we re-train the model using $d - 1$ days of data. This worked well in our experiment, but in a very long-term installation, re-training the model on just the most recent N days of data may allow the system to be more responsive to changing seasonal conditions. In this work, we have not investigated the optimal choice of N , and we leave this for future work.

3.3 Short-term Model

As our system was designed to require no manual configuration from the installer, at install-time the OPTICS system does not understand which sprinklers and sensors are within range of each other. For later use in optimization in Section 4, we must understand how the actuation of the sprinklers in the system will influence the moisture in the soil as shown in Figure 1. For this task, we employ a *Short-Term Model*, which captures the moisture movement effects *during* irrigation. An example irrigated space is shown in Figure 10; differences in sprinkler overlap affect the amount of water that lands on the surface of the soil, surface topography affects how quickly runoff will occur, and heterogeneous soil composition and depth will affect the rate of infiltration. As it is difficult and error-prone to manually measure these effects, we wish to learn them in an automated, data-driven way.

We choose to use a linear regressor to model these effects as a more complex non-linear model would make optimization slower and more prone to local optima, as later discussed in Section 4. The model is defined using the variables enumerated in Table 1. As input, we provide the current VWC at each of the K sensor locations as vector $\mathbf{s}_t$, and the current binary actuation of each of the K sprinklers as vector $\mathbf{f}_t$. The output of the linear model is the predicted VWC for each of the K sensor locations at some time Δt in the future, as vector $\mathbf{s}_{t+\Delta t}$. In practice, as this model will be

Fig. 11. Sensor data from selected nodes during Short-term model training period.

later used to compute optimal irrigation schedules, the length of Δt is chosen to be the same as the control actuation period, 1 minute in our experiments. The linear function g defines the following relationship:

$$g(s_t, f_t) = s_{t+\Delta t}. \quad (3)$$

Ideally, we would train our model by running all possible combinations of sprinkler actuations and measuring the sensor response. However, as this would require an immense amount of time and potentially wasteful irrigation actuation to achieve, we instead chose to actuate each sprinkler one-by-one for model training, which can be done much more quickly. While the resulting model will have some error in moisture estimation when many sprinklers are active at once, the OPTICS system will continuously re-train this model as more complex schedules are used in irrigation, improving prediction over time.

In this way, each sprinkler is triggered one-by-one for a fixed time (1 hour in our case) and the resulting soil moisture is measured at all locations. Moisture levels during our training cycle can be seen for four selected devices in Figure 11 with rises caused by the activation of the nearest sprinkler, and although each sprinkler is active for the same amount of time, it can be seen that the amounts of increase are different for each device due to spatial heterogeneity of runoff, sprinkler coverage, and soil characteristics. Furthermore, as sprinkler overlap would cause multiple sensors to rise simultaneously as a single sprinkler is active, it can be seen that this irrigation system has minimal sprinkler overlap, as only one sensor rises at a time. In a system with more sprinkler overlap, the methods of data collection and processing would be identical.

The data from our deployment training period is parsed into ~ 850 training pairs of $[s_t, f_t]$ vectors as inputs, and $[s_{t+\Delta t}]$ vectors as outputs, and the regressor is trained. As sensor data is inevitably noisy it is important to choose a Δt that is not too big, which would result in increased predictive error or too small, where noise dominates the state signal. Our choice of 1 minute seemed to work well in practice, and further evaluation of this choice can be found in Section 7.2.

4 OPTIMIZATION OVER THE SCHEDULE

We wish to use our models as described in Section 3 to compute irrigation schedules that will allow us to reach our goal volumetric water content (VWC) on each sensor/actuator node while minimizing system water consumption. With a goal state, $\theta_{\text{goal},k}$ and recently measured VWC θ_k for each node index $k \in \{1, \dots, K\}$, we construct the following optimization problem using

Table 2. Optimization Variables

Variable	Description
$t \in \{0, \dots, T\}$	Temporal index
$k \in \{1, \dots, K\}$	Sprinkler location index
$\mathbf{s}_t \in \mathbb{R}^K$	Vector of moisture levels at time t , size K
$\mathbf{f}_t \in \{0, 1\}^K$	Vector of binary sprinkler actuation at time t , size K
$s_{kt} \in \mathbb{R}$	Volumetric water content (VWC) of location k at time t
$f_{kt} \in \{0, 1\}$	Sprinkler k actuation at time t
$c_k \in \mathbb{R}$	Consumption of sprinkler k (Constant, known beforehand)
$\theta_k \in \mathbb{R}$	Measured VWC of sensor k (Constant, known beforehand)

optimization variables as defined in Table 2:

$$\min_{\{f_{kt}, s_{kt}\}_{k=1, t=0}^{K, T}} \sum_{k=1}^K \sum_{t=0}^T c_k f_{kt} \quad \text{s.t.} \quad (4a)$$

$$0 \leq f_{kt} \leq 1 \quad k = 1, \dots, K \quad t = 0, \dots, T, \quad (4b)$$

$$s_{kt} \geq \theta_{\text{pwp}} \quad k = 1, \dots, K \quad t = 0, \dots, T-1, \quad (4c)$$

$$s_{kT} \geq \theta_{\text{goal}, k} \quad k = 1, \dots, K, \quad (4d)$$

$$s_{k0} = \theta_k \quad k = 1, \dots, K, \quad (4e)$$

$$\mathbf{s}_t = \mathbf{g}(\mathbf{s}_{t-1}, \mathbf{f}_{t-1}) \quad t = 1, \dots, T. \quad (4f)$$

We define f_{kt} to be the actuation of sprinkler k at discrete time index $t \in \{0, \dots, T\}$, relaxed to $[0, 1]$ in the optimization. The objective function in Equation (4a) is the sum of f_{kt} for all $k \in \{1, \dots, K\}$, $t \in \{0, \dots, T\}$, weighted by the water consumption rate of each sprinkler k , c_k , a function of the sprinkler's angle of coverage as defined in the sprinkler datasheet. With pressure-regulated sprinklers, this weighted sum represents the total system water consumption under schedule F, which we are trying to minimize. The VWC at each discrete sensor location k at temporal index t is defined as s_{kt} . This moisture level is constrained to remain above the minimum acceptable moisture threshold, θ_{pwp} at all times in Equation (4c), and above the goal state at final time index $t = T$ in Equation (4d). Equation (4e) assigns this state variable the most recently measured sensor value θ_k at starting time index $t = 0$. Changes in moisture level as a result of sprinkler actuation is modeled by the linear function $\mathbf{g}$ in Equation (4f), representing our *Short-term Model* as defined in Section 3.

As sprinkler valves can physically be either on or off, sprinkler actuation f_{kt} is a binary variable in practice. This makes the defined problem an integer linear program (ILP), which is NP-Complete. We have found that solving this problem with reasonable values of K and T can take as long as several minutes on our computationally weak base station. As this optimization may be required to run as often as once per control timestep, chosen in our system to be 1 minute, we find an approximated solution more quickly by treating f_{kt} as a real number within $[0, 1]$ as shown in Equation (4b), and then rounding the computed optimal value to the closest binary value. This simplification makes the resulting problem a linear program (LP), which is much simpler to solve in practice. Evaluation of this choice can be found in Section 7.3.

We found that with our irrigation system architecture and environmental characteristics, our irrigated space generally requires 30–60 minutes of irrigation to be sufficiently watered. To give our optimization time to find schedules that are as efficient as possible, we give the optimizer an

Fig. 12. Side-by-side experimental layout.

irrigation window of 2 hours, well above the required time of a simple schedule. With a control timestep of 1 minute, this 2-hour irrigation window is converted to $T = 120$. By allowing a larger irrigation window, the optimizer may find schedules that are more efficient, but due to University water pressure limitations, irrigation in different regions of the campus must operate within slots, making very large time windows impractical.

We chose to use the Julia programming language [20] as an interface to the GNU Linear Programming Kit (GLPK) [5] solver. We chose these tools for their ease of use and sufficient performance. Our linear program has $2K(T + 1)$ variables and $4K(T + 1)$ constraints. In our deployment, solving for a schedule with a selected setup of $K = 9$ and $T = 120$ takes less than a second on our computationally weak base station.

5 CASE STUDY: LIVE DEPLOYMENT

To perform a fair comparison of our OPTICS system against a baseline, we installed two identical irrigation systems that allow us to run two control strategies side-by-side. As shown in Figure 12, the two systems were oriented next to each other with a gap purposefully left in-between to prevent cross-contamination between the two systems. An alternate technique to ensure fair comparison between two control strategies would be to periodically alternate the strategies between the two irrigation systems to ensure results aren't skewed by environmental variations. However, this would require us to wait for soil moisture to settle between each alternation, causing all experimentation to take significantly longer. Instead, we chose this side-by-side arrangement, so that each system would have as homogeneous solar exposure, slope characteristic, and soil characteristics as possible.

5.1 Environmental Description

The two irrigation systems were installed side-by-side, and were designed to be identical in hardware, sprinkler coverage, and so on. Each irrigation system measured 60×60 feet, with sprinklers arranged in a 3×3 grid, each 30 feet from the next. The sprinklers chosen were MP Rotators by Hunter Industries [8], which are currently considered state-of-the-art in sprinkler technology. In these devices, the water pressure is focused through many rotating nozzles on the sprinkler head, which allow much greater range at lower water flow rates, applying water more efficiently than their rotor counterparts. The MP Rotators can be adjusted to a range of 15–30 feet, making them ideally suited for our system.

The deployment area was located on a sloped area that drops 3 feet from the highest to the lowest point. During installation of the sprinkler system, we noted that the topsoil was a “Clay

Fig. 13. Sensing and actuation node.

Loam” type, approximately 10 inches deep. Beneath this layer sat a thick clay layer that went beyond 3 feet deep. The grass growing in the installation area is a natural field grass and not the type found on a sports field or in university landscaping, but the goal of irrigation is identical, providing a satisfactory amount of moisture to sustain healthy turf.

5.2 Hardware Description

The hardware used in our experiments stemmed from the design introduced in Reference [40], with battery life and system safety as primary concerns. Control was provided by a latching solenoid, which requires a 50-ms pulse of power in either the positive or negative direction to open or close the valve. The sensor chosen was the Decagon EC-5 [2], commonly used in research for its high accuracy of $\pm 3\%$ and power consumption of just 10 mA for 10 ms to take each sample. Although the EC-5 outputs a raw voltage, Decagon provides a linear function that maps this voltage to Volumetric Water Content (VWC) for our use. Each sensor is inserted into the soil at the expected depth of the root zone for the turf on site, within the first six inches of soil in the case of our experimental system. A standalone board was developed to sit on the General Purpose Input/Output (GPIO) pins of the Tmote Sky, whose purpose is to route power from the attached 4×AA battery source to the peripheral devices. With these peripherals installed, our devices could communicate with each other, monitor the local soil moisture conditions, and control the flow of water to the attached sprinkler. These primary hardware components in their waterproof case can be seen in Figure 13.

Collocated with the irrigation system was a base-station module, consisting of a Raspberry pi with a Tmote Sky attached via USB and a wifi hotspot to allow the system to be accessible remotely. The data processing pipeline as described in Section 2 and the optimization as described in Section 4 was all run on this device. Schedules computed by the *Schedule Optimization Module* were forwarded over USB to the Tmote Sky for wireless distribution, and incoming data was pushed by the Raspberry Pi to an off-site database for remote monitoring and analysis. Later explained in Section 6, the only equipment failure that occurred in our deployment was a failed USB connection between the Tmote and the Raspberry Pi, likely due to environmental factors. However, firmware running on the sensing/actuation devices is designed to handle such a failure by automatically returning to the default “Off” state, preventing significant water loss.

5.3 Baseline Strategies

To allow the two side-by-side irrigation systems to operate independently, all sprinklers are installed with a sensing/actuation node. In this way, the only difference between the two systems are the schedules sent to the sensing/actuation nodes. In this work, we compared the OPTICS system to two baseline systems, an Evapotranspiration control strategy, the current industry leader in system efficiency, and MAGIC [40], the current state-of-the-art in academia. As the University irrigation systems operate on a daily schedule, all baseline systems and OPTICS was configured to irrigate daily as well. The OPTICS system would operate identically on a different irrigation cycle with no reconfiguration necessary.

5.3.1 Evapotranspiration (ET). Many weather stations are available to the public that provide ET estimates based on measurements of the other weather factors. To mimic an ET controller, we query a local weather station for the previous day's ET losses, provided in units of surface water height. With this data, we use our sprinkler's surface application rate to compute how many minutes the system must activate to replace the previous day's losses. In a commercial ET controller, this amount is then the amount irrigated, *plus* a safety margin of water. However, despite contacting two of the largest providers of ET controllers, Hunter [7] and Rain Bird [13], we were unable to find the safety margin used in practice, so we assumed NO safety margin. This means two things—it means a commercial system using a safety margin may provide better quality of service than what we see in our deployment, but at the cost of increased water consumption. Please note that the OPTICS system is able to achieve both goals, water savings *and* improved quality of service.

5.3.2 MAGIC. The MAGIC control framework requires that the installer pre-defines key irrigation and field characteristics before use. The irrigation system characteristics including coverage of sprinklers, application rates, angles, and positions are defined as described in Section 5.1 to match the physical deployment. Likewise, the topography was modeled to reflect the 3' elevation drop of the field, and the estimated soil type was chosen as observed to be a "Clay Loam" of depth 10 inches, sitting atop a deep clay layer. To allow fair comparison to the OPTICS system, the MAGIC optimization was defined with a 2-hour irrigation evening window, to match the campus' irrigation scheduling policy to avoid over-use of the system pressure.

5.4 End-to-end Performance Metrics

As previously discussed in Sections 2, 3.2, and 4, we have chosen a minimum acceptable moisture threshold for all systems that will ensure the system is maintaining satisfactory water levels to keep the plant healthy. As maintaining plant health is the primary goal of an irrigation system, our primary metric for irrigation quality is the system's ability to maintain soil moisture levels above the chosen threshold at all times across the space. By doing so, we are guaranteeing that the plant has sufficient moisture to be healthy. In this article, we call this the quality of service provided by the irrigation system.

Although we must maintain moisture above a minimum threshold, it is also detrimental to over-water the space. In addition to the environmental and financial impact, an over-abundance of water in the soil can, over time, lead to the rotting of the plant roots, discoloration of the plant (aesthetic penalty), and in extreme cases excess irrigation has been linked to the leaching of fertilizer chemicals into human drinking water supplies. As each sprinkler uses a pressure-regulated water supply and we directly control the times at which each sprinkler is active, we can monitor the amount of water consumed by both systems at all times to determine the efficiency of each system. Thus, another metric that is relevant is the water consumption, which we would like to minimize subject to the quality of service constraints.

Fig. 14. Collected VWC data across deployment for all Evapotranspiration (top) and OPTICS (bottom) nodes.

Finally, an aesthetic side-effect of uneven moisture distribution is the appearance of “hotspots” where not enough water is received and oversaturated regions where standing water remains on the surface. These can be identified by a difference in color, and detract from the appearance of the space. Although they can take a long time to develop, with our sensing/actuation platform, we can investigate the long-term moisture trends under our comparative control strategies; a more even moisture distribution prevents these localized effects from happening, and uniformity can be monitored through the deployment’s sensor data.

6 EXPERIMENTAL RESULTS

In this section, we discuss the experimental results of OPTICS when compared to both the ET and the MAGIC systems side-by-side. These tests are performed under the same conditions as explained in Section 5.1, analyzing the quality of service, water consumption, and moisture uniformity metrics discussed in Section 5.4.

6.1 Quality of Service

Irrigation systems are installed to maintain health in the planted turf. However, these systems often fall short of their quality goals. As such, a potential replacement system must either maintain or improve the quality of service. Figure 14 shows the raw moisture data for each sprinkler in the field for both ET and OPTICS. The black horizontal line in the center shows the permanent wilting point (pwp), representing the minimum moisture level we wish to always maintain. We can see that the ET system spends more time and in some cases it is well below that minimum threshold

Fig. 15. ET vs. OPTICS quality of service results (lower is better).

Fig. 16. MAGIC vs. OPTICS quality of service results (lower is better).

for many of the moisture sensors. To quantify how much below the minimum moisture threshold (pwp) each system spends over time, we plot the sum of the squared amount below the minimum moisture level over time in Figures 15 and 16 for the experiments comparing OPTICS with ET and MAGIC, respectively. We used the squared amount to emphasize that the larger the amount below the pwp, the worst the quality of service provided.

In our deployment of OPTICS against the ET control strategy in Figure 14, we see that at least two nodes in the ET system spend significant time below the minimum moisture threshold we wish to maintain, causing the ET system to provide reduced quality of service across the entire deployment as shown in Figure 15. In addition, looking closely, we can see that there are several nodes in the ET system that are significantly *above* our minimum moisture threshold we wish to maintain as well. This emphasizes the limitations of ET and the core of our work. The irrigated regions do not receive moisture the same way, and without learning these eccentricities, a system will often provide unsatisfactory quality of service and/or consume more water than necessary, as in our ET baseline system. This is a common problem in irrigation systems, as uniform irrigation across the field, without understanding local variations will result in moistures that can vary wildly. To make this uniformly irrigating system provide perfect quality of service, we would have

Fig. 17. Consumption of ET vs. OPTICS.

to irrigate *all* sprinklers enough to raise the driest area above our minimum moisture threshold. This would be a significant waste of water, as the rest of the space would be severely over-watered.

It should be noted that before the OPTICS system was turned on, the ET control strategy was used to irrigate both irrigation systems. This can be seen in day 1 in Figure 14, where several sensors have moisture levels below our threshold. However, in the first 4 days, the OPTICS system learns these increased needs and applies tailored moisture to raise them above the threshold, resulting in a steady improvement in quality of service as shown in Figure 15. In the comparison against MAGIC in Figure 16, we can see that several days are spent with a poorer quality of service in comparison to OPTICS. On days 1 and 4 in particular, it is clear that MAGIC's model believes it can afford to reduce water consumption, causing decreased quality of service on the following days. Likewise, although the MAGIC system attempts to send increased water on day 2, the mismatch between the model and the physical deployment does not send enough to reach this goal.

Overall, OPTICS improved irrigation quality of service by an average factor of 4.04 in comparison to the ET system, and if we ignore the first day, by a factor of 24.7. When compared to MAGIC, OPTICS improved irrigation quality of service by an average factor of 2.47. In all, our system provides significant improvements with respect to quality of service than the other irrigation systems.

6.2 Water Consumption

When a new irrigation control system is considered, a primary concern is the efficiency of the proposed system. The system's ability to return its investment based on increased efficiency will often dictate the acceptance of the technology. In addition, the environmental benefits of reduced freshwater consumption are clear and help promote system adoption.

In our experimental setup, the water source for each sprinkler is pressure-regulated to the industry standard, 40 psi. A pressure-regulated sprinkler head distributing water at a known angle uses a clearly defined amount of water per unit time, as described in the sprinkler documentation. By tracking exactly when each sprinkler is actuated by the system, we can determine very accurately how much water has been consumed. In this way, we compute the daily system consumptions for the ET vs. OPTICS and MAGIC vs. OPTICS, as shown in Figures 17 and 18, respectively.

As discussed in Section 6.1, the experimental system started with moisture levels significantly beneath our desired threshold. For this reason, the first 5 days of OPTICS control in our first deployment had steadily improving quality of service as it learned its models and raised moisture

Fig. 18. Consumption of MAGIC vs. OPTICS.

to satisfactory levels, but on days 2 and 4 this resulted in slightly higher water consumption than the ET system. Day 8 saw slightly higher consumption than the ET system as well, but these were some of the ET system's worst days in terms of quality of service as shown in Figure 15.

Day 11 of our first deployment saw increased water consumption as well, caused by a hardware malfunction. A command telling four of the nine OPTICS nodes to "Stop irrigation" was lost due to a faulty USB connection to the base-station mote, causing unintentional irrigation that was not corrected until failsafes in the node's firmware automatically disabled irrigation. This caused the OPTICS system to consumed more water than intended, as shown on day 11 in Figure 14. We can also see OPTICS's ability to recover from such mistakes on days 5 and 12, where significantly less water is required due to the residual moisture from the day before.

The water consumption of OPTICS when compared to MAGIC as seen in Figure 18 was much closer, with some days using slightly less and others slightly more water. However, on days 1 and 4, it is particularly evident that although MAGIC saves significant water, the quality of service suffers immensely.

Across the two deployments, the OPTICS system reduced water consumption by an average of 11.99% compared to the ET system, and 3.28% compared to the MAGIC system.

6.3 Moisture Uniformity

Moisture uniformity is not considered a primary goal, but is a nice side-effect of good irrigation control. If an irrigation system provides water coverage that is not uniform, then drier turf can turn brown and crispy, while wetter turf can turn a different shade of green and become muddy, both effecting aesthetics and usability of the space. With our learning model, a core assumption is that as losses occur at different rates at different spatial locations, we are required to apply different amounts of water across the space. However, if we apply the appropriate amount of moisture across the space, the water in all sensing locations will settle toward a uniform distribution as the moisture levels approach the uniform minimum moisture *just before* irrigation.

Throughout our deployments, we periodically recorded the soil moisture at all sensor locations in both side-by-side irrigation systems. To determine the moisture uniformity, we step temporally through these data traces and compute at each timestep the variance in soil moisture readings for both irrigation systems. This moving variance of soil moisture across the irrigation system can be seen for deployment 1 in Figure 19 and for deployment 2 in Figure 20, where lower values indicate a more uniform soil moisture level across the deployment. Each day during watering, the variance for all systems is seen to spike, as different areas will not receive water at the same rates. Then, as

Fig. 19. Spatial moisture variance over time, deployment 1 (lower is better).

Fig. 20. Spatial moisture variance over time, deployment 2 (lower is better).

the soil moisture diffuses across the space, the variance will decrease to a daily minimum before the next irrigation occurs. It is clear in both deployments that the OPTICS system is able to maintain a much lower moisture variance in general, indicating OPTICS's ability to learn the heterogeneous water needs across the space and apply water in response. In the first deployment, the variance is seen to be higher on days 1 and 11, but these are due to OPTICS's poor initial soil moisture conditions and the hardware failure that resulted in unintentional excess water, as described in detail in Section 6.2. Additionally, the two figures demonstrate that the MAGIC system is able to maintain a tighter band of irrigation than the ET controller, providing comparably low variance at times later in the second deployment.

Finally, we consider each individual sensing location and compute its average volumetric water content (VWC) across the weeks of deployment and visualize them as a heatmap in Figure 21 for each of the three compared systems. For ease of visualization, a bilinear interpolation is used to produce the figure (a more complex interpolation such as cubic may create artifacts *between* sampling points). We see that the ET control strategy results in the least uniform coverage, with a heavily over-irrigated area around the center of the field, and an under-irrigated area close to the lower right corner of the field. In contrast, both MAGIC and OPTICS strategies result in a more uniform moisture distribution, with MAGIC being slightly above the minimum in a cross-like pattern, and OPTICS being slightly above the minimum in the upper right and lower left corners.

Fig. 21. Average VWC (%) of compared systems across deployments, where $\theta_{pwp} = 20\%$.

Fig. 22. Energy profile of the sensor node.

6.4 Energy Consumption

In our devices, the three peripherals that consume significant energy are the sensor, solenoid, and the radio, as visualized in Figure 22. Each sensor sample consumes 10 mA of power for 10 ms, and each flip of the latching solenoid requires 400–450 mA of power for 50 ms. In our system, to ensure we do not cut power too early, we add a safety band of 50% on the timing on both of these devices, powering each for 15 and 75 ms for the sensor and solenoid, respectively. The tmote sky [15] radio consumes 23 mA max when in transmitting mode. With clever use of these peripherals, this system can achieve a substantial system lifetime using our current power source of 4×AAs [3].

In our 25-day deployment, we found that on average OPTICS flips the solenoid of a node 12 times per day. Assuming we sample our sensor 1 time per minute, flip the solenoid 12 times per day, and utilize a constant 1% radio duty cycle ratio using techniques such as Low Power Listening [34], our system lifetime is estimated to be 1.7 years, with 95% of this energy consumed by the radio. However, as fresh data is only required less than 3 hours per day during irrigation, by simply leaving the radio off the other 21 hours each day while storing sampled data in memory to be sent at the beginning of the irrigation cycle, this lifetime is easily extended to ~11.9 years. We note that the MAGIC system achieved similar system lifetime, but required a 4 D-cell power supply to do so, whereas the OPTICS system uses only a 4×AA-cell power supply. This reduced system consumption is due to more efficient voltage regulation hardware on our power supply board, and minor improvements in firmware solenoid control routines, preventing the solenoid from receiving power to enable/disable the valve when it is already in the desired state.

Fig. 23. ET prediction of KNN regressor using historical temperature and humidity trends as features. Days that were particularly challenging to predict are emphasized with an arrow and an oval.

Due to the relatively small size of our irrigation system, radio communications between sensor nodes and base station only required one or two hops using a simple routing strategy. In a larger deployment, we would need to either provide more base stations to keep the number of hops from each node to a base station small, or employ a routing protocol such as ORPL [24], CTP [25], or ORW [26] designed for low-power communication between sensor nodes and the base station at large scale. Routing across a very large irrigation system will surely lead to an increase in energy consumption across the network. Fortunately, our application’s low data rate and lax timing requirements will allow the system to de-activate the radio for the majority of the day, and duty-cycle the radio aggressively during irrigation when data communication between sensing nodes and base stations is required. In Section 9, we discuss techniques that will nearly completely eliminate data communication requirements between the sensing nodes and the base stations.

7 PERFORMANCE ANALYSIS AND EVALUATION

7.1 Weather Prediction Evaluation

Our weather prediction method uses hourly readings of at least one of the 4 ET indicators, temperature, humidity, wind, and solar irradiance, to predict the resulting ET losses due to these weather conditions. When OPTICS is used to control a live irrigation system, it uses forecasted hourly readings of these ET indicators as input to our KNN regressor model, which then predicts the next day’s ET losses. The use of these forecasted ET indicators incorporates some error into our final prediction that is outside our control, as these forecasts are provided by local weather stations or government sources that are imperfect. Before determining the end-to-end predictive accuracy of our solution, we first wish to remove the forecasting error that is out of our control to determine the accuracy of our KNN predictor on its own. To do so, we refer to historical data trends of the hourly ET indicators and the associated daily ET losses *of the same day* to determine the accuracy of this mapping between indicators and resulting ET. By doing so, we remove the error associated with the error forecasting the hourly trends, and focus on best-case predictive power.

Figure 23 shows the daily ET as predicted using the *same day’s* hourly vectors of temperature and humidity, as well as the ground-truth ET that occurred and a moving average of each. We can see that in general, the predicted ET follows the trend of ground-truth ET, but on one day in mid-July (emphasized with an arrow) our system predicts a “normal” ET level when in fact an abnormally low ET value occurs. Additionally, on several days in early December (emphasized

Fig. 24. ET prediction of KNN regressor with all four features available. Days that were particularly challenging to predict are emphasized with an arrow and an oval.

Table 3. Predictive Error Across 120 Days of Collection

Data Used	RMSE	NRMSE	Cumulative predicted ET	Relative error
All four indicators	0.0171"	0.0813	30.05"	+0.9%
Temp/humidity	0.0325"	0.1547	30.64"	+2.9%
Temp/humidity (Forecasted)	0.0348"	0.1655	29.29"	-1.6%

with an oval), we can see that our system predicts more abnormal ET when in reality the weather trends were closer to the previously seen weather trends. In contrast, Figure 24 demonstrates the ET prediction *IF* all four ET indicators were available. We can see that our predicted ET levels are much closer to the ground truth on the days in mid July and early December where regression using only temperature and humidity did not provide enough information in Figure 23. This indicates that days in Figure 23 with particularly high error are caused by heavy influence due to solar and wind factors on these days. In practice, the fit quality using temperature and humidity as features is more indicative of the real-world accuracy of this prediction technique as these features are the most likely to be available using forecasting. Table 3 shows the root mean square error (RMSE) of prediction with these feature subsets, as well as the RMSE normalized by the magnitude of the individual ground-truth values (NRMSE). We can see that availability of all four features results in half the error as prediction using just temperature and humidity.

Next, to better understand the practical accuracy, we re-introduce the forecasting of hourly trends by using the *forecasted* temperature and humidity from the day before as input to our KNN regressor using 120 days of collected forecasting data around our period of deployment. Figure 25 shows the daily ground-truth ET, along with the predicted ET using forecasted hourly trends, along with the associated error of each prediction. Here, we can see clearly that our challenging day in mid July (emphasized with an arrow) had the most error as it also did *without* forecasting in Figure 23, as neither of these predictions had wind and solar exposure information available as features. Whereas prediction using these ground-truth features resulted in a root mean square error of 0.0325, we find that using forecasted features increases this error metric to 0.0348. Put another way, forecasting our hourly data trends increases our daily prediction error by just 7% in comparison to performing regression using ground-truth trends.

The cumulative distribution function (CDF) of the individual daily errors in forecasted ET prediction across our 120 days of data collection can be seen in Figure 26. Here, we see that about 50%

Fig. 25. ET prediction of KNN regressor using *forecasted* temperature and humidity features. Days that were particularly challenging to predict are emphasized with an arrow.

Fig. 26. CDF of weather prediction errors.

of our predictions are within 10% of the ground-truth ET, and 90% of predictions are within 25%. To get a sense of the overall performance, we consider that across these 120 days, ground-truth data shows that a total of 29.78 inches of ET losses occurred. Table 3 shows that the cumulative losses of our predictive methods with and without prediction using various features all fall within 3% of error across this period, which we believe is more than adequate for our irrigation controller.

7.2 Short-term Model Analysis

One of the high-level system parameters that affects our modeling is the choice of Δt , determining how far in the future our model must predict and the temporal granularity of our control routines. The correct way to determine the *optimal* choice for this parameter would be to repeatedly vary Δt , launch a full irrigation system deployment with modeling and control using the new choice of Δt and analyze the primary metrics of success (water consumption and quality of service). However, due to the substantial effort required to repeatedly perform full experimentation makes this process impractical. Instead, we consider the effect of this parameter choice just on the accuracy of our short-term model's predictive accuracy.

Across our irrigation system deployment, we store detailed records of the schedules used and the sensor trends at all node locations over time. To determine how model accuracy responds to changes in model Δt , we take these deployment data traces and step through temporally at

Fig. 27. Box and whisker plot of errors as model Δt is varied. Crosses denote outlier values.

Fig. 28. Mean error as model Δt is varied.

changing values of Δt , building training pairs of $[s_t, f_t]$ vectors as inputs, and $[s_{t+\Delta t}]$ vectors as outputs, as described in detail in Section 3.3. We then train our linear regressor on a random 90/10 train/test split of the dataset and note the model prediction accuracy on the test set. As the random training/testing data split will change the prediction accuracy, this is repeated many times for each value of Δt , with the error distribution box plot as shown in Figure 27, and the mean error shown separately in Figure 28. The error shown in these figures is the root mean squared error across the full dataset, averaged across all datapoints and across the K sprinklers. The change in moisture that occurs in Δt units of time is very small relative to the full range of moisture values that can occur, so we then normalize this absolute error to the mean moisture change that occurred in Δt units of time. In this way, we show the relative error between the ground-truth moisture change and the predicted value for each individual sensor prediction Δt seconds in the future.

In these plots, we can see that when Δt is very small, the average prediction error quickly increases as noise becomes a dominant factor in prediction. In our experiments, our finest temporal granularity in soil moisture data is 30 seconds, and using $\Delta t = 30$ s to build our linear model, we see a maximum outlier error on the order of 25% in Figure 27. In addition, we see in Figure 28 that at $\Delta t > 800$ s, our error begins to increase again, with the lowest error experienced around $600\text{s} < \Delta t < 800$ s at around 4.5% mean relative error. It is important to note that this range of optimal Δt is specific to the environmental conditions that exist in *our* irrigation system, and will be different if the system is installed elsewhere, due to differences in sprinkler type, soil parameters,

Fig. 29. Daily error of MAGIC [40], Learning (Re-learns each day w/new data), and Learned (Trained once with all data).

topography, and so on. At the $\Delta t = 60$ s we chose to use in our experimentation, we can see that the average error can be expected to be around 50% higher than at the optimal Δt value. This sub-optimal parameter was selected for our experiments as it would allow us finer granularity of control, and as we frequently re-correct our moisture conditions during optimization using the most recently collected moisture data, we ensure that our model does not cause significant drift in the system.

In addition, to evaluate the OPTICS model, we consider the single-step predictive error as it changed across our 25-day dataset from our live deployment in Figure 29. Here, we consider the error of three different models: the first is the PDE model used in the MAGIC system, as it was configured for our experimentation in our deployment. The second is the OPTICS system, employing an *on-line model improvement*; this model is first trained using our training day of data as described in Section 3.3, and then retrained on each day D using the previous $D - 1$ days of data. This technique simulates a model that has been freshly installed on a new irrigation system, and must learn its model adaptively day-by-day. The third model is a pre-trained OPTICS model, where all data from our pre-deployment *AND* our deployment were used to train a model beforehand that then does not change. This technique simulates a model that has been trained over time on our irrigation system, and no longer requires updating. As in the previous sections, the error shown is the average single-step predictive error of a single sensor node, normalized to the average change in moisture experienced in Δt seconds, with $\Delta t = 60$ s as it was in our deployment.

As we can see in Figure 29, the OPTICS model that is constantly being retrained has more variation in errors, with consistently lower error on days 4–14 and significantly higher error on days 3 and 16 where error spikes occur. These periods of high error are caused when the model experiences moisture conditions or valve scheduling that it has not experienced before in the data used for training. We can see that these increases are temporary, and that once these conditions are included into the model, the error decreases again quickly in the following days. In all, we see that some days this model can have as little as 3% error, with temporary spikes as high as 16.5%. In comparison, the MAGIC PDE model tends to be more stable in its predictive error as the moisture predictions are not data-driven, but made based on deterministic models that can still operate reasonably when faced with unseen conditions. The accuracy of this model across our deployments tends to lie between a minimum error of 7% on day 16 to a maximum of 12.5% on day 21. Finally, we can see that the OPTICS model that is pre-trained with all data is significantly more stable in its predictions, with the exception of unusually high error in the very first day. This technique shows a minimum of about 4% error and a maximum peak of 12.5% error on the first day but

Fig. 30. Percent of actuations requiring rounding across the two deployments.

Fig. 31. Daily error in irrigation schedules due to rounding of LP solution.

provides an average predictive error of just 6.5% across the deployments. In general, we find that the OPTICS model with on-line updating reduced predictive error by 26.0% compared to the MAGIC PDE model, and the pre-trained OPTICS model reduced predictive error by 30.4% compared to the MAGIC PDE model across our deployment dataset. The improved predictive accuracy of OPTICS stems from its ability to learn the heterogeneous factors of water movement that are assumed to be homogeneous in MAGIC's PDE model. The slight average accuracy improvement of the pre-trained model is expected, as the more extensive dataset it had been trained with included the atypical conditions that were difficult for the model that was doing on-line updates.

7.3 Effect of Optimization Simplification

As discussed in Section 4, we simplify an integer linear program (ILP) by relaxing the integer valve constraints to be continuous on $[0, 1]$, and then rounding the resulting values to be binary. Although this makes the optimization a linear program (LP) that is easier to solve, it can potentially cause the resulting schedule to deviate from the one chosen by the optimizer. To determine this effect, we consider the individual valve actuations produced by the optimizer across our 4 weeks of deployment. As shown in Figure 30, we find that the resulting schedules tend to lie on the 0/1 integer boundary, with fewer than 1% of actuations requiring rounding. Figure 31 shows how the amount of irrigation deviates each day due to this rounding, and we can see that the worst day

Fig. 32. Time required to optimize schedules.

(day 17) has a resulting schedule deviation of just over 1 second, negligible when the total irrigation is on the order of 1 hour.

7.4 Performance and Scalability of OPTICS

In the deployment of the OPTICS system, we chose to re-run the optimization using the most up-to-date sensor readings every control timestep, chosen to be 1 minute. This has the benefit of preventing drift due to errors in the short-term model by constantly re-correcting, but imposes an upper bound on optimization time. To determine the effect this limitation has on the scalability of the system, we train a short-term model using spatially repeating trends of deployment data to simulate an irrigation system that has an arbitrarily large number of sprinklers (K), and track the time required to optimize schedules. This can be seen broken into two discrete time intervals in Figure 32, where set-up time is the time required of our chosen optimization tool to set up the constraints, and solve-time is the time to solve the LP. Also shown as a dotted line is the 60 s maximum available time to solve optimization. We can see that as the number of sprinklers in the system increases, the amount of time required to optimize schedules increases as well, until solve-time exceeds our 60 s bound around $K = 225$ sprinklers.

Appendix A derives how to compute the spatial coverage area of an irrigation system with K sprinklers of radius R arranged in a square grid in the three possible irrigation system architecture arrangements. As shown in Figure 32, enforcing our 1 minute maximum solve-time, the OPTICS system can provide optimal schedules for an irrigation system of $K \approx 225$ sprinklers. Estimating using the third case of Equation (6), which applies to our coverage conditions here, we find that for sprinklers such as those used in our deployment with radius $R = 30$ ft and assuming a conservative, dense sprinkler deployment where each sprinkler can reach all the way to the next adjacent sprinkler in the grid, the OPTICS system can cover ≈ 5.2 acres of land using a single short-term model. With sprinklers used more commonly in sports fields with radius $R = 60$ ft, this coverage increases to ≈ 20.7 acres of land with a single model. Although the largest irrigated sports fields, 18-hole golf courses, cover an average of 100 acres [6], these irrigated spaces tend to be naturally partitioned with golf cart paths that would prevent these areas from being continuous anyways. Furthermore, due to the very slow speed of moisture movement through the soil, in practice moisture has a relatively limited region of movement. Both of these characteristics of the space allow us to break such a large irrigated space into smaller, more manageable models of water movement. By doing so, we facilitate superior performance and scalability, as further discussed in Section 9.

Table 4. Sprinkler Node
Manufacture Cost

Component	Price
Mote	\$37.57
Moisture Sensor	\$110
Batteries	\$4
Solenoid	\$15
Waterproof Enclosure	\$10
Manufacture & Assembly	\$10
	\$186.57

In practice, although the OPTICS system is *able* to optimize schedules for an irrigation system with hundreds of sprinklers, the *training* of a short-term model at this scale becomes challenging. In our deployment, we simply stepped through each sprinkler for a fixed period of irrigation of 1 hour, but scaling this technique to a system with K sprinklers would take K hours. As discussed in Section 9, however, in practice each sprinkler's area of effect is limited by the sprinkler radius and water movement speed across the surface and through the sub-surface. In the most densely configured lawn irrigation systems, the actuation of one sprinkler may affect the soil moisture at the next adjacent node in the grid, but not the next. Due to this limitation of movement, we can break an irrigation system's sprinkler grid into blocks of 3×3 sprinklers. Then during training, we can actuate, for instance, the sprinkler in the (1, 1) position of each 3×3 block simultaneously across the entire space, and any sensing node will have at most 1 active sprinkler within range. At the end of the hour, we can activate the (1, 2) position in each 3×3 block, and so on, until we have activated all nine sprinklers in each block. In this way, we can irrigate an arbitrarily large grid of sprinklers in at most 9 hours without cross-contamination between active sprinklers.

8 RETURN ON INVESTMENT ANALYSIS

There are social and political motivations behind a project like this, but a primary consideration before the purchase and installation of a replacement control system is the return on investment, or the time it takes a system to save enough money to cover the cost of installation and usage. To calculate the return on investment, we must take into account the initial cost of the replacement system and the monetary savings expected from the increased efficiency of the replacement system. In Table 4, we list the cost of production for one of our irrigation control devices. Other than screwing these devices under each sprinkler head, the original infrastructure does not need to be modified in any way.

Financial savings stem from the system's ability to save water. In Figure 33, we show how OPTICS will return its own investment based on water savings alone. As the return on investment will differ depending on the sprinkler heads used, we consider OPTICS's installation on a representative University sprinkler head with 11.99% water efficiency improvement compared to industry best, and the fact that our University pays \$5.60 per thousand gallons for irrigation. Additionally shown in Figure 33 is a $\pm 10\%$ band in water pricing, to take variation of water pricing into account. In this way, each unit device is estimated to return its investment in 14–17 months. OPTICS's improved efficiency and reduced need for a powerful computation machine makes it even more financially attractive than MAGIC, especially due to MAGIC's additional setup costs that we are not including, as discussed in Section 10.

Fig. 33. Return on investment curve.

9 LIMITATIONS AND FUTURE WORK

As shown in Section 8, the unit cost of our control devices is dominated by our soil moisture sensor, which was chosen in our experiments due to its very high accuracy ($\pm 3\%$ [2]). 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. For instance, previous work [35] has found that the Watermark line of soil moisture sensors can be calibrated to have an accuracy of $\pm 5\%$, and costs only \$35. By using these sensors instead, the unit cost of our sensor/actuator devices would be reduced by 40%, with a similar reduction in return on investment and increased adoptability.

As OPTICS is designed for turf irrigation, it is unlikely to provide benefit in shrubbery or tree irrigation, where much simpler drip irrigation systems can be used. In addition, some very different turf species may require varying minimum moisture levels to maintain health. Therefore, in turf irrigation systems where the system covers *multiple* turf species such as a golf course irrigation system, minimal configuration will be required beforehand to tell OPTICS where each grass type is located. For instance, “Sprinklers 1–90 irrigate turf species A, Sprinklers 91–100 irrigate turf species B,” so that differing minimum moisture constraints can be spatially assigned based on the species. Assigning heterogeneous minimum moisture constraints will not require any modification to our processing pipeline.

OPTICS can guarantee optimal moisture levels at the sensing locations, but due to uncertainty on environmental conditions such as objects occluding the sprinklers and other localized features, it is possible that perfect irrigation is not achieved in between sensors. A future step may be the addition of an emerging plant health imaging technology [12, 22] to provide supplemental long-term feedback to ensure these locations are also satisfied at all times.

As discussed in Section 3.2, we chose to fit our Long-term moisture losses as an exponential decay function and re-train our model each day using all previous data. However, the system may be more reactive to changing conditions by only taking the latest N days of data in retraining or another weighted updating strategy to put emphasis on the freshest data, representing the latest environmental conditions. Furthermore, in some irrigation systems where the sprinklers do not provide a uniform coverage onto the ground, this method may be sensitive to delayed effects in moisture movement, and may benefit from another modeling strategy that does not require the losses to occur precisely between the end and the start of irrigation, but rather follows the min/max moisture trends across the daily cycle.

As discussed in Section 6.4, the current iteration of the OPTICS system has a potential system lifetime of around 12 years with our battery power supply. However, we note that even very rare battery changes can be costly and time-consuming in the system at large scale. As each distributed device is collocated with a source of relatively high-pressure running water, we propose that a small water turbine is capable of providing power orders of magnitude higher than required to recharge the distributed devices. With clever use of energy-heavy peripherals, the use of energy harvesting will allow the system to run perpetually, even across long winters without irrigation. We leave this system design improvement to future work.

In the OPTICS system, the Short-term model introduced in Section 3.3 describes water movement across the entire irrigation system due to short-term effects such as sprinkler application, runoff, and so on. In irrigation systems that are small to medium-sized, this is perfectly reasonable, but when you consider an irrigation at very large scales such as those irrigating golf courses or large parks, modeling the effect of each sprinkler across the entire space becomes unnecessary and wasteful. Due to the limited range of each sprinkler, and the relatively slow rate of surface and sub-surface water flow, in practice each sprinkler will only have an area of effect with a radius only a few times larger than the sprinklers'. Due to the quadratic relationship between irrigation system size and the time it takes to find optimal schedules as discussed in Section 7.4, we could greatly improve the scalability of the system without sacrificing model accuracy by sub-dividing the irrigation system into smaller regions, each with its own *local* model, rather than maintaining a single global model. As the data-driven model was chosen to be a lightweight linear regressor, model training and prediction can potentially be implemented on the sensor nodes directly, removing the need for centralized data processing and thus multi-hop data communication between sensing nodes and the base station. As the only cross-network communication required would be the dissemination of the predicted future weather from the base station to the sensing nodes, significant radio energy savings are possible using this technique.

10 RELATED WORK

Whereas all standard irrigation systems employ centralized control, where a single water valve controls the flow of water to many sprinklers, Reference [38] was the first system to propose *distributed* control, where the granularity of control would be further increased with the use of distributed sensing and actuating devices installed on each sprinkler. The benefits of such a system are huge, as it allows water usage to be tailored to the diverse needs of the irrigated space, which are heterogeneous in nature and thus improperly irrigated by centrally controlled irrigation systems. A distributed control system such as this had not been used before due to the additional challenge required to generate good irrigation schedules for an increased number of actuators in an efficient manner. To overcome this challenge, a cellular automata model was created to consider the movement of moisture due to sprinkler application, surface, and sub-surface flow. By then minimizing water consumption subject to quality of service constraints, this model was used to find sprinkler schedules to be run by the irrigation system. Although experiments were performed solely in simulation and model complexity limited the system's use at scale, the study found that significant efficiency and quality of service benefits were possible as higher granularity of control was provided, paving the way for future research in this field.

Work in Reference [40] then introduced the distributed irrigation sensor/actuator, allowing a greenskeeper to wirelessly send irrigation schedules to each individual sprinkler head, breaking the traditional limitation of a single valve per irrigation system and allowing more customized irrigation schedules to be computed and run. In conjunction, a control strategy was proposed that builds a mechanistic PDE model of moisture movement within irrigated space. Using this

model, optimal schedules with respect to water consumption were computed, which would maintain proper moisture levels until the end of irrigation. Although this was found to reduce water consumption and improve irrigation quality, the system had practical limitations. No process was proposed to correct the model over time, and future weather prediction was not taken into account, which can result in increased water consumption. Furthermore, the size and complexity of the model and optimization problem resulted in it requiring simplification via linearization and spatio-temporal discretization to make it tractable. Even with these simplifications that sacrifice model accuracy, significant processing was still required and could only guarantee satisfactory moisture levels for the immediate hours after irrigation, not the full 24-hour cycle.

The models used in Reference [40] were configured manually and statically using approximated parameters at installation-time under the assumption that these model parameters were either spatially homogeneous or that they would not change over time. However, many of these parameters such as soil depth can significantly vary spatially and others such as solar irradiance certainly change over time (seasonally). These assumptions were made in Reference [40] due to the difficulty or infeasibility of accurately measuring these parameters at scale. As an example, to measure unsaturated soil conductivity, there are several techniques as listed by the USDA [10]. One of them, relying on core samples, is low-cost but *each* measurement requires *several days* depending on the soil type [31]. Another technique, instantaneous profile, requires equipment costing around \$3000 USD and *each* measurement can take *up to a week* to complete. To make tens or hundreds of such measurements across a large irrigated space to generate accurate models for a static framework simply does not scale, especially when some parameters change seasonally. In contrast, as water retention and movement through the soil is influenced by all of these factors, the learning model used by OPTICS can learn and adapt to these environmental characteristics with *data-driven* model generation using direct soil moisture measurements without having to make crude assumptions to fit a mechanistic model.

With the introduction of more accurate and efficient soil moisture sensors, work has been done to create irrigation controllers that react directly to moisture levels in the soil [16, 29]. However, without a model of the way water is lost, these systems must either over-irrigate to artificially create a buffer, or reactively trigger irrigation during the day, which can lead to plant sunburn and an increase of wasted water to evaporation. In our work, we use sensor feedback *and* predictive modeling together to control in a way that is more water efficient and improves quality of control without having to make assumptions about water needs.

As weather is a primary water source or sink in an irrigated space, systems have been developed to use weather as input for control. The simplest of these systems use standard fixed-schedule irrigation, but allow a precipitation sensor to override control to save water during rain [9]. The more complicated systems, now industry standard, use evapotranspiration, an estimate of the amount of water lost to evaporation and plant transpiration to do efficient water-loss replacement [18, 28]. Some providers boast an average 30% reduction in water consumption, but as with all industry irrigation systems, ET-based systems are installed on traditional single-valve irrigation systems limited by centralized control and can not provide site-specific irrigation, reducing potential system efficiency and quality of control.

To control intelligently, we must take into account *future* temperature shifts, precipitation, and other effects that affect irrigation requirements as discussed in Section 3.1. The standard weather metric for irrigation control has become evapotranspiration (ET), a measure of how much water is lost from the soil due to solar radiation, temperature, humidity, and wind. In Reference [33], a patent describes an irrigation controller that predicts future ET losses, but the main contribution is the combination of a reference ET estimate offset by predicted precipitation, to produce a system

that will not over-irrigate with rain in the near future. Although the author recognizes the potential to use other forecasted weather metrics to predict future ET, they offer no implementation of this feature. An extensive study in Reference [21] finds that exceptional prediction of ET is possible when predictions for all four ET variables (solar radiation, temperature, humidity, and wind) are available in local data sources. The authors recognize that this is often not the case, and this holds true in our work, as wind speed and solar irradiance predictions are unavailable for our locale. As no suitable ET prediction could be found, we design our own.

11 CONCLUSIONS

Turf is the largest crop by surface area in North America, and using fresh water for irrigation puts significant pressure to make this process efficient for such a delicate resource. In this work, we seek to improve the efficiency of turf irrigation systems by designing, implementing and evaluating OPTICS, a data-driven control strategy that automatically adapts to local conditions and weather patterns, requiring virtually no human input in both setup and maintenance. Our system reduces the water consumption by an average of 12.0% against the industry best and 3.3% against state-of-the-art research. Despite this reduced water use, OPTICS was found to reduce turf exposure to unhealthy levels of moisture by a factor of $4.0\times$ and $2.5\times$ with respect to the two systems mentioned above. The OPTICS system is expected to return its investment in 14–17 months based on water savings alone.

APPENDIX

A SPATIAL SYSTEM COVERAGE AS A FUNCTION OF NUMBER OF SPRINKLERS

To evaluate the scalability of our control strategy, it is useful to know how the spatial coverage of the irrigation system will change as we increase the number of sprinklers in the system. Here, we consider this relationship on a theoretical irrigation system in a grid of K sprinklers arranged in a square of $k \times k$ sprinklers. Each sprinkler installed has a radius R and will be installed a distance d to the next adjacent sprinkler in the grid. Reference [37] demonstrates that the area of overlap between two circles of equal radius R with distance d between them as shown in Figure 34 can be written as

$$A_{\text{intersection}} = \begin{cases} 2R^2 \cos^{-1}\left(\frac{d}{2R}\right) - \frac{d}{2} \sqrt{4R^2 - d^2}, & \text{if } d \leq 2R, \\ 0, & \text{if } d > 2R. \end{cases} \quad (5)$$

Fig. 34. Overlapping of circles.

Using this identity, we derive irrigation system coverage in the three possible irrigation system configurations to be

coverage =

$$\begin{cases} K\pi R^2, & \text{if } \frac{R}{d} < \frac{1}{2}, \\ 4\left((2k-1)\frac{\pi R^2}{4} - (k-1)\frac{A_{\text{intersection}}}{2}\right) + (k-1)^2\left(\pi R^2 - 4\frac{A_{\text{intersection}}}{2}\right), & \text{if } \frac{1}{2} \leq \frac{R}{d} \leq \frac{\sqrt{2}}{2}, \\ 4\left((2k-1)\frac{\pi R^2}{4} - (k-1)\frac{A_{\text{intersection}}}{2}\right) + (k-1)^2d^2, & \text{if } \frac{R}{d} > \frac{\sqrt{2}}{2}. \end{cases} \quad (6)$$

Case 1: $\frac{R}{d} < \frac{1}{2}$

In the first possible system architecture, none of the sprinklers in the irrigation system will overlap each other as shown in Figure 35. In this simple case, each of the K sprinklers will have a spatial coverage of πR^2 , with a total irrigation system coverage of $K\pi R^2$.

Fig. 35. Case 1: No sprinkler overlap is present in the system.

Case 2: $\frac{1}{2} \leq \frac{R}{d} \leq \frac{\sqrt{2}}{2}$

In this system architecture, the sprinkler can sufficiently reach the nearest adjacent sprinkler in the grid, but can not reach the diagonal sprinklers in the grid, as shown in Figure 36. For this reason, there are areas that will not receive direct irrigation in the center of the system. To compute the coverage of the system, we break the space into two areas. The first area lies *outside* the grid, as shown by four symmetrical regions surrounded by red dotted lines in Figure 36(a). The second is the area *within* the sprinkler grid, shown in Figure 36(a) surrounded by a solid black line. Together, these two areas constitute the full coverage of the irrigation system.

To compute the coverage of the irrigation system *outside* the grid, we consider one of the 4 outer boundary areas as shown in Figure 36(b). As shown, the outer area is composed of $2k - 1$ quarter-circles of coverage, for a coverage of $\frac{\pi R^2}{4}(2k - 1)$. To remove the darker overlapping coverage appearing in Figure 36(b), we compute the area of each of the $k - 1$ overlapping areas, written as $A_{\text{intersection}}$ using Equation (5), noting that only half of each sprinkler's overlap will fall in this border region with the other half falling into the interior region. In this way, each of the four

Fig. 36. Case 2: Sprinkler overlap is present, but is not sufficient to cover the diagonals grid distance. This results in incomplete coverage within the grid of sprinklers.

border areas will have coverage:

$$(2k-1)\frac{\pi R^2}{4} - (k-1)\frac{A_{\text{intersection}}}{2}. \quad (7)$$

To compute the sprinkler coverage *within* the sprinkler grid, we break the grid into $(k-1)^2$ cells as shown in Figure 36(c). Within each grid cell, each of the four sprinklers will provide a quarter-circle of coverage for a total area of πR^2 . We will then remove the four areas of overlapping coverage that appear darker in the figure written as $A_{\text{intersection}}$, again using Equation (5), noting that half of each overlapping region will fall into the grid cell we are considering. Then, each grid cell will have a coverage of

$$\pi R^2 - 4\frac{A_{\text{intersection}}}{2}. \quad (8)$$

In total, then, the irrigation system as shown in Figure 36(a) with four symmetrical strips of coverage along the external boundary and $(k-1)^2$ internal grid cells will cover an area of

$$4\left((2k-1)\frac{\pi R^2}{4} - (k-1)\frac{A_{\text{intersection}}}{2}\right) + (k-1)^2\left(\pi R^2 - 4\frac{A_{\text{intersection}}}{2}\right). \quad (9)$$

Case 3: $\frac{R}{d} > \frac{\sqrt{2}}{2}$

Finally, in the system architecture that is found in all standard irrigation system, all regions within the grid of sprinklers can receive direct sprinkler coverage, as shown in Figure 37. Similarly to Case 2, above, we break the irrigation system as shown in Figure 37(a) into four symmetrical boundary regions as shown surrounded by red dotted lines and the region within the grid surrounded with a solid black line.

Computing the coverage of each of the four surrounding boundary regions is identical to the process introduced in Case 2, above. Considering the coverage of the $2k-1$ quarter-circles and removing the overlap between the sprinklers, each boundary strip as shown in Figure 37(b) has a total coverage area of

$$(2k-1)\frac{\pi R^2}{4} - (k-1)\frac{A_{\text{intersection}}}{2}. \quad (10)$$

Fig. 37. Case 3: Sprinkler overlap is sufficient to provide direct irrigation to all regions within the irrigation system sprinkler grid.

To compute the sprinkler coverage *within* the grid of sprinklers, we break the system into $(k - 1)^2$ grid cells, one of which is shown in Figure 37(c). As the sprinkler radius R is sufficient in this architecture for complete coverage within each grid cell, the coverage within each cell is simply d^2 , where d represents the distance between the sprinklers as shown in Figure 34. Then, the total coverage of the irrigation system with four symmetrical boundary regions and $(k - 1)^2$ interior grid cells of coverage can be written as

$$4 \left((2k - 1) \frac{\pi R^2}{4} - (k - 1) \frac{A_{\text{intersection}}}{2} \right) + (k - 1)^2 d^2. \quad (11)$$

REFERENCES

- [1] 2019. CIMIS Weather Data. Retrieved from <http://www.cimis.water.ca.gov/>.
- [2] 2020. Decagon EC-5 Device. Retrieved from <https://www.metergroup.com/environment/products/ec-5-soil-moisture-sensor/>.
- [3] 2019. Energizer Ultimate Lithium. Retrieved from <http://data.energizer.com/pdfs/l91.pdf>.
- [4] 2020. Freshwater Crisis. Retrieved from <https://www.nationalgeographic.com/environment/freshwater/freshwater-crisis/>.
- [5] 2019. GNU Linear Programming Kit. Retrieved from <https://www.gnu.org/software/glpk/>.
- [6] 2020. Energy Use and Environmental Practices on U.S. Golf Courses, Phase II, Volume I. Golf Course Superintendents Association of America Environmental Report. Retrieved from https://www.gcsaa.org/docs/default-source/Environment/phase-2-water-use-survey-full-report.pdf?sfvrsn=2b39123e_4.
- [7] 2019. Hunter: Advanced Evapotranspiration Weather Control. Retrieved from <https://www.hunterindustries.com/en-metric/irrigation-product/sensors/et-system>.
- [8] 2019. Hunter MP Rotator. Retrieved from <https://www.hunterindustries.com/irrigation-product/mp-rotator/mp-rotator>.
- [9] 2019. Hunter Rain-Click Rain Detection. Retrieved from <https://www.hunterindustries.com/irrigation-product/sensors/rain-clicktm>.
- [10] 2019. Measuring Hydraulic Conductivity For Use in Soil Survey. Retrieved from https://www.nrcs.usda.gov/Internet/FSE_DOCUMENTS/nrcs142p2_053204.pdf.
- [11] 2019. NOAA Maps and Models. Retrieved from <https://www.weather.gov/akq/mapsmodels>.
- [12] 2020. Photonics—Plant health imaging filters. Retrieved from https://www.photonics.com/Products/Plant_Health_Imaging_Filters/pr60994.
- [13] 2020. Rain Bird Evapotranspiration Manager. Retrieved from https://www.rainbird.com/sites/default/files/media/documents/2018-02/man_ETManager.pdf.
- [14] 2019. Raspberry Pi. Retrieved from <https://www.raspberrypi.org/>.
- [15] 2020. Tmote Sky. Retrieved from <http://www.crew-project.eu/sites/default/files/tmote-sky-datasheet.pdf>.
- [16] 2019. UGMO Irrigation. Retrieved from <http://www.ugmo.com/>.

- [17] 2019. Wunderground—Weather Forecast and Reports. Retrieved from <https://www.wunderground.com/>.
- [18] Richard G. Allen, Luis S. Pereira, Dirk Raes, Martin Smith, et al. 1998. Crop evapotranspiration—Guidelines for computing crop water requirements—FAO irrigation and drainage paper 56. *FAO Rome* 300, 9 (1998), D05109.
- [19] J. Bear. 1972. *Dynamics of Fluids in Porous Media*. Dover. <https://books.google.com/books?id=-XEaxd3hGzoC>.
- [20] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. 2014. Julia: A fresh approach to numerical computing. Retrieved from <http://arxiv.org/abs/1411.1607>.
- [21] Jiabing Cai, Yu Liu, Tingwu Lei, and Luis Santos Pereira. 2007. Estimating reference evapotranspiration with the FAO Penman-Monteith equation using daily weather forecast messages. *Agric. Forest Meteorol.* 145, 1 (2007), 22–35. DOI: <https://doi.org/10.1016/j.agrformet.2007.04.012>
- [22] Laury Chaerle and Dominique Van Der Straeten. 2001. Seeing is believing: Imaging techniques to monitor plant health. *Biochimica Biophysica Acta—Gene Struct. Express.* 1519, 3 (2001), 153–166. DOI: [https://doi.org/10.1016/S0167-4781\(01\)00238-X](https://doi.org/10.1016/S0167-4781(01)00238-X)
- [23] Ernest Carr Childs. 1969. *An Introduction to the Physical Basis of Soil Water Phenomena*. A Wiley Interscience Publication, John Wiley and Sons Ltd., London/New York/Sydney/Toronto.
- [24] Simon Duquennoy, Olaf Landsiedel, and Thiemo Voigt. 2013. Let the tree bloom: Scalable opportunistic routing with ORPL. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys’13)*. ACM, New York, NY. DOI: <https://doi.org/10.1145/2517351.2517369>
- [25] Rodrigo Fonseca, Omprakash Gnawali, Sukun Kim Kyle Jamieson, Philip Levis, and Alec Woo. 2006. *The Collection Tree Protocol (CTP)*. Technical Report. TinyOS Enhancement Proposals, TinyOS 2.0 Network Protocol Working Group. Retrieved from <http://www.tinyos.net/tinyos-2.1.0/doc/html/tep123.html>.
- [26] Euhanna Ghadimi, Olaf Landsiedel, Pablo Soldati, Simon Duquennoy, and Mikael Johansson. 2014. Opportunistic routing in low duty-cycle wireless sensor networks. *ACM Trans. Sen. Netw.* 10, 4 (June 2014). DOI: <https://doi.org/10.1145/2533686>
- [27] Thomas Harter and Jay R. Lund. 2012. *Addressing Nitrate in California’s Drinking Water*. Technical report. University of California, Davis.
- [28] Marvin Eli Jensen, Robert D. Burman, and Rick G. Allen. 1990. Evapotranspiration and irrigation water requirements. Technical report. ASCE.
- [29] Yunseop Kim, R. G. Evans, and W. M. Iversen. 2008. Remote sensing and control of an irrigation system using a distributed wireless sensor network. *IEEE Trans. Instrument. Measure.* 57, 7 (July 2008), 1379–1387. DOI: <https://doi.org/10.1109/TIM.2008.917198>
- [30] M. A. Maupin, J. F. Kenny, S. S. Hutson, J. K. Lovelace, N. L. Barber, and K. S. Linsey. 2014. *Estimated Use of Water in the United States in 2010*. U.S. Geological Survey Circular 1405. DOI: <https://doi.org/10.3133/cir1405>
- [31] A. Nemes, M. G. Schaap, F. J. Leij, and J. H. M. Wosten. 2001. Description of the unsaturated soil hydraulic database UNSODA version 2.0. *J. Hydrol.* 251 (June 2001).
- [32] Tamoghna Ojha, Sudip Misra, and Narendra Singh Raghuwanshi. 2015. Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. *Comput. Electron. Agric.* 118 (2015), 66–84. DOI: <https://doi.org/10.1016/j.compag.2015.08.011>
- [33] M. J. Oliver. 1997. Evapotranspiration forecasting irrigation control system. Retrieved from <https://www.google.com/patents/US5696671>. U.S. Patent No. 5,696,671.
- [34] Joseph Polastre, Jason Hill, and David Culler. 2004. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. ACM, 95–107.
- [35] Egbert J. A. Spaans and John M. Baker. 1992. Calibration of watermark soil moisture sensors for soil matric potential and temperature. *Plant Soil* 143, 2 (1992), 213–217. DOI: <https://doi.org/10.1007/BF00007875>
- [36] Aqeel ur Rehman, Abu Zafar Abbasi, Noman Islam, and Zubair Ahmed Shaikh. 2014. A review of wireless sensors and networks’ applications in agriculture. *Comput. Stand. Interfaces* 36, 2 (2014), 263–270. DOI: <https://doi.org/10.1016/j.csi.2011.03.004>
- [37] Eric W. Weisstein. 2003. Circle-Circle Intersection. Retrieved from <https://mathworld.wolfram.com/Circle-CircleIntersection.html>.
- [38] Daniel Winkler and Alberto E. Cerpa. 2014. Distributed independent actuation for irrigation control. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings (BuildSys’14)*. ACM, New York, NY, 148–151. DOI: <https://doi.org/10.1145/2674061.2674077>
- [39] Daniel A. Winkler, Miguel Á. Carreira-Perpiñán, and Alberto E. Cerpa. 2018. Plug-and-play irrigation control at scale. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IEEE Press, 1–12.
- [40] D. A. Winkler, R. Wang, F. Blanchette, M. Carreira-Perpinan, and A. E. Cerpa. 2016. MAGIC: Model-based actuation for ground irrigation control. In *Proceedings of the 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN’16)*. DOI: <https://doi.org/10.1109/IPSN.2016.7460680>

Received August 2018; revised October 2019; accepted October 2019

ACM Transactions on Sensor Networks, Vol. 16, No. 3, Article 22. Publication date: May 2020.