2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks

Plug-and-play Irrigation Control at Scale

Daniel A. Winkler
University of California, Merced
dwinkler2@andes.ucmerced.edu

Miguel Á. Carreira-Perpiñán
University of California, Merced
mcarreira-perpinan@ucmerced.edu

Alberto E. Cerpa
University of California, Merced
acerpa@andes.ucmerced.edu

ABSTRACT

Lawns, also known as turf, cover an estimated 128,000km^2 [9] in North America alone, with landscape requirements representing 30% of freshwater consumed in the residential domain [27]. 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 don't revise the strategy over time in response to collected data. In this work, we introduce PICS, 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 PICS 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, PICS also was found to improve quality of service by a factor of 4.0x compared to industry best and 2.5x compared to academic state of the art.

ACM Reference Format:
Daniel A. Winkler, Miguel Á. Carreira-Perpiñán, and Alberto E. Cerpa. 2018. Plug-and-play Irrigation Control at Scale. In *Proceedings of ACM Information Processing in Sensor Networks (IPSN'18).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Turf is the largest irrigated crop by surface area, covering an estimated 128,000 km^2 in North America alone. 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 [27]. 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.

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 [23]. 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, all 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 developed in [32], 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 this work, the control system 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. PICS 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 accross the field.

In this work, PICS 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 using approximated parameters and fixed assumptions. The contributions of this work are as follows: (1) As no manual input is required of the installer, PICS 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 PICS to automatically adapt to unforeseen/changing environmental conditions and seasonal variations, and weather forecasting allows PICS control to react to future weather conditions; (3) To improve system scalability, PICS decouples short- and long-term models, allowing the latter to become spatially independent; (4) As PICS 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, we

0-7695-6377-5/18/$31.00 ©2018 IEEE
DOI 10.1109/IPSN.2018.00008

IEEE computer society

demonstrate that in addition to improved ease-of-use and scalability, PICS reduces water consumption against all baselines, while simultaneously improving quality of irrigation.

2 RELATED WORK

In [32], the distributed irrigation sensor/actuator was introduced, 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 which 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 [32] 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 [32] 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]; The least accurate technique, relying on core samples, is low-cost but *each* measurement requires *a few hours to several days* depending on the soil type. The most accurate 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 parameters, the learning model used by PICS 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, 25]. 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 [8]. 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 [17, 24]. Some providers boast an average 30% reduction in water consumption, but as with all industry irrigation systems, ET-based systems are limited by centralized control, and can not provide site-specific irrigation, reducing potential system efficiency and quality of control.

In order to control intelligently, we must take into account *future* temperature shifts, precipitation, and other effects that affect irrigation requirements as discussed in Section 4.1. 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 [29], 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 [20] 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.

3 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 we like.

The goal of our irrigation system is to keep the turf healthy and in order 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, the volumetric water content (VWC) at which plants

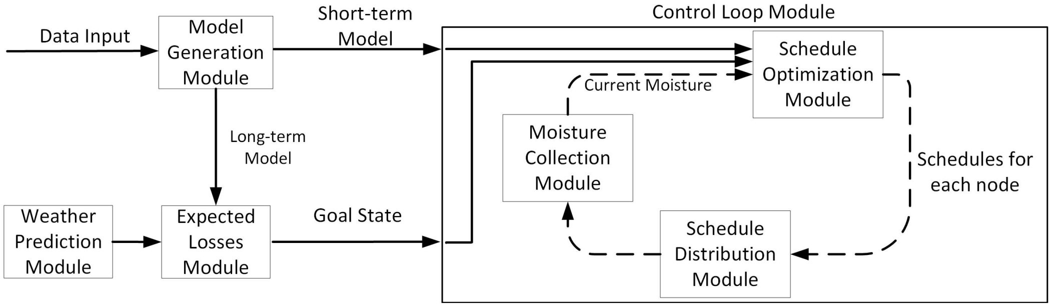


Figure 1: PICS System Architecture

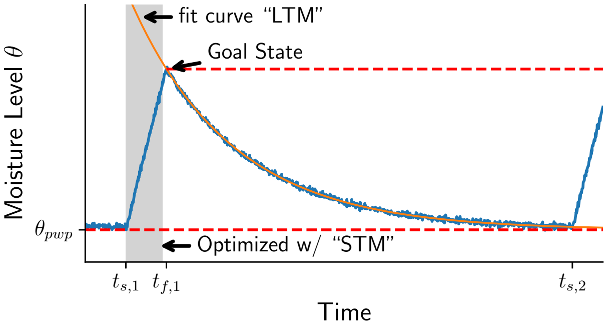


Figure 2: Sample fluid curve across the 24-hour cycle

can no longer extract water from the soil is known as the Permanent Wilting Point, or θ_{pwp}[14, 26, 28]; if a plant spends an extended period of time in soil beneath this threshold, it will begin to wilt and die. To provide ample moisture to the plant at all times, we aim to minimize the amount of time the soil VWC remains below this threshold, as verified in our system with the installation of VWC sensors to constantly monitor soil moisture levels. By achieving this goal, the plant will have the water it needs to thrive.

Figure 2 shows an example 24-hour irrigation cycle. Between day 1's start of irrigation $t_{s,1}$ and finish of irrigation $t_{f,1}$, 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 a pre-computed *Goal State* for this sensor node. During this irrigation time, we utilize a *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.

After day 1's irrigation ends at $t_{f,1}$ as shown in Figure 2, 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 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 amount of water that is lost between irrigation cycles for each individual sensing node, which can then be used to set future *Goal States* in a way that will ensure we stay above minimum VWC θ_{pwp} at all times without wasting water.

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 1 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 [22], and a separate weather prediction module predicts future weather trends in the form of evapotranspiration [24].

Next, using the long-term model, the *Expected Losses* module computes a *Goal State* for each node in the space. This state is computed by taking the minimum acceptable VWC, and added the node's expected moisture losses between irrigation cycles. As later explained in Section 4.3, 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 4.1. In this way, the minimum moisture, experienced right before the next irrigation will begin, should be at or above the minimum VWC threshold. This *Goal State* is later used by the optimization module.

$$\textit{Goal State} = \theta_{\text{pwp}} + \textit{Expected Losses} + ET_{\text{forecasted}} \tag{1}$$

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

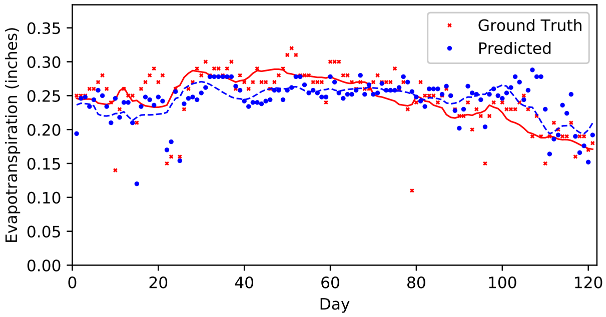


Figure 3: Evapotranspiration Prediction

space, such that the *Goal State* VWC will be reached by the end of the irrigation period. If our short-term model used in optimization is accurate, we can perform this control loop only once. However, for safety and for accuracy, it is advantageous to occasionally re-run this control loop during the irrigation period. 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 [13] 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.

4 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. Towards 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 allow movement that will occur for seconds or minutes, movement that tends to comes to an end when irrigation is completed or soon thereafter. Once this moisture infiltrates fully into the soil, much slower effects begin to take place. 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 [18], and will continue to move for hours or days.

Previous work [32] combines all of these 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 an *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 few hours of irrigation time, allowing optimization to be run on very computationally-weak machines.

4.1 Weather Forecasting

A large portion of losses on the 24-hour cycle is expected to be caused from the effects of evapotranspiration [24], which combines losses caused by solar irradiance, wind, plant transpiration, and other effects due to weather conditions. However, as future weather conditions aren't necessarily similar to past weather trends, we decouple weather effects from the rest of the long-term model by performing separate evapotranspiration prediction. In our research, we found forecasts readily available for standard weather metrics such as temperature, humidity, and rainfall, but were unable to find data sources providing future ET estimates. To enable this feature, we implement our own weather forecasting module.

The four weather variables required to perfectly compute evapotranspiration [17] are solar radiation, temperature, humidity, and wind. In our locale, solar radiation and wind prediction were not available, but temperature and humidity forecasting were available at hourly measurement intervals from a local weather station in our city. With 15 years of historical weather data at hourly intervals[1], we set up a k-nearest-neighbors regression based on similarities of the hourly temperature and humidity vectors. Choosing a k-value of 5, this method will find the 5 days with the most similar hourly weather trend, and find an evapotranspiration value based on the weighted distance from the current day's weather trend. Note that in this work, for simplification, we combine forecasted precipitation and evapotranspiration. In the case of precipitation, the evapotranspiration, rather than being a net loss, would be a net gain, and units remain unchanged.

Figure 3 shows the ground-truth evapotranspiration values as measured by our local weather station, alongside the predicted value from the day before using just forecasted temperature and humidity. Also depicted is a moving average for each, which shows

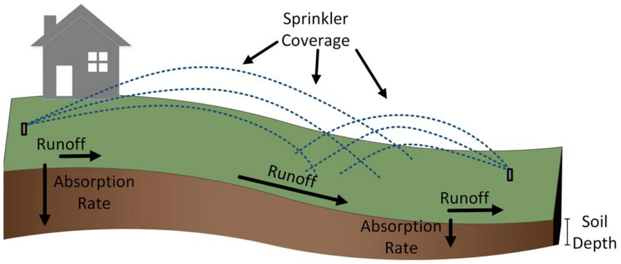


Figure 4: Sources of water movement *during* irrigation (Short-term)

how the two trends vary across the 4 months of weather collection. In all, our prediction is found to have a normalized root mean squared error (NRMSE) value of .164, with 90% of predictions having less than 20% relative error. Outlier values occur when cloud cover or extreme wind affect the solar radiation or wind terms, neither of which we have available for prediction.

4.2 Short-term Model

For later use in optimization in Section 5, we must understand how the actuation of the sprinklers in the system will influence the moisture in the soil as shown in Figure 2. 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 4; 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. These short-term factors are influenced by sprinkler positions, slope of the land, and other characteristics that do not change over time, so this model is trained once automatically when the system is set up. It may be desirable to trigger for the model to be re-trained if the root mean square error of the model against ground truth is seen to exceed a selected threshold, but we leave this to future work.

We choose to use a linear regressor to model these effects using values shown 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 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 $\mathbf{g}$ defines the following relationship:

$$\mathbf{g}(\mathbf{s}_t, \mathbf{f}_t) = \mathbf{s}_{t+\Delta t} \tag{2}$$

To train the short-term model, we run a training cycle of irrigation after installation that triggers each sprinkler one-by-one for a fixed time to identify how moisture levels change in all sensor locations. Moisture levels during our training cycle can be seen for 4 selected devices in Figure 5 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.

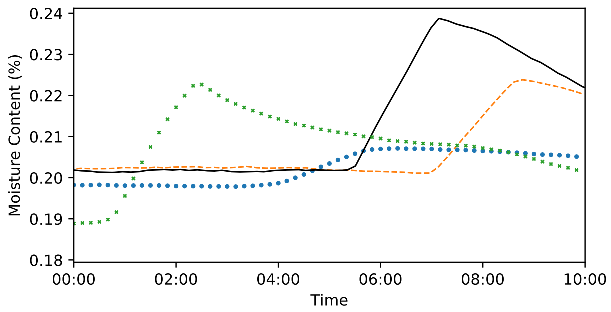


Figure 5: Sensor data from selected nodes during Short-term model training period

Table 1: Short-Term Model Variables

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

The data from our deployment training period is parsed into ∼850 training pairs of $[\mathbf{s}_t, \mathbf{f}_t]$ vectors as inputs, and $[\mathbf{s}_{t+\Delta t}]$ vectors as outputs, and the regressor is trained. At the end of our deployment, we use this regressor to perform single-step prediction across our 25 days of experimentation and compute the Root Mean Squared Error (RMSE) between the vector of predicted VWC against the measured ground truth; by then normalizing to the range between minimum and maximum values of $\mathbf{s}_t$, we get a Normalized RMSE of just 0.2%. By analyzing the predictions for individual sprinklers, we find that the range of errors among our K individual sprinklers falls between 0.08% and 0.25%.

As sensor data is inevitably noisy it is important to choose a Δt that is not too big, which would result in high predictive error or too small, where noise dominates the state signal. Our choice of one minute seemed to work well in practice, but in future work we will analyze so see how this choice impacts prediction accuracy and robustness.

4.3 Long-term Model

To guarantee a high quality of control, we must compute a *Goal State* for each node in the space as a target volumetric water content (VWC) for the *Schedule Optimization Module* to reach as shown in Figure 2. In order to choose a useful *Goal State*, we must consider

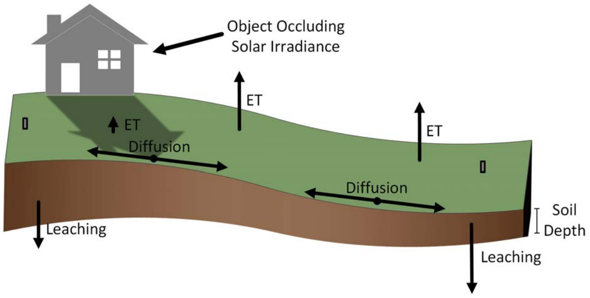


Figure 6: Sources of water movement *between* daily irrigation (Long-term)

both future predicted losses due to weather patterns (evapotranspiration) as discussed in Section 4.1 and 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 6. 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 soil can be on the order of $10^{-2} - 10^{-3}$ cm/s [18]. 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.

Figure 7 shows how moisture losses occur at one selected sensor site across three consecutive days. 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 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 7. In addition, for each day $d \in \{1, \ldots, D\}$, we record the measured evapotranspiration on each day the system is run as 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.

$$\textit{Expected Loss} = \theta_{\text{pwp}} + (\text{LTM}(0) - \text{LTM}(t_{s,d+1} - t_{f,d}) - \frac{1}{D}\sum_{d=1}^{D} ET_d \quad (3)$$

We perform this weather offset 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. 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 PICS to be more generalizable and reactive we choose to utilize local weather forecasting.

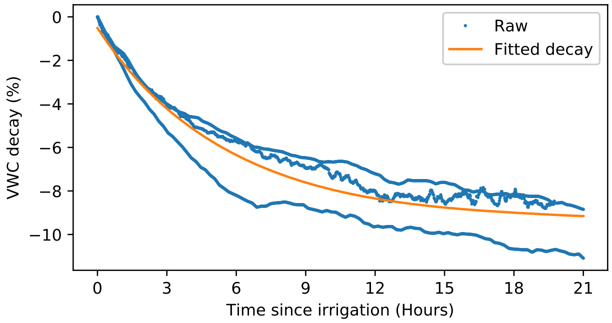


Figure 7: Sample moisture decay fit between irrigation

Table 2: Optimization Variables

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

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 4.2, 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.

5 OPTIMIZATION OVER THE SCHEDULE

We wish to use our models as described in Section 4 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 measured VWC θ_k for each node index $k \in \{1, \ldots, K\}$, we construct the following optimization problem using optimization variables as defined in Table 2. We define $f_{k,t}$ to be the binary actuation of sprinkler k at discrete time index $t \in \{0, \ldots, T\}$. The objective function is the sum of $f_{k,t}$ for all $k \in \{1, \ldots, K\}$, $t \in \{0, \ldots, T\}$, weighted by the water consumption rate of each sprinkler k, c_k, a

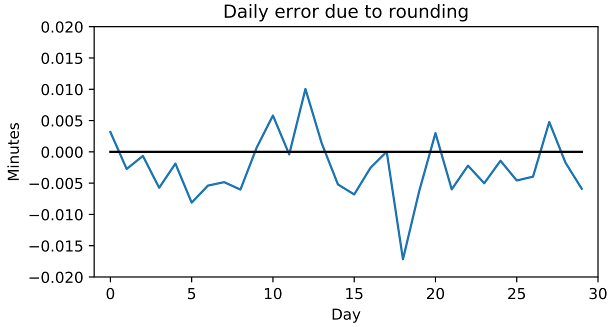


Figure 8: Daily error in irrigation schedules due to rounding of LP solution

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**. The VWC at each discrete sensor location k at temporal index t is defined as $s_{k,t}$. This state variable is assigned the most recently measured sensor value θ_k at starting time index $t = 0$. This moisture level is constrained to remain above the minimum acceptable moisture threshold, θ_{pwp} at all times, and above the goal state at final time index $t = T$. Changes in moisture level as a result of sprinkler actuation is modeled by the linear function **g**, representing our *Short-term Model* as defined in Section 4.

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

$$0 \leq f_{k,t} \leq 1 \qquad k = 1, \ldots, K \quad t = 0, \ldots, T \tag{4b}$$

$$s_{k,t} \geq \theta_{\text{pwp}} \qquad k = 1, \ldots, K \quad t = 0, \ldots, T-1 \tag{4c}$$

$$s_{k,T} \geq \theta_{\text{goal},k} \qquad k = 1, \ldots, K \tag{4d}$$

$$s_{k,t=0} = \theta_k \qquad k = 1, \ldots, K \tag{4e}$$

$$\mathbf{s}_t = \mathbf{g}(\mathbf{s}_{t-1}, \mathbf{f}_{t-1}) \qquad t = 1, \ldots, T \tag{4f}$$

As sprinkler valves can physically be either on or off, sprinkler actuation $f_{k,t}$ 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 basestation. 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 $f_{k,t}$ as a real number within $[0, 1]$, and then rounding the computed optimal value to the closest binary value. This simplification makes the resulting problem a linear program (LP), which are much simpler to solve in practice. In our 4 weeks of deployment, we find that the resulting schedules tend to lie on the 0/1 integer boundary, and that fewer than 1% of actuations require rounding. Figure 8 shows how the amount of irrigation deviates each day due to this rounding, and we can see that the worst day (day 17) has a resulting deviation of just over 1 second, negligible when the total irrigation is on the order of an hour.

We found that 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 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 [19] 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 $2 \times K \times T$ Variables and $2K \times (T + 1)$ 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 basestation.

6 CASE STUDY: LIVE DEPLOYMENT

To perform a fair comparison of our PICS system against a baseline, we installed an irrigation system that allows us to run two control strategies side-by-side. As our campus greenskeepers were unwilling for us to debug an operational campus irrigation system, we were required to officially request a plot of land, design, and install our own system. The approval process took between 2 and 3 months, and system design and installation required hundreds of man-hours. One 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 much longer. Instead, we chose to install the two systems directly side-by-side, where each would have as homogeneous solar exposure, slope characteristic, and soil characteristics as possible.

6.1 Environmental Description

The two irrigation systems were installed side-by-side, and were designed to be identical in hardware, sprinkler coverage, etc. Each irrigation system measured 60'x60', with sprinklers arranged in a 3x3 grid, each 30' from the next. The sprinklers chosen were MP Rotators by Hunter Industries [7], 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', making them ideally suited for our system.

The deployment area was located on a sloped area that drops 3' from the highest to the lowest point. During installation of the sprinkler system, we noted that the topsoil was a "Clay Loam" type, approximately 10" deep. Beneath this layer sat a thick clay layer that went beyond 3' 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.

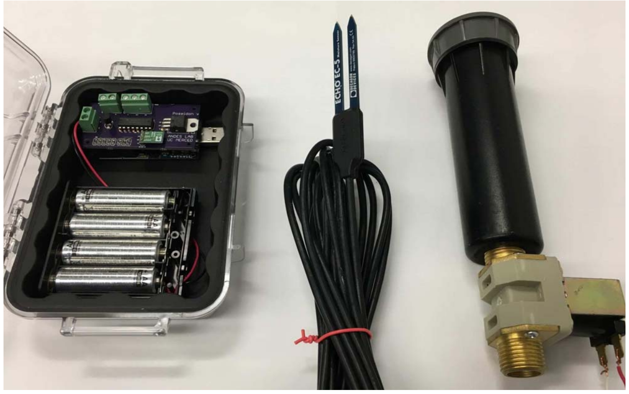


Figure 9: Sensing and actuation node

6.2 Hardware Description

The hardware used in our experiments stemmed from the design introduced in [32], with battery life and system safety as primary concerns. Control was provided by a latching solenoid, which requires a 50ms 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 ±3% and power consumption of just 10mA for 10ms. 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. A standalone board was developed to sit on the General Purpose Input/Output (GPIO) pins of the Tmote Sky, whose purpose was to route power from the attached 4xAA battery source to the peripheral devices. With these peripherals installed, our devices could communicate with eachother, 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 9.

Collocated with the irrigation system was a basestation 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 3 and the optimization as described in Section 5 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 7, the only equipment failure that occurred in our deployment was a failed USB connection between the Tmote and the Raspberry Pi, possibly 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 massive water loss.

6.3 Baseline Strategies

To allow 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 PICS system to two baseline systems, an Evapotranspiration control strategy, the current industry leader in system efficiency, and MAGIC [32], the current state-of-the-art in academia. As the University irrigation systems operate on a daily schedule, all baseline systems and PICS was configured to irrigate daily as well. The PICS system would operate identically on a different irrigation cycle with no reconfiguration necessary.

6.3.1 Evapotranspiration. Evapotranspiration is an estimate of moisture lost from soil, subject to weather factors. In the current standard, computing evapotranspiration requires wind, temperature, humidity, and solar irradiance measurements. Many weather stations are available to the public that calculate and provide evapotranspiration data based on measurements of the other weather factors. To mimic an evapotranspiration controller, we query a local weather station for the previous day's ET losses, which is provided in units of surface water height. With this information, we can simply use our sprinkler datasheet for surface application rate to compute exactly how many minutes the system should be activated to directly replace the previous day's losses. In a commercial evapotranspiration 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 [6] and Rain Bird [12], we were unable to find the safety margin they use 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 PICS system is able to achieve both goals, water savings *and* improved quality of service.

6.3.2 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 6.1 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", sitting atop a deep clay layer. To allow fair comparison to the PICS 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.

6.4 Performance Metrics

The fields of plant physiology and soil physics make it clear that to cultivate healthy plants, turf must be in an environment with an abundance of necessary minerals (fertilized, kept in healthy soil), must receive adequate solar exposure, and must be within roots'-reach of an adequate supply of water. If soil is kept too dry, the plant will be unable to suck the necessary moisture out of the soil. This level of moisture is known as the permanent wilting point (pwp) [14, 26, 28], and keeping soil below this threshold of moisture for an extended period of time will cause the plant to eventually wilt and die. Although the irrigation system has no control over solar exposure and soil nutrients, it has direct control over the moisture levels in the soil. For this reason, our primary metric for

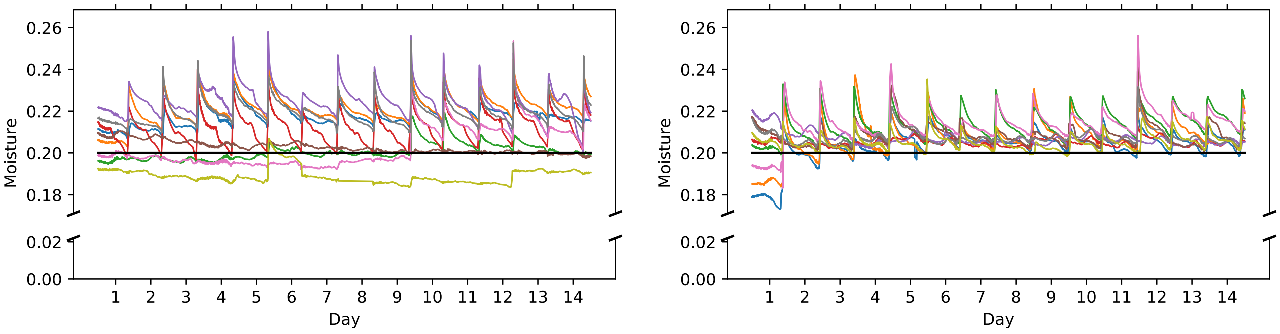


Figure 10: Collected VWC data across deployment for all Evapotranspiration (left) and PICS (right) nodes

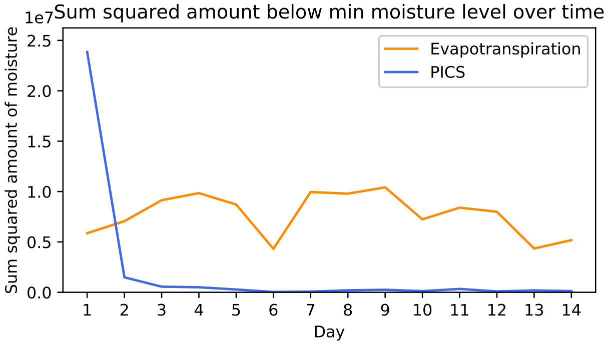


Figure 11: ET vs PICS quality of service results

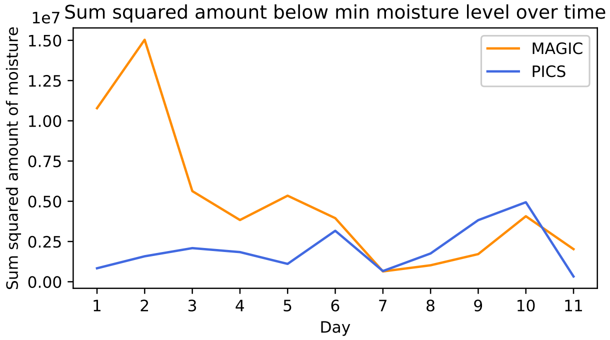


Figure 12: MAGIC vs PICS quality of service results

irrigation quality is the system's ability to maintain a soil moisture above this threshold at all times at all of our measured locations. By doing so, we are guaranteeing that the plant has sufficient moisture to be healthy. In this paper, we call this the quality of service of 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.

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.

7 EXPERIMENTAL RESULTS

In this section, we discuss the experimental results of PICS 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 6.1, analyzing the quality of service, water consumption, and moisture uniformity metrics discussed in Section 6.4.

7.1 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 10 shows the raw moisture data for each sprinkler in the field for both ET and PICS. The center line shows the permanent wilting point (pwp). We can see that the ET system spends more time and in some cases it is well below that mininum threshold for many of the moisture sensors. In order 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 11 and 12 for the experiments comparing PICS 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.

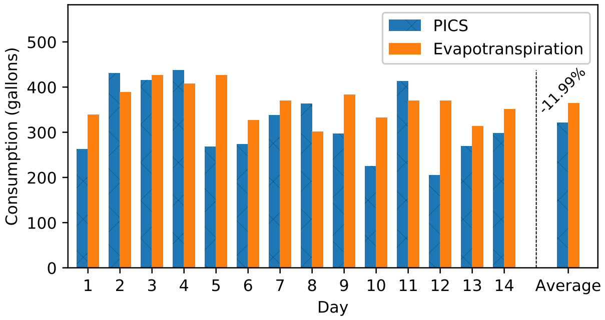


Figure 13: Consumption of ET vs PICS

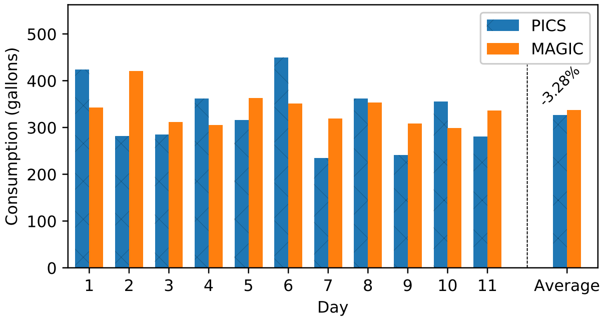


Figure 14: Consumption of MAGIC vs PICS

In our deployment of PICS against the evapotranspiration control strategy in Figure 10, we see that one node in the control system is well below the threshold we wish to hit. This emphasizes the limitations of ET and the core of our work. The irrigated regions don't receive moisture the same way, and without learning these eccentricities, it's very difficult to provide the best quality of service. It should be noted that before the PICS system was turned on, the ET control strategy was used to irrigate both irrigation systems. This can be seen in day 1 in Figure 10, where several sensors have moisture levels below our threshold. However, in the first 4 days, the PICS system learns these increased needs and applies tailored moisture to raise them above the threshold. This is a common problem in irrigation systems, as uniform irrigation across the field, without understanding local variations results in moistures that can vary wildly. In order to make this uniformly-irrigating system provide perfect quality, we would have to irrigate *all* sprinklers enough to raise the driest area above our threshold. This would be a significant waste of water, as the rest of the space would be severely over-watered. In the comparison against MAGIC in Figure 12, we can see that several days are spent with a poorer quality of service in comparison to PICS. On days 1 and 4 in particular, it's 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, PICS spends an average of 4.04 times less than the ET system, and ignoring the first day, we average 24.7 times less below the threshold. When compared to MAGIC, PICS spends an average of 2.47 times less than MAGIC. Our system provides significant improvements with respect to quality of service than the other irrigation systems.

7.2 Water Consumption

When a decision must be made to switch to a new landscape irrigation control system, 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 providing for each sprinkler is pressure-regulated to the industry standard, 40psi. 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 PICS and MAGIC vs PICS, as shown in Figures 13 and 14 respectively.

As discussed in Section 7.1, the experimental system started with moisture levels significantly beneath our desired threshold. For this reason, the first 5 days of PICS control had steadily improving quality of service as it learned its models and raised moisture 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.

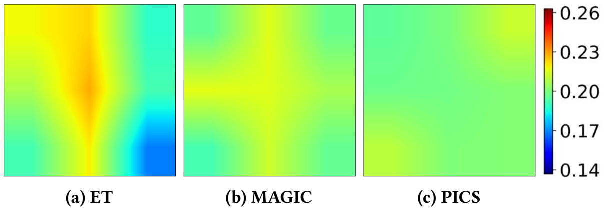
Day 11 of our first deployment saw increased water consumption as well, caused by a hardware malfunction. A command telling 4 of the 9 PICS nodes to "Stop irrigation" was lost due to a faulty USB connection to the basestation mote, causing unintentional irrigation that was not corrected until failsafes in the node's firmware automatically disabled irrigation. This caused the PICS system to consumed more water than intended, as shown on day 11 in Figure 10. We can also see PICS'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 PICS when compared to MAGIC as seen in Figure 14 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 PICS system reduced water consumption by an average of 11.99% compared to the ET system, and 3.28% compared to the MAGIC system.

7.3 Moisture Uniformity

Moisture uniformity is a good side-effect, but is not considered a primary goal. In particular, 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 towards a uniform distribution as the moisture levels approach the uniform minimum moisture *just before* irrigation.



(a) ET (b) MAGIC (c) PICS

Figure 15: Average VWC (%) of compared systems across deployments

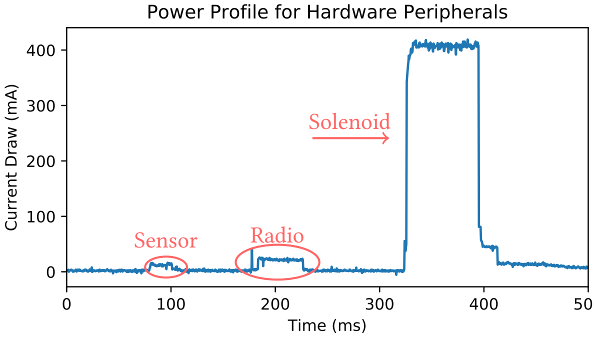


Figure 16: Energy profile of the sensor node

Figure 15 shows the average volumetric water content (VWC) colormap of the three compared systems as periodically sampled across the entire deployment. 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 has the least uniform coverage, with a heavy 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 PICS had a more uniform moisture distribution, with MAGIC being slightly above the minimum in a cross-like pattern, and PICS being slightly above the minimum in the upper right and lower left corners.

7.4 Energy Consumption

In our devices, the three peripherals that consume significant energy are the sensor, solenoid, and the radio. However, through clever use of these peripherals, this system can achieve a substantial system lifetime using our current power source of 4xAAs [3]. Each sensor sample requires 10mA of power for 10ms, and each flip of the latching solenoid requires 400-450mA of power for 50ms. In our system, to ensure we don't cut power too early, we add a safety band of 50% on the timing on both of these devices, triggering for 15ms and 75ms for the sensor and solenoid, respectively. The tmote sky [15] radio consumes 23mA max when in transmitting mode.

In our 25 day deployment, we found that on average our control strategy 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 1% duty cycle ratio across the full 24 hours for communication using techniques such as Low Power Listening [30], even with our conservative timing of the peripherals, 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 continue sampling the data and storing it in memory to be sent at the beginning of the irrigation cycle, this lifetime is easily extended to ~11.9 years.

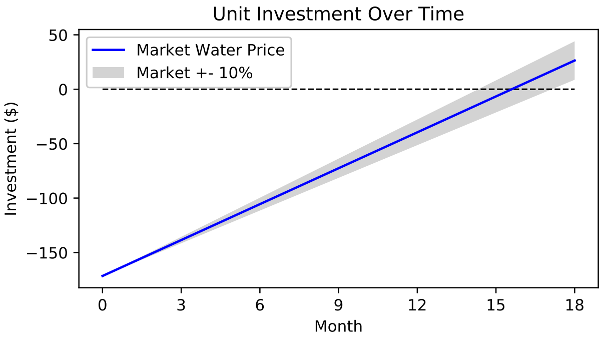


Figure 17: Return on investment curve

Table 3: Sprinkler Node Manufacture Cost

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

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 3, 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 17, we show how PICS 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 PICS'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 17 is a ±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.

PICS'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 2.

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 (±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 [31].

When setting up our system, we chose to use a control timestep of 1 minute, and continued to use this to learn our short-term model through our experiments. In future work, we will perform a more in-depth analysis to see how the choice of this timestep affects predictive power and practicality in irrigation control.

As PICS 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 PICS 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 heterogenous minimum moisture constraints will not require any modification to our processing pipeline.

PICS 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 [11, 21] to provide supplemental long-term feedback to ensure these locations are also satisfied at all times.

10 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 PICS, 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, PICS was found to reduce turf exposure to unhealthy levels of moisture by a factor of 4.0x and 2.5x with respect to the two systems mentioned above. The PICS system is expected to return its investment in 14-17 months based on water savings alone.

11 ACKNOWLEDGEMENTS

We would like to thank our referees and our shepherd Xiaofan (Fred) Jiang for their feedback and assistance. This material is based upon work partially supported by the National Science Foundation under grants #CNS-1254192 and #CNS-1430351.

REFERENCES

[1] CIMIS weather data. http://www.cimis.water.ca.gov/.

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

[3] Energizer ultimate lithium. http://data.energizer.com/pdfs/l91.pdf.

[4] Freshwater crisis. http://environment.nationalgeographic.com/environment/freshwater/freshwater-crisis/.

[5] GNU linear programming kit. https://www.gnu.org/software/glpk/.

[6] Hunter: Advanced evapotranspiration weather control. https://www.hunterindustries.com/en-metric/irrigation-product/sensors/et-system.

[7] Hunter MP rotator. hunterindustries.com/irrigation-product/nozzles/mp-rotator.

[8] Hunter rain-clik rain detection. hunterindustries.com/irrigation-product/sensors/rain-clikr.

[9] Looking for lawns. http://earthobservatory.nasa.gov/Features/Lawn/printall.php.

[10] Measuring hydraulic conductivity for use in soil survey. https://www.nrcs.usda.gov/Internet/FSE_DOCUMENTS/nrcs142p2_053204.pdf.

[11] Photonics - plant health imaging filters. https://www.photonics.com/Product.aspx?PRID=60994.

[12] Rain bird evapotranspiration manager. https://www.rainbird.com/documents/turf/bro_ETManager.pdf.

[13] Raspberry pi. https://www.raspberrypi.org/.

[14] Soil quality indicators. https://www.nrcs.usda.gov/Internet/FSE_DOCUMENTS/nrcs142p2_053288.pdf.

[15] Tmote sky. http://www.snm.ethz.ch/Projects/TmoteSky.

[16] UGMO irrigation. http://www.ugmo.com/.

[17] R. G. Allen, L. S. Pereira, D. Raes, M. Smith, et al. Crop evapotranspiration-guidelines for computing crop water requirements-FAO irrigation and drainage paper 56. *FAO, Rome*, 300(9):D05109, 1998.

[18] J. Bear. *Dynamics of Fluids in Porous Media*. Dover Civil and Mechanical Engineering Series. Dover, 1972.

[19] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *CoRR*, 2014.

[20] J. Cai, Y. Liu, T. Lei, and L. S. Pereira. Estimating reference evapotranspiration with the FAO penmanâĂŞmonteith equation using daily weather forecast messages. *Agricultural and Forest Meteorology*, 145(1):22–35, 2007.

[21] L. Chaerle and D. V. D. Straeten. Seeing is believing: imaging techniques to monitor plant health. *Biochimica et Biophysica Acta (BBA) - Gene Structure and Expression*, 1519(3):153 – 166, 2001.

[22] E. C. Childs. *An introduction to the physical basis of soil water phenomena*. A Wiley Intercience Publication John Wiley And Sons Ltd,; London; New York; Sydney; Toronto, 1969.

[23] T. Harter and J. R. Lund. Addressing nitrate in california's drinking water. Technical report, University of California, Davis, 2012.

[24] M. E. Jensen, R. D. Burman, and R. G. Allen. Evapotranspiration and irrigation water requirements. ASCE, 1990.

[25] Y. Kim, R. G. Evans, and W. M. Iversen. Remote sensing and control of an irrigation system using a distributed wireless sensor network. *IEEE Transactions on Instrumentation and Measurement*, 57(7):1379–1387, July 2008.

[26] M. Kirkham. *Principles of Soil and Plant Water Relations*. Elsevier Science, 2004.

[27] M. A. Maupin, J. F. Kenny, S. S. Hutson, J. K. Lovelace, N. L. Barber, and K. S. Linsey. *Estimated use of water in the United States in 2010*. U.S. Geological Survey Circular 1405, 2014.

[28] S. McCreath and R. Delgoda. *Pharmacognosy: Fundamentals, Applications and Strategies*. Elsevier Science, 2017.

[29] M. J. Oliver. Evapotranspiration forecasting irrigation control system, Dec. 1997. US Patent 5,696,671.

[30] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107. ACM, 2004.

[31] E. A. Spaans and J. Baker. Calibration of watermark soil moisture sensors for soil matric potential and temperature. *Plant and Soil*, 143(2):213–217, 1992.

[32] D. A. Winkler, R. Wang, F. Blanchette, M. Carreira-Perpinan, and A. E. Cerpa. MAGIC: Model-based actuation for ground irrigation control. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Apr. 2016.