DICTUM: Distributed Irrigation aCtuation with Turf hUmidity Modeling

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

Lawns make up the largest irrigated crop by surface area in North America and carry with it a demand for over 7B gallons of freshwater each day. Despite recent developments in irrigation control and sprinkler technology, state-of-the-art irrigation systems do nothing to compensate for areas of turf with heterogeneous water needs. In this work, we overcome the physical limitations of the traditional irrigation system with the development of a sprinkler node that can sense the local soil moisture, communicate wirelessly, and actuate its own sprinkler based on a centrally computed schedule. A model is then developed to compute moisture movement from runoff, absorption, and diffusion. Integrated with an optimization framework, optimal valve scheduling can be found for each sprinkler node in the space. In a turf area covering over 10,000ft^2, two separate deployments with four weeks of fine-grained data collection show that DICTUM can reduce water consumption by 23.4% over traditional campus scheduling, and by 12.3% over state-of-the-art evapotranspiration systems while substantially improving conditions for plant health. In addition to environmental, social, and health benefits, DICTUM is shown to return its investment in 16 to 18 months based on water consumption alone.

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: Distributed actuation, irrigation control, fluid flow model, linear program optimization, experimental evaluation, water use minimization, quality of service for plant health

ACM Reference format:
Daniel A. Winkler, Robert Wang, Francois Blanchette, Miguel Á. Carreira-Perpiñán, and Alberto E. Cerpa. 2019. DICTUM: Distributed Irrigation aCtuation with Turf hUmidity Modeling. *ACM Trans. Sen. Netw.* 15, 4, Article 41 (October 2019), 33 pages.
https://doi.org/10.1145/3342514

1 INTRODUCTION

Only 1% of Earth's water is fresh and available for use [4]. Due to its scarcity, there is high incentive to reduce its usage across the board. In North America, turf, also known as lawn, is the

A previous version of this work, titled "MAGIC: Model-Based Actuation for Ground Irrigation Control" [38], has appeared in the Proceedings of the 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'16).
Authors' addresses: D. A. Winkler, R. Wang, F. Blanchette, M. Á. Carreira-Perpiñán, and A. E. Cerpa, University of California, Merced, 5200 North Lake Rd., Merced, CA 95343, USA; emails: dwinkler2@andes.ucmerced.edu, {rwang6, fblanchette, mcarreira-perpinan}@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 ACM 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.
© 2019 Association for Computing Machinery.
1550-4859/2019/10-ART41 $15.00
https://doi.org/10.1145/3342514

largest irrigated crop by surface area, covering over 128,000km^2 [8] and was estimated in 2015 to consume in excess of 9B gallons of freshwater each day [16]. With a historic drought afflicting the western United States following a similar shortage in the southeast United States, improved irrigation efficiency at this massive scale can help reduce the strain on limited freshwater reserves.

Although we wish to reduce water consumption as much as possible, the primary goal of these irrigation systems is to maintain plant health. To keep turf healthy, a proper amount of water must be periodically applied. Providing too little water to the turf will cause it to turn brown and die. Although traditional irrigation control strategies generally over-irrigate to be safe, this can cause its own problems. In the short-term, excess irrigation can cause increased waste caused by evaporation and runoff, and in the long-term can cause root rot that will kill the plant. Furthermore, over-watering can cause erosion of the surrounding soil and even leach unsafe fertilizer chemicals beyond the root zone and into the ground water, as occurred in California's Salinas Valley and Tulare Lake Basin, investigated by Reference [27].

Improper irrigation is the cause of these issues. Great improvements in irrigation system design have been made recently: new sprinkler heads apply water much more slowly to avoid runoff and leaching [5], and new irrigation controllers schedule irrigation using weather data to take into account the water lost each day due to evaporation and plant transpiration, known coupled as evapotranspiration. Even the best control strategies still behave as though all turf requires the same amount of water, when in fact there often exist large variations in soil type and depth, topography, and direct sunlight. If this information were utilized, every location throughout the irrigated space could be given the amount of water it needs. However, as the infrastructure of traditional irrigation systems is usually configured for each valve to actuate many sprinklers, such a system could not even make proper use of fine-grained water requirement information, as all sprinklers must be actuated for the same amount of time.

Our contributions addresses both of these limitations. First, we develop a computationally light model that uses characteristics of the irrigated space to analyze the fundamental causes of fluid movement. This model is then integrated into an optimization framework to allow for optimal valve scheduling to be computed. The second contribution is the development of the DICTUM sprinkler node, capable of actuating its attached sprinkler, sensing the moisture in its surrounding environment, and communicating wirelessly with its sister nodes in the environment. With our experience of two large-scale irrigation system deployments over a total of seven weeks, with four weeks of fine-grained data collection and monitoring, we demonstrate that the model-based DICTUM system can help provide more precise irrigation control to turf areas, reducing water usage and substantially improving the quality of service over common practice and state-of-the-art control strategies.

2 RELATED WORK

To accomplish the minimization later described in Equation (10a), we must know how fluid moves across and through soil. This is well studied in the field of soil physics; very accurate models including Hydrus [7] and Comsol Multiphysics' Subsurface Flow Module [1] exist, which solve partial differential equations for pressures that exist between soil and water particles in the porous media. Although very accurate, these models are designed to compute water flow and absorption on a much smaller scale than what is necessary for approximate irrigation control. As such, they each require heavy processing and take a tremendous amount of time to complete.

An overly complicated fluid model makes optimization unreasonable, so alternative methods are considered to simplify this complex system. These methods include the incorporation of observations of an accepted fluid flow model into a simplified model through the use of Data Assimilation. Used to predict states of advanced systems given a particular input such as weather forecasting [29] and large-scale hydrological patterns from satellite images [33], data assimilation is often used to create approximated system models that can then be used for optimization. However, the immense size and discontinuity of the solution space can lead to an approximated model that is unable to reasonably predict outcomes of the system. Another option, called Lumped Element Modeling, is a method of breaking complex problems into simpler "lumped" sub-components. This method has been used to create a simplified model of fluid jets for prototype analysis [24] and to model aortic blood flow from arterial pressure in humans [36], among other applications. Although such simplifications sacrifice accuracy, these approximations can greatly improve performance.

Researchers in Reference [23] build a one-dimensional soil water balance model to simulate the vertical movement of moisture in an irrigated space based on soil characteristics, irrigation schedule, real-time weather data, and irrigation infrastructure. With this simple model of moisture depth, an interactive tool was designed for homeowners and landscapists to allow quick evaluation of an installed system. This system provides some key insight on moisture changes, such as evapotranspiration, that take place on a daily timescale, but over-simplifies short-term effects like runoff, which can provide key insights into the movement of water across the surface. Although it provides evaluation of a particular schedule, no strategies are offered for schedule improvement.

In Reference [37], a model was created using a cellular automata approach that incorporated irrigation, surface, and sub-surface flow. By analyzing this model using an optimization framework, it was determined in simulation that system efficiency can be improved by increasing the granularity of control. However, due to model complexity and problem size, the optimization framework had difficulty finding a globally optimal solution with large numbers of timesteps and sprinklers.

Irrigation controllers exist that utilize one or more soil moisture sensors spread throughout the turf [15, 19] to gauge need for irrigation. However, as these systems do not perform any modeling, they can easily become mis-calibrated (from placing different sensors in regions of turf with different soil depths, for instance). Furthermore, these systems still have the same physical constraints as a traditional irrigation system; it can not target irrigation to regions that need more or less.

Work done in Reference [32] identifies potential improvements to agriculture irrigation systems. On-site and remote control systems are discussed, as well as their learning curves, which may prevent unskilled farmers from adopting them. Bottlenecks are recognized in system development, standardization of technology, and business models that must advance before control system alternatives become accepted. Although a control system is not introduced, this work goes to show that control strategy improvements are necessary and highly sought after.

Becoming more prevalently used in irrigation, evapotranspiration technologies [17, 28] use weather stations to monitor air temperature, humidity, and other factors to estimate water lost to the environment throughout the day. Using this information, irrigation systems can use knowledge of their sprinklers' coverage to compute required on-time for water replacement. As this has become widely adopted, it will be used as side-by-side comparison with the DICTUM system.

3 SYSTEM OVERVIEW

Figure 1 shows an overview of the DICTUM system architecture. Our irrigation control system uses multiple modules to provide control to the space. To explain how these modules work together to create a fluid processing pipeline, we first describe their roles. This processing pipeline is described at irrigation time each day, when schedule generation occurs.

Our deployment consists of a distributed network of sensing and actuation (DICTUM) nodes, integrated into the plumbing infrastructure of the irrigation system. Each DICTUM node is equipped with a soil moisture sensor, a solenoid to control the flow of water, and a mote to provide radio communication capabilities. To allow the DICTUM system to react to changing soil conditions, an attached volumetric water content (VWC) sensor is periodically sampled from the environment

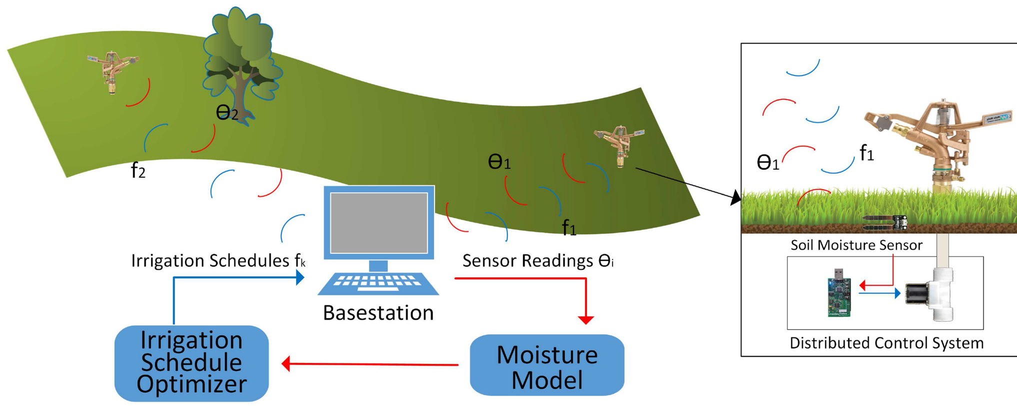


Fig. 1. DICTUM system architecture.

by each sprinkler node in the space. This collected data is then routed through the wireless sensor network to the *Basestation*, interfacing between the 802.15.4 network and another communication medium such as an ethernet or 4G network. Once received, this data is then incorporated with the sensor readings collected from other nodes to create a "snapshot" of the soil moisture across the entire space.

Once the current state of VWC in the soil is determined, it is fed into the *Moisture Model* for integration. The *Moisture Model*, described in detail in Section 4, contains a mathematical formulation for moisture movement throughout the system. This model is all-inclusive, modeling the characteristics of the irrigation system and soil characteristics within the space, providing means to calculate the conductivity through the soil. With the VWC collected, the model is passed to the *Irrigation Schedule Optimizer* for analysis.

The *Irrigation Schedule Optimizer* sets up and solves the following constrained optimization problem: The fluid flow model is incorporated as equality constraints at each spatial location and time, which must be satisfied for an optimal solution to result in a valid flow. To ensure adequate moisture levels across the space and thus a high quality of service, a goal water-saturation level provides inequality constraints at each spatial location at the end of irrigation. Although in principle the PDEs defining the model are nonlinear, we linearize them to make the optimization problem convex, without local optima, and easier to solve computationally. The final result is a linear program that, although large (due to the discretization), can be solved accurately in a reasonable time. As the objective function minimizes the total water consumption of the irrigation system, the solution provides optimal activation schedules for each actuation node in the space while maintaining minimum moisture levels.

Once the schedules are received by the *Basestation*, they are disseminated through the wireless sensor network to their respective DICTUM node. Upon reception of a schedule, the *Distributed Control System* routes power to the attached solenoid following the received schedule, allowing water to flow to the sprinkler.

4 MODEL DEVELOPMENT

We wish to model a particular irrigation system and use this information to find improved control techniques. This two-dimensional model incorporates water movement from the sprinkler heads to the ground, across the surface, through the sub-surface, as well as absorption into the soil. Although many models exist that describe one or more of these components, we present here the first model that efficiently combines them.

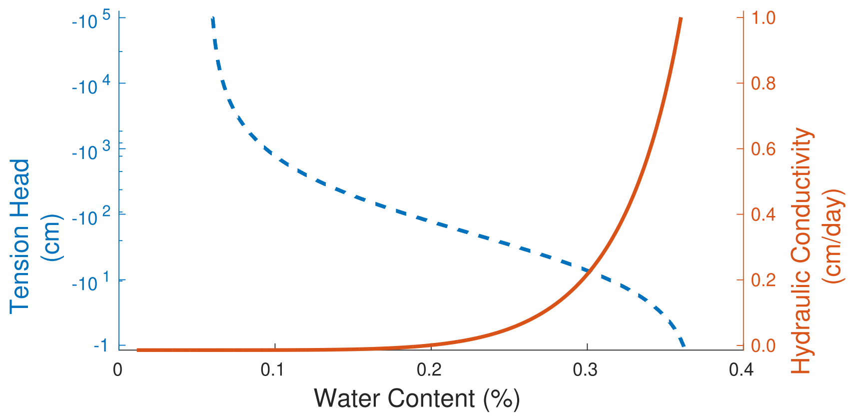


Fig. 2. Sample retention and hydraulic conductivity.

4.1 Soil Characteristics

We first emphasize the differences between soil and typical porous media. Generally, a porous medium maintains constant characteristics across its entire range of saturation. However, in soils, two functional relationships govern the retention and movement of water through soil. First, an attraction exists between water and the soil particles, known as matric suction. When the soil experiences very low levels of saturation, the matrix exerts a strong suction, trying to pull water from the surrounding environment. The relationship between tension head and volumetric water content is known as the "water retention curve," a characteristic equation of the soil type as defined in Reference [25]. This relationship, as shown on the left axis of Figure 2, strongly impacts the movement of water through the soil, as dry soil will act as a sink until increased saturation is reached. Second, the hydraulic conductivity of soil is dependent on the local volumetric water content. Due to the matric suction, the soil will increasingly resist the movement of moisture as the volumetric water content decreases. A typical relation for hydraulic conductivity is also given in Reference [25] and is shown on the right axis of Figure 2.

4.2 Fluid Flow Model

We model water displacement above the soil surface and through the subsurface of the soil as flow through two different porous media. Fluid flow through porous media is well studied, dating back to the work of Henry Darcy [22], now known as Darcy's Law. To model the movement through the soil, we use Darcy's Law for isotropic porous media:

$$\vec{u} = \frac{\kappa}{\eta}(-\nabla P + \vec{\tau}), \tag{1}$$

where P is the pressure, $\vec{\tau}$ is the tangential component of gravity along the surface of the porous media, $\vec{u}$ is the fluid velocity averaged over the thickness of the soil, and other quantities as defined in Table 1. This model assumes that porous media everywhere has the same dependence on water content. As our model tracks soil moisture at small scales, such a simplification is more practical than the alternative of collecting and analyzing samples across the entire space.

Darcy's equation requires the determination of the pressure, which is generally linearly related to the amount of water above the point in question. In our model (see Figure 3), we compute the depth-averaged subsurface flow by considering a soil depth L. In this case, the mean pressure driving flow includes the weight of water in the subsurface above a point, $\rho g L\theta$, the weight of water on the surface, $\rho g h$, where h is the height of water above the soil surface, and the matric

Table 1. Model Variable Reference

Variable	Usage	Variable	Usage
f_k	Actuation function of sprinkler k	ρ	Fluid density
c_k	Coverage of sprinkler k	ζ	Sub-surface boundary constant
θ	Volumetric soil moisture content	μ	Surface boundary constant
h	Surface fluid height	α_h	Surface flow parameter
$\vec{u}$	Velocity of water in soil	$K(\theta)$	Hydraulic conductivity
$\vec{v}$	Velocity of water on surface	$\psi(\theta)$	Matric suction
κ	Soil permeability	F_s	Fluid from sprinklers
κ_g	Grass permeability	$\vec{\tau}$	Tangential component of gravity
η	Fluid viscosity	θ_{pwp}	Minimum θ for Healthy Plants
L	Thickness of soil (m)	ϕ_s	Porosity of soil layer

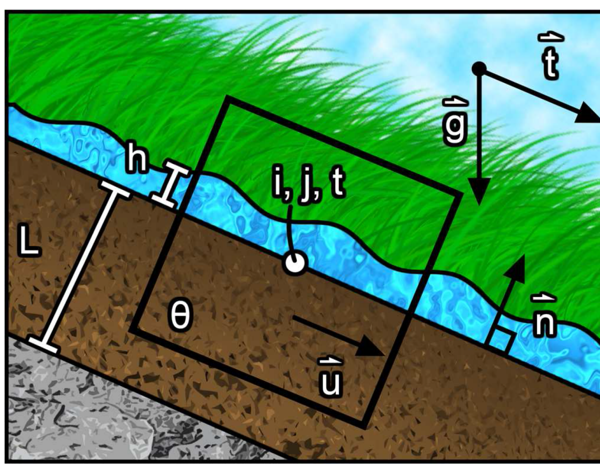


Fig. 3. Physical model unit diagram.

suction of the soil $\rho g \psi(\theta)$. We thus express the mean pressure in soil as $P = \rho g(h + L\theta + \psi(\theta))$. We can therefore express the liquid velocity in the subsurface as

$$\vec{u} = -K(\theta)\nabla h + \frac{K(\theta)\vec{\tau}}{\rho g} - K(\theta)(L + \psi'(\theta))\nabla\theta, \tag{2}$$

where we defined the hydraulic conductivity as $K(\theta) = \rho g \kappa(\theta)/\eta$.

To track the time-rate of change of the volumetric soil moisture content, θ, we use the divergence of the moisture flux, $\vec{u}\theta$, and the inflow from surface water, $\zeta h K(\theta)$, where $\zeta = 1/(L^2\phi_s)$ is a proportionality constant mapping surface water height to volumetric content in the subsurface based on soil porosity and depth, calculated by balancing the pressure gradient with the soil permeability. We thus have

$$\frac{\partial\theta}{\partial t} = -\nabla \cdot (\theta\vec{u}) + \zeta h K(\theta). \tag{3}$$

As the velocity, $\vec{u}$, is itself the gradient of the volumetric soil moisture content, the movement of water will effectively behave as a diffusive process, moving sub-surface moisture towards areas of lower concentration. With the inclusion of a gravity term, $\vec{\tau}$, in the velocity equation, water will tend to move in the direction of steepest descent. Last, by allowing water to move through the boundary from the surface into the sub-surface, we allow irrigation on the surface of the soil to positively affect the amount of water moving through the sub-surface. Together, these terms result in sub-surface water movement that realistically depends on volumetric water content, local topography, and surface conditions.

In addition to tracking the soil moisture, we need to determine the height of the surface water, h. Depending on the species of lawn chosen to model, a square inch of turf can have tens to hundreds of blades of grass, each of which will impede the movement of water across the surface of the soil. By comparing the inertia of the fluid and the drag forces caused by the blades of turf, we find that for surface water velocities less than 1cm/s, the surface flow through the turf can be modeled as fluid flow through a (very) porous medium. Additional information and the calculations that support this approach can be found in Appendix A. The velocity of water through the turf averaged over the height h, denoted by $\vec{v}$, is therefore computed using Darcy's Equation for isotropic porous media, as was done for the sub-surface velocity in Equation (2):

$$\vec{v} = \frac{\kappa_g}{\eta}(-\rho g \nabla h - \vec{\tau}) = -\alpha_h \nabla h - \frac{\kappa_g}{\eta}\vec{\tau}, \tag{4}$$

where the fluid density, permeability, and gravity terms have been absorbed into α_h. Similarly to sub-surface movement, the velocity through the layer of grass is dependent on the orientation of the surrounding topography, as well as the viscosity and density of the fluid.

The permeability used in Equation (4), κ_g, is not as well defined as that of the sub-surface fluid flow. The shape, size, and density of the grass layer is dependent on the species of grass, as well as its health. Research has been conducted [21, 34] for the application of filter design that defines the permeability of a fibrous media based on its overall porosity and the size of the fibers. Using the proposed method, we find an approximate permeability of 10^{-5}m, a value similar to that of well-sorted gravel. This completes the description of the velocity of surface water, and we may use this velocity to keep track of the height of water on the surface:

$$\begin{aligned} \frac{\partial h}{\partial t} &= -\nabla \cdot (h\vec{v}) + F_s - \mu h K(\theta), \\ F_s &= \sum_{k=1}^{n} c_k f_k(t), \end{aligned} \tag{5}$$

where F_s is the rate of irrigation, determined using the activation $f_k(t)$ of sprinkler k at time t and coverage c_k of sprinkler k. The amount of water lost to soil is the same as that added to soil moisture, converted from soil moisture to pure water, where $\mu = \zeta L \phi_s / \phi_g$, with ϕ_s, ϕ_g as the porosities of soil and grass, and L is the soil depth. We note that evaporation and leaching terms were not included in our formulation, due to the way our case study was conducted. At the request of campus authorities, all irrigation was performed in late evening, providing ample time to absorb into the soil and allowing only minimal evaporation to occur. Although these terms are omitted in the present study, they could easily be introduced for an application that requires them.

4.3 Boundary and Initial Conditions

For the PDE-based model to accurately represent the movement of moisture through the turf and soil, we must also specify boundary conditions. Primarily, these boundary conditions must capture how moisture moves in and out of our domain on the sides and on the bottom. In our deployment location, the turf is a sod layer with a thickness of approximately 1.5 inches. Below this sod layer lies a thick layer of clay. As the hydraulic conductivity of clay is 60–3700× smaller than the conductivity of a loamy soil [25] and our optimization only includes movement of water within the first couple of hours following the start of irrigation, we consider that leaching into deeper soil is negligible, and thus omitted. The boundary conditions on the sides of the surface assume an expanse of identical surface without fluid so that $h = 0$. With this constraint, any water on the surface located directly adjacent to the boundary will begin to lose moisture in the direction of the boundary subject to gravity. Likewise, the boundary conditions for the sides within the soil layer assume a surround of soil with a fixed volumetric water content of $\theta = \theta_{\text{pwp}}$, the minimum volumetric content to maintain turf health. Any moisture within the soil adjacent to the boundary will lose or gain moisture in the direction of the boundary subject to gravity or suction effects.

In addition to those boundary conditions, we must also provide an initial state of the system as a starting configuration. As we irrigate once daily, it is assumed that there is ample time for all surface water to be absorbed into the soil. As such, the initial condition for surface moisture is set to zero. The soil moisture content, however, must be measured from the soil to provide an accurate snapshot of the moisture distribution prior to irrigation. To provide this, current data from each sensing node is referenced and fed into the model. As the sensors are coarsely distributed spatially throughout the area, the data is upsampled to the same granularity as the optimization problem using a bilinear interpolation.

4.4 Fluid Flow Model Simplification

We simplify the model in two ways for reasons of computational efficiency. First, we linearize the model PDEs. Although this is not necessary if one only wants to solve the PDEs to obtain the flow over time given a schedule, it considerably facilitates the numerical optimization over the schedule. Nonlinear equality constraints make the feasible set nonconvex and give rise to local optima, which complicate finding a good optimum. They also require nonlinear optimization, which is slower. Second, we discretize the spatial and temporal domains and approximate the derivatives using finite differences. With these simplifications, since the objective function and inequalities are already linear in our application, the resulting optimization problem is a linear program for which efficient solvers that can handle millions of variables and constraints are available. As seen later, this allows us to obtain a valid schedule in a relatively small amount of time.

4.4.1 Fluid Flow Model Linearization. The goal of the linearization is to characterize each equation in the model in terms of linear combinations of optimization variables. To remove non-linearities arising from optimization variables multiplied by each other, we make reasonable assumptions about the behavior of the system to substitute these non-linearities with linear counterparts.

We break each optimization variable into a base value, with subindex 0, and a small deviation, denoted with a hat. For example, the volumetric moisture content, θ, is rewritten in the form $\theta = \theta_0 + \hat{\theta}$. Each occurrence of the original four optimization variables is replaced with a similar representation and simplified to achieve the following four linear equations, where we define a function $\varphi(\theta) = K(\theta)(L + \psi'(\theta))$ to simplify notation:

$$\frac{\partial h}{\partial t} = -\nabla \cdot (\underline{\hat{h}\hat{\vec{v}}} + \hat{h}\vec{v}_0 + h_0\hat{\vec{v}} + \underline{h_0\vec{v}_0}) + F_s - \eta\Big(h_0K(\theta_0) + h_0K'(\theta_0)\hat{\theta} + \hat{h}K(\theta_0) + \underline{\hat{h}K'(\theta_0)\hat{\theta}}\Big), \tag{6}$$

$$\frac{\partial \theta}{\partial t} = -\nabla \cdot (\underline{\hat{\theta}\hat{\vec{u}}} + \hat{\theta}\vec{u}_0 + \theta_0\hat{\vec{u}} + \underline{\theta_0\vec{u}_0}) + \zeta\Big(h_0K(\theta_0) + h_0K'(\theta_0)\hat{\theta} + \hat{h}K(\theta_0) + \underline{\hat{h}K'(\theta_0)\hat{\theta}}\Big), \tag{7}$$

$$\begin{aligned}\hat{\vec{u}} = {} & -K(\theta_0)\nabla\hat{h} - K(\theta_0)\nabla h_0 - K'(\theta_0)\hat{\theta}\nabla h_0 - \underline{K'(\theta_0)\hat{\theta}\nabla\hat{h}} + \frac{K(\theta)\vec{\tau}}{\rho g} - \varphi(\theta_0)\nabla\theta_0 \\ & -\varphi(\theta_0)\nabla\hat{\theta} - \varphi'(\theta_0)\hat{\theta}\nabla\theta_0 - \underline{\varphi'(\theta_0)\hat{\theta}\nabla\hat{\theta}} - \vec{u}_0,\end{aligned} \tag{8}$$

$$\hat{\vec{v}} = -\alpha_h\nabla h + \frac{\kappa_g}{\eta}\vec{\tau} - \vec{v}_0. \tag{9}$$

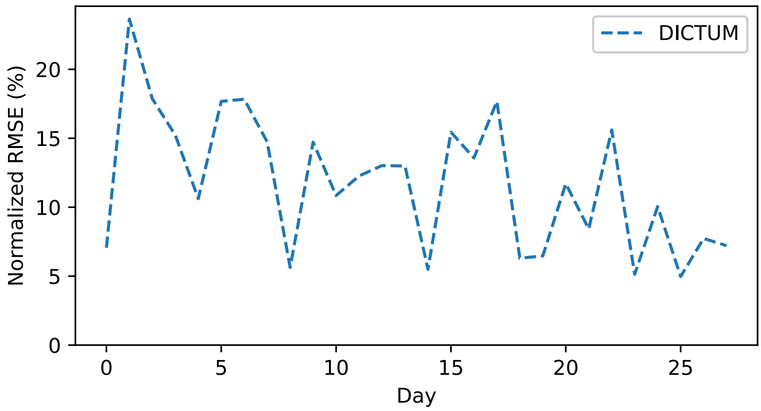


Fig. 4. Accumulated model error across irrigation period for each day of experiment.

As we assume that the deviations are small relative to the base value, the underlined terms that represent the nonlinearities of the original formulation should be much smaller than other terms appearing in the equations and are ignored. As $h_0\vec{v}_0$ in Equation (6) and $\theta_0\vec{u}_0$ in Equation (7) do not change spatially, the divergence of these terms is zero as well, so they are omitted.

4.4.2 Fluid Flow Model Discretization. The derivatives appearing in our modeling equations are approximated with finite differences. We use forward differences for the time derivatives for θ and h; for example, $d\theta/dt \approx \frac{\theta_{i,j,t+1}-\theta_{i,j,t}}{\Delta t}$, where Δt is the time interval size and t is the temporal index, with $t = 0, \ldots, N_t$. We use centered differences for the spatial derivatives for θ and h; for example, $d\theta/dx \approx \frac{\theta_{i+1,j,t}-\theta_{i-1,j,t}}{2\Delta x}$, where Δx is the spatial grid size and i, j are the indices of a spatial cell for $i = 0, \ldots, N_x$ and $j = 0, \ldots, N_y$. Consider the discretization on a continuous range $[0, L_x]$, where length $L_x = N_x * \Delta x$ in the x direction. The entire range is broken into N_x segments of length Δx. This discretization is performed on all variables of our linear model equations, resulting in the six equations found in Appendix B.

4.5 Model Accuracy

Across four weeks of DICTUM system deployment, we track the evolution of soil moisture in the system in response to the valve actuations chosen by the model-based optimization of the DICTUM system. As later described in Section 5, each day an optimization problem is solved that finds valve schedules that will minimize system water consumption while maintaining adequate moisture levels everywhere in the space. In addition to the optimized valve schedules, this optimization also produces the predicted sequence of soil moisture levels that will occur during irrigation using the PDE model. To determine the accuracy of the PDE model, each day, we compare the final moisture level *predicted* by the model in optimization to the *true* final moisture levels experienced across the space after irrigation following the optimized schedules. Each day, we track the root mean square error (RMSE) between the predicted and true final moisture values and normalize it across the range of moisture levels experienced in this dataset, a range of 17% volumetric content in this particular experiment. These errors, tracked each day, can be seen in Figure 4, where we can see the error in the first days of the deployment can reach as high as 25%. This is in part due to the coarseness of the initial moisture distribution measurements. Moreover, as we show in Section 7, the first days of an experiment tend to be where the control strategy must correct for poor initial moisture conditions of the space, and in this case, we suspect that these atypical initial conditions tend to reduce the accuracy of the model. This is supported by the fact that as the system runs and

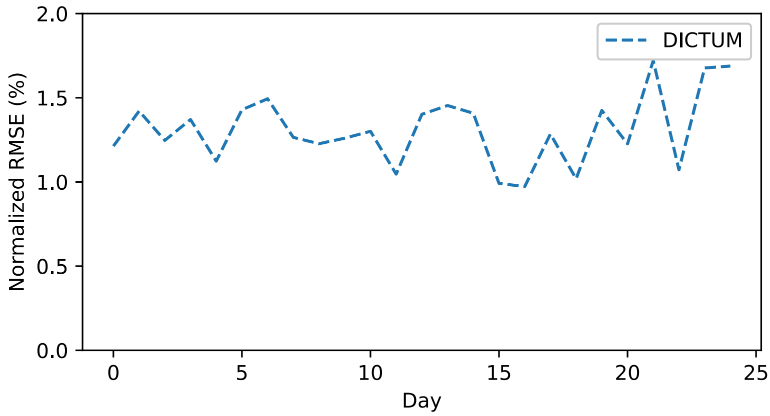


Fig. 5. Average model error that accumulates over a 60-second window in each day of experiment.

guides moisture conditions into a more typical range, the daily model error reduces to the order of ~10% in the last several days of Figure 4.

In its current operation, the DICTUM system is used once at irrigation time to optimize valve schedules for the day's irrigation. However, if the optimization performance allows for it, it would be preferable for the system to also perform *intermediate* schedule optimization using freshly sensed moisture values across the space in the middle of irrigation, to correct for error in the PDE model. To determine the effect this re-correction would have on model accuracy on the same historical data as the previous error analysis, we consider a system that uses fresh data and re-computes optimization at *every* 60-second control timestep. Then, considering the same metric as before, we compute the average model error that is able to occur within this 60-second interval for each day in the experiment. Figure 5 shows that the average error tends to be between 1% and 2% of the true value, indicating that using re-correction in the DICTUM system, on average, the predicted moisture level will be no more than 2% away from the true moisture level in the space, a significant accuracy improvement in comparison to the 10%–25% error possible when no re-correction is done as shown in Figure 4.

As we discuss at length in Section 2, there are no existing models that have the features required for a model-based optimization in irrigation control; for optimization, we require a programmatic interface, the ability to model both sub-surface and surface fluid as well as the movement between the two, and relatively high performance to make schedule optimization tractable. Although the model used in the DICTUM system has some error in operation, it meets all the requirements for the system to function. Despite these imperfections, we later demonstrate in Section 7 that the DICTUM system is able to achieve significant system benefits in both efficiency and quality of service.

5 OPTIMIZATION OVER THE SCHEDULE

For use as an irrigation control system, we must now use our model from Section 4 to produce optimal sprinkler scheduling for the system. The objective of this optimization is to produce a schedule that provides enough moisture at all points in the space to maintain health while minimizing system water consumption.

Our optimization problem is the following linear program (LP) with variables defined in Table 2: The objective function is total water spent over a period N_t of time. We define $f_k(t)$ as a binary function that equals 0 if sprinkler k is off at time t and 1 otherwise, then the water spending is

Table 2. Optimization Variables

Variable	Description
i, j	Spatial index $\in \{0, \ldots, N_x\}, \{0, \ldots, N_y\}$
t	Temporal index $\in \{0, \ldots, N_t\}$
k	Sprinkler location index $\in \{1, \ldots, K\}$
f_{kt}	Sprinkler k actuation at time $t \in \{0, 1\}$
$h_{(ijt)}$	Height of water on surface
$\theta_{(ijt)}$	Soil volumetric water content
$\theta_{\text{initial}(ij)}$	Sensed soil moisture at time $t = 0$, upsampled to PDE grid $N_x \times N_Y$
$u^x_{(ijt)}$ $u^y_{(ijt)}$	Soil water velocity, x, y direction
$v^x_{(ijt)}$ $v^y_{(ijt)}$	Surface water velocity, x, y direction

proportional to $\sum_{k=1}^{K} \int_0^{N_t} f_k(t)dt$. Discretizing over time gives as objective function $\sum_{k,t=0}^{K,N_t} f_{kt}$, where f_{kt} are $K \times N_t$ optimization variables (the sprinklers' schedule).

We have as additional optimization variables the values of $h, \theta, u^x, u^y, v^x, v^y$, (6 variables) at each spatiotemporal cell, with a total of $N_t \times N_x \times N_y \times 6$ variables. So, the complete set of *optimization variables* is $\{f_{kt}, h_{ijt}, \theta_{ijt}, u^x_{ijt}, u^y_{ijt}, v^x_{ijt}, v^y_{ijt}\}_{i,j,t=0}^{N_x,N_y,N_t}$.

The equality constraints arise from the necessity of the joint values of these variables to satisfy the fluid flow PDEs everywhere in time and space. There are 6 PDEs, hence, we have 6 equality constraints for each spatiotemporal cell. They are given by the linearized, discretized PDEs of Appendix B. As each constraint involves only 4 variables, because of the spatial neighborhood relation induced by the finite differences, the matrix of equality constraints is sparse.

As discussed in Section 4.3, we initialize our soil fluid levels at time $t = 0$ with the most recent soil moisture levels sampled across the space with our distributed sensors, upsampled to the spatial discretization level of our PDE model with a bilinear interpolation. This upsampled soil moisture snapshot is defined in Table 2 as θ_{initial} and is used to constrain the soil moisture levels at time $t = 0$ as shown in Equation (10e). Similarly, we constrain the surface water height to be zero at all locations at the beginning of irrigation as shown in Equation (10f), as irrigated moisture will have been absorbed into the soil well before the next irrigation is to occur.

The inequality constraints define the goal state of the system; namely, a schedule must provide volumetric water content in the soil exceeding the minimum water content plants need, θ_{pwp}, and be within prescribed lower and upper limits θ_{l} and θ_{u}. These constraints are of the bound type, i.e., they have the form "variable ≤ constant" for each variable. The LP is defined as follows:

$$\min_{\{f_{kt}, h_{ijt}, \theta_{ijt}, u^x_{ijt}, u^y_{ijt}, v^x_{ijt}, v^y_{ijt}\}_{i,j,t=0}^{N_x,N_y,N_t}} \sum_{k=1}^{K} \sum_{t=0}^{N_t} f_{kt} \quad \text{s.t.} \tag{10a}$$

$$0 \le f_{kt} \le 1, \qquad k = 1, \ldots, K, \quad t = 0, \ldots, N_t, \tag{10b}$$

$$\theta_{\text{l}} \le \theta_{ijt} \le \theta_{\text{u}}, \qquad i = 0, \ldots, N_x, \quad j = 0, \ldots, N_y, \quad t = 0, \ldots, N_t, \tag{10c}$$

$$\theta_{\text{pwp}} \le \theta_{ijN_t}, \qquad i = 0, \ldots, N_x, \quad j = 0, \ldots, N_y, \tag{10d}$$

$$\theta_{ij0} = \theta_{\text{initial},\, ij}, \qquad i = 0, \ldots, N_x, \quad j = 0, \ldots, N_y, \tag{10e}$$

$$h_{ij0} = 0, \qquad i = 0, \ldots, N_x, \quad j = 0, \ldots, N_y. \tag{10f}$$

PDE model equations (12)–(17)

The PDE model equations are as calculated in Section 4.4.2 and can be found in Appendix B.

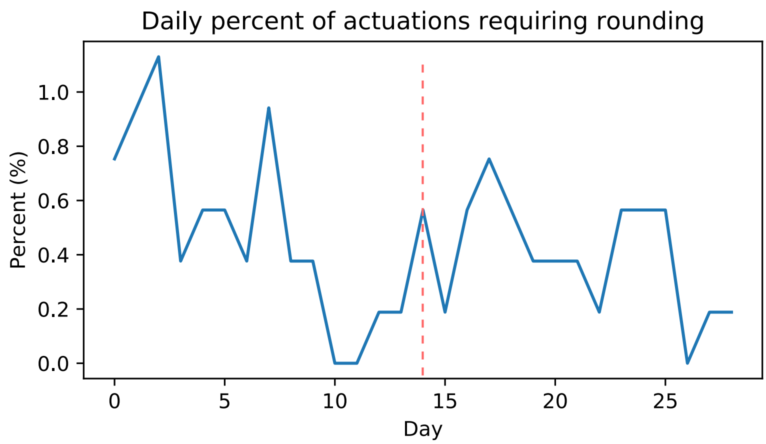


Fig. 6. Percent of actuations requiring rounding across deployment 1 (left of red line) and deployment 2 (right of red line).

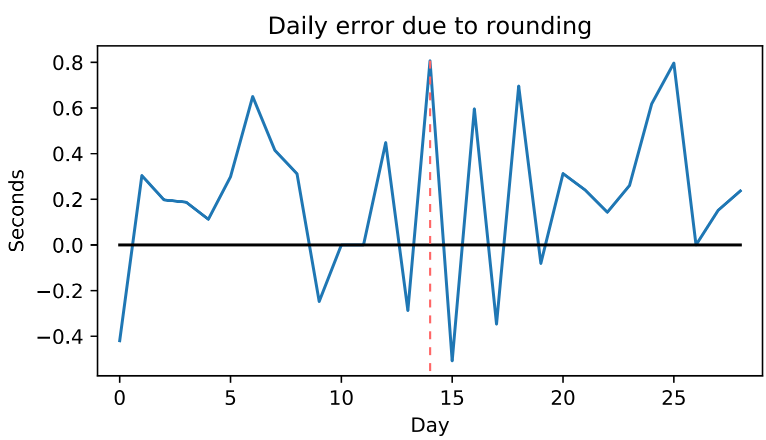


Fig. 7. Daily effect of rounding on sprinkler on-time across deployment 1 (left of red line) and deployment 2 (right of red line).

The optimization variables f_{kt} represent the binary actuation of a physical water valve, which makes the problem an integer linear program (ILP). ILPs are NP-complete and must be approximated in practice. Here, we simply relax them to the continuous LP by letting each variable be real in [0,1] and then rounding them to integer values for use in the physical system. Although other approaches exist that can give better approximations (such as branch-and-bound), they are impractical for the size of our problem. While this method will introduce some error, it is expected to be small relative to errors introduced due to the model simplifications discussed in Section 4.4. Furthermore, we found across our two deployments that the optimal schedules tend to leave the sprinklers on for long periods of time, with optimal values of f_{kt} on the 0/1 boundary. Figure 6 shows the percent of f_{kt} values that do NOT lie on the 0/1 boundary and thus require rounding; it can be seen that on the worst day, just over 1% of actuations require rounding. By then comparing the optimal schedules before and after their values are rounded, we can determine the effect this rounding has on the schedules used for irrigation. Figure 7 shows that on the worst day of our experiments, this rounding caused less than 1s of deviation in our schedules, which is insignificant,

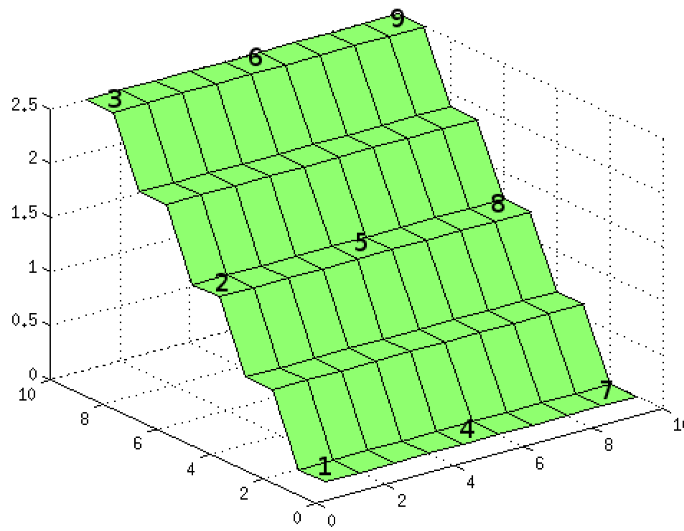


Fig. 8. Sensing and actuation node locations.

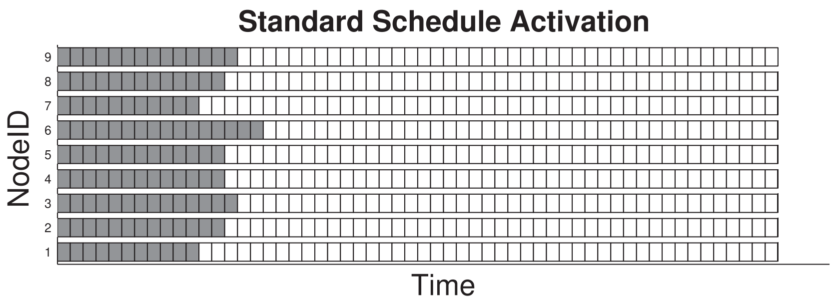
as irrigation takes nearly an hour each day. Interestingly, a higher percent of actuations require rounding in the first deployment but do not lead to an increased effect on the sprinkler on-time.

The discretization in space and time results in a large number of variables and constraints. For example, using a coarse spatial grid of 10×10 with 100 timesteps results in 10,000 cells, and so 60,000 variables (plus $100{\times}K$ schedule variables for K sprinklers), 60,000 equality constraints, and 10,000 inequality constraints. Fortunately, the equality constraints are sparse and the inequality constraints are simple bounds. In the initial use of this system [38], Stanford's CVX convex optimization library [26] was used to solve the LP due to its convenient programmatic interface. However, we quickly realized that its performance suffered for large optimization problems such as ours, taking on the order of an hour to find optimal schedules even at a very low spatial/temporal discretization level. Most of this run-time was spent setting up the constraints and variables. The optimization codebase was later ported to the Julia programming language [18] for its similar ease of use and much improved speed (both in setting up the data and in solving the actual LP). A performance comparison between these two tools can be found in Section 5.2.

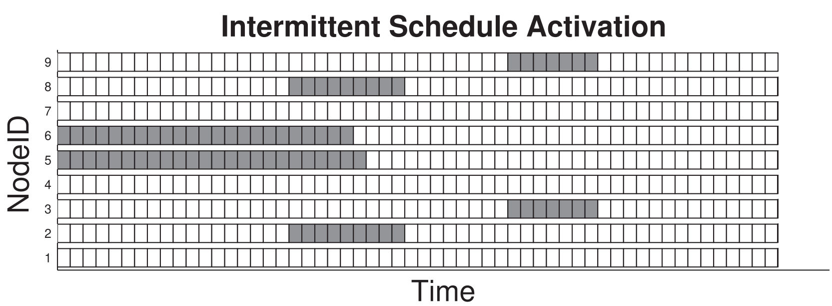
5.1 Proof-of-concept Simulation

As the model is integrated into the optimization framework, we can perform a proof-of-concept experiment to ensure optimization produces schedules that follow intuition. Figure 8 shows the topography used to test and node locations, made to resemble the hillside used in our case study, as shown in Figure 11. The hillside was modeled with soil characteristics as would be found in the deployment location, and the optimization was performed. The schedule produced can be seen in Figure 9(a), where the dark blocks correspond to active sprinklers as time progresses on the x axis. As this is an example problem, the time is purposefully unitless, as it does not represent a real scenario. This schedule can be observed to favor irrigation at the top of the hill and favor Nodes 1 and 7 least, as they are located in the bottom corners. This follows intuition, as any unused water at the top of the hill will move away as runoff and benefit the downhill turf.

To continue the example, the soil was adjusted to mimic a less-absorbent clay and reduce its thickness. The thin soil requires less water, but as the clay is less absorbent, watering all at once would cause most water to be lost as runoff. The schedule produced in response to this



(a) Optimized Schedule without Intermittency



(b) Optimized Schedule with Intermittency

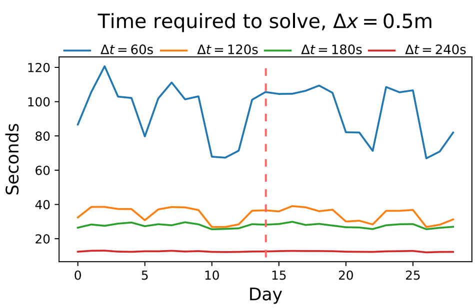
Fig. 9. Example of optimized schedules.

environmental change is shown in Figure 9(b), with the dark blocks corresponding to active sprinklers. The optimizer finds a solution that causes actuation to occur intermittently, making irrigation non-continuous. As the less-absorbent soil causes runoff to occur much more dramatically, the optimal solution prevents the lower sprinklers from actuating at all, allowing the runoff from above to provide adequate moisture to the region below.

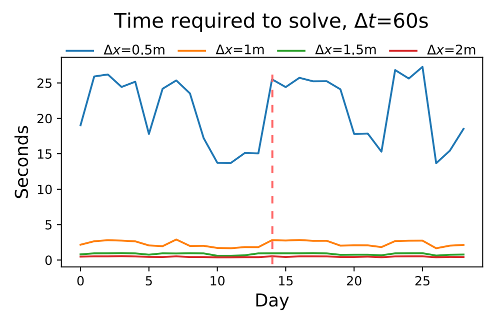
5.2 Effect of Model Granularity

As discussed in Section 4.4.2, we must select both a temporal and spatial discretization level for our discretized moisture movement model, which will affect both the tractability of the problem and the accuracy of the model. Although the use of the slower CVX during our physical deployments prevented a thorough analysis, with the initial moisture conditions of each day of our deployments, we can re-create the identical optimization problem that would have been run on each day of our deployment in the new Julia optimization framework and vary the discretization. The resulting schedule could then be run using the maximum spatial-temporal granularity model available as ground truth to see the effect of using a lower granularity model when solving the optimization problem and corresponding solution schedule.

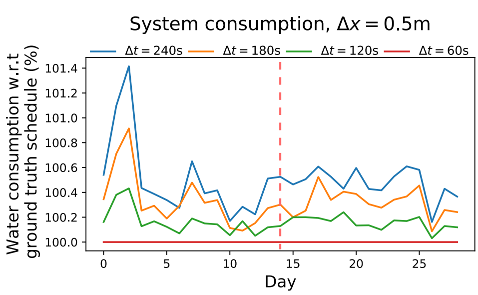
To consider the effect granularity has on performance, optimization is run for each day of the deployment using several temporal and spatial discretization levels, and the run-time is recorded as shown in Figures 10(a) and 10(b), respectively. When varying the spatial discretization, we fix the temporal discretization to a very fine level; and likewise when varying the temporal discretization, we fix spatial discretization to a very fine level. As we expect, a very fine discretization level in both time and space results in the highest run-time of just a couple of minutes, with reduced granularity allowing the problem to be solved much faster, as a rougher discretization reduces



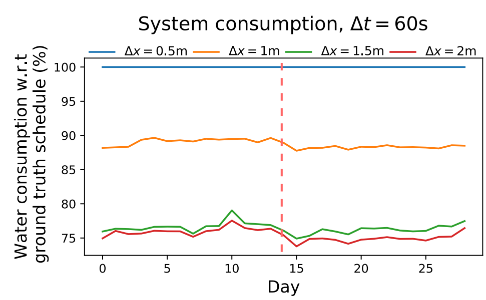
(a) Time required to optimize schedule as the model's temporal granularity is varied



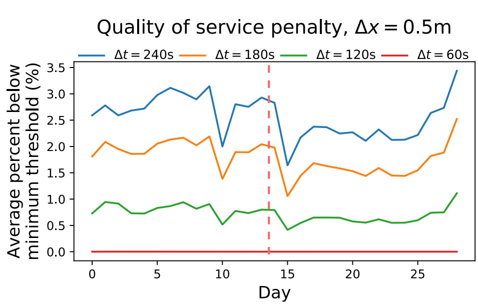
(b) Time required to optimize schedule as the model's spatial granularity is varied



(c) Simulated system water consumption as temporal discretization is varied



(d) Simulated system water consumption as spatial discretization is varied



(e) Degradation of quality of service as temporal discretization is varied

(f) Degradation of quality of service as spatial discretization is varied

Fig. 10. Discretization effects across first (left of red line) and second (right of red line) deployments. Lower is better for all metrics.

the number of optimization variables and constraints that must be evaluated by the optimizer. In addition, we consider the run-time of the CVX framework used in our deployments in comparison to the Julia framework; we find that at the standard spatial and temporal discretization level used in our deployments, CVX takes around 40mins to optimize the schedules. In comparison, we found Julia is able to solve the same optimization problem in just under 1s, a performance improvement of about 2400×. Using the Julia optimization framework, we found that we could solve optimization problems with significantly more constraints and optimization variables much faster, but as the problems grow very large, the limiting factor eventually becomes system memory on the machine running the optimization. For instance, considering optimization of our schedules with spatial granularity of .1m and temporal granularity of 15s, the number of optimization variables exceeds 28M, with a similar number of equality and inequality constraints. While this LP can be solved within tens of minutes using Julia, reasonable as optimization takes place only once per day, the Julia process solving this optimization reports that around 12G of system memory is required.

It is clear that using a model with a coarse granularity has performance benefits, but we also want to see how this sacrifice will affect model accuracy. The best way to determine this effect would be to vary the temporal/spatial granularities and re-run the experiments. However, as each experiment takes weeks, this would severely limit our ability to thoroughly search the full range of discretization intervals. Instead, for each day of experimentation, we solve the optimization at varying temporal and spatial granularities and then run the resulting schedule through our model at the highest granularity to act as a simulated *ground-truth model*. By comparing the computed schedule to one computed with the ground-truth model, we can see the effect it has on system efficiency; and by finding on average how much each simulated location falls beneath our desired moisture level at the end of irrigation, we can see its effect on quality of service.

The change in irrigation water consumption as the temporal discretization is varied can be seen in Figure 10(c), and the change in quality of service can be seen in Figure 10(e). We can see that as the temporal granularity is made more coarse, the resulting schedules consume slightly more water, but on average, we see that the quality of service slightly degrades as well. As the causes of water movement in the space occur at different timescales, the timing of water application to the soil can have a strong effect on the final distribution of soil moisture. This is an effect described in Section 5.1, where variations in environmental characteristics can make it necessary to utilize more advanced irrigation schedules—for instance, through intermittent actuation. However, as the model used in optimization becomes more temporally coarse, it becomes more difficult to accurately anticipate the timing requirements of irrigation and can result in a decreased quality of service despite using slightly more water. As neither increased water consumption or degraded quality of service is desired, it is clear that the finer the temporal granularity used to compute schedules, the better.

In comparison, varying the spatial granularity has a more profound effect; the simulated change in system efficiency can be seen in Figure 10(d), and quality of service can be seen in Figure 10(f). Reduction in spatial granularity removes spatial locations from the moisture movement model, meaning their moisture requirements are no longer considered when optimizing schedules. As shown in Figure 10(f), this results in a significantly decreased quality of service, with locations across the space missing their required moisture levels by an average of almost 10%. However, by underestimating water needs in the irrigation system, the system also requires less water to reach the perceived (although incorrect) moisture requirements. As Figure 10(d) shows, this leads to significant water savings in the system, with as much as 25% less water consumed with the roughest spatial granularity. A sacrifice in the model's spatial granularity causes a clear tradeoff between quality of service and system water consumption. It is difficult to estimate the long-term effects of this degraded quality of service on the health of the plant, and it is possible that very hardy grass species may survive, potentially resulting in water savings in the system. In general, however, maintaining the high quality of service is absolutely necessary in maintaining plant health, and this is achieved by using a model with the finest spatial granularity.

6 CASE STUDY: LIVE DEPLOYMENT

In many applications, it is possible to compare a newly developed model to other accepted models. However, as an all-inclusive model for our application does not exist, we compare to reality

Fig. 11. Deployment side-view.

by evaluating the performance of an irrigation system using the control modifications we have proposed.

We chose to deploy two systems side-by-side once a suitable location is found. Ideally, the two systems will cover similar soil and turf, face the same direction so sun exposure will be equivalent, and have completely independent irrigation to avoid cross-contamination. In addition, to ensure all sprinkler coverage is the same, the same water source is used to power both systems, and actuation is provided to both sides with the installation of our DICTUM nodes. The only difference between the systems are the control schedules sent to each side.

At the beginning of our project, we intended to use an existing irrigation system to perform our deployment. In looking for a suitable location, we came to realize that the granularity of irrigation control on our university's campus was less than ideal. Locations spanning more than 10K square feet across heterogeneous terrain were actuated by a single control valve. As such, it would not be feasible to show the benefits of higher-granularity actuation using the existing system, so we began planning a custom irrigation system.

6.1 Seeking a Suitable Location

With the help of university personnel, a suitable location was found on a stepped hillside (see Figure 11) far away from the nearest foot-traffic. The hill, rising about 9 feet over a distance of 70 feet, acted as an elevated surround for a university soccer field, the hill stretching more than 200 feet. To take advantage of this topography, it was decided to place two irrigation systems side-by-side, each spanning a 70'×70'' area along the hillside. Between the two systems were 5 feet of unirrigated space to prevent any spray from one system from entering the other side. As a whole, the two irrigation systems spanned an area of 70'×145'', approximately 10,150ft^2.

6.2 System Development and Deployment

The underground irrigation system used by groundskeepers to maintain the hillside we deployed on was plumbed with high-flow PVC water lines leading to each of their sprinklers. As it would be risky to tap into this permanent infrastructure to test our temporary system, we used lower-flow quick-couplers, fitted with standard hose spigots. We discovered that the nearest quick-coupler could provide enough water flow to activate one of our side-by-side irrigation systems, but not both simultaneously. Throughout our case study, irrigation of the two systems is performed one immediately after the other to avoid this issue.

To choose the type and number of sprinklers for our side-by-side deployment, we consulted sprinkler manufacturer specifications; a general rule of thumb for system planning is that the coverage of one sprinkler should reach 75%–100% of the distance to the next closest sprinkler to avoid uncovered areas on the diagonal. To cover one of the two proposed areas covering 70’×70’ with these recommendations, we would require four sprinklers in a 2×2 grid, each with a reach exceeding 52 feet, or 9 sprinklers in a 3×3 grid, each with a reach exceeding 26 feet. Sprinklers that can exceed 52 feet are generally of the “rotor” variety, pointing in only one direction and rotating slowly to cover the region. The flow required by these larger rotors is typically more than our quick-coupler water source could provide. It was found the the low-flow MP-Rotator 3000 sprinkler [5], a new rotor that is known to be remarkably efficient, could reach up to 30,’, 9 of which can be easily powered by our quick-coupler water source. As the MP-Rotator is quickly replacing older sprinkler technologies at our university for their slow-application and minimum runoff, we decided they would be the best choice for our deployment.

As the deployment was meant to be temporary, we designed it such that it could be removed in a reasonable amount of time. Although a commercialized version of DICTUM would be installed in the ground, our temporary system was placed on the surface for ease of access to our prototype. The nodes themselves were placed on the ground just next to the sprinklers, while the solenoid providing actuation was fixed to the sprinkler riser. By placing our system on the surface, we are able to avoid damage to the lawn that installing into the ground would cause.

Last, our deployment included a central basestation fitted with power, a small Sheeva Plug computer [12], an elevated 802.15.4 mote to receive data from the sensing nodes, and a 4G hotspot to allow us to communicate with the wireless sensor network from a remote location. Although our basestation was overbuilt to facilitate ease of debugging and close monitoring of the prototype system, a commercialized system may have only the mote to interface with the sensor network, and an interface to any external service (local, cloud, etc.).

6.3 Node Development

In our first deployment, our device was fitted with an ordinary solenoid designed to regulate the flow of high-pressure water. This solenoid was found to work very reliably, and as it defaults to an off state when power is cut, it is also a safe choice. However, its weakness was in its high power requirement, as it must pull a constant 350mA to hold the valve open. Later discussed in Section 7.4, this dominates the power profile of the device and caused the node lifetime in our first deployment to fall under one week. Additionally, as the power supply in our first hardware version was unregulated, the collected sensor data would begin to sag at the end of battery life, as the sensor’s supply voltage would drop. We mitigated these weaknesses during our first deployment with frequent battery changes to prevent this sensor sag from occurring.

These problems are addressed in the hardware version used for our second deployment, where we chose to use a latching solenoid for sprinkler actuation to extend system lifetime. Whereas a normal solenoid requires constant power to stay in an open position, a latching solenoid requires only a 50ms pulse of either positive or negative voltage to switch between an open or closed state. One advantage to a non-latching solenoid is the inherent safety; a loss of power in the control board will automatically deactivate water flow. To provide this safety in a latching solenoid, a simple circuit can be built into the control board that will close the solenoid if the board fails, as shown in Figure 12. A pin is maintained on the microcontroller that changes from 0 to 1 in event of failure (or from 1 to 0; a NOT gate can be used to invert it). Once this pin is active, it immediately brings the top pin of an XOR logic gate to 1, allowing current to flow through while the capacitor on the bottom pin charges. Once the capacitor charges, the second pin of the XOR logic gate activates, terminating the current. This pulse, generated while the capacitor charges, is

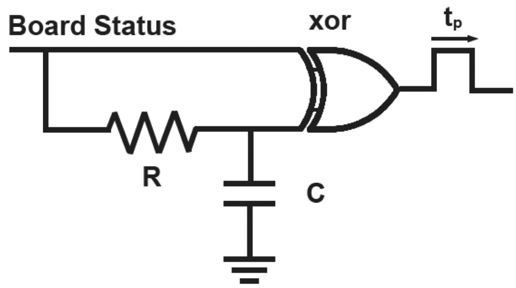


Fig. 12. Example of safety mechanism for latching solenoids.

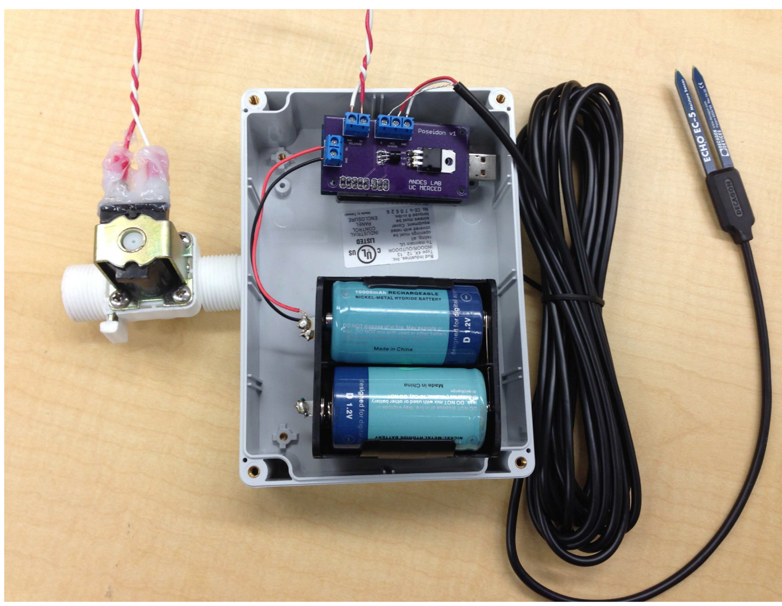


Fig. 13. Sensing/actuation (DICTUM) node.

for a period $t_p \propto CR$, where C and R are the capacitance and resistance values used in the circuit. This pulse will deactivate the solenoid, preventing additional water consumption after node failure. With this configuration, the control node will benefit from the extremely low-power operation of a latching solenoid without sacrificing safety.

The main challenge in the development of the DICTUM node was the organization of multiple input/output connections from the mote. To address this, we manufactured custom printed circuit boards that would organize the connections from the Tmote Sky. This interface board, shown with the rest of our prototype device in Figure 13, has connections for battery power, sensor (right), and solenoid (left). The different voltages required by the mote and solenoid were provided by voltage regulators built into the board. As the chosen latching solenoid requires the board to produce a positive (opening) and negative (closing) voltage for operation, the board was equipped with an h-bridge. Commonly used in robotics to drive a motor forwards and backwards from a single DC power source, an h-bridge is designed for applications like ours that require bi-directional current.

The other key feature of the DICTUM node is the ability to measure the volumetric water content in the surrounding soil. We opted to purchase research-quality Decagon EC-5 [2] sensors,

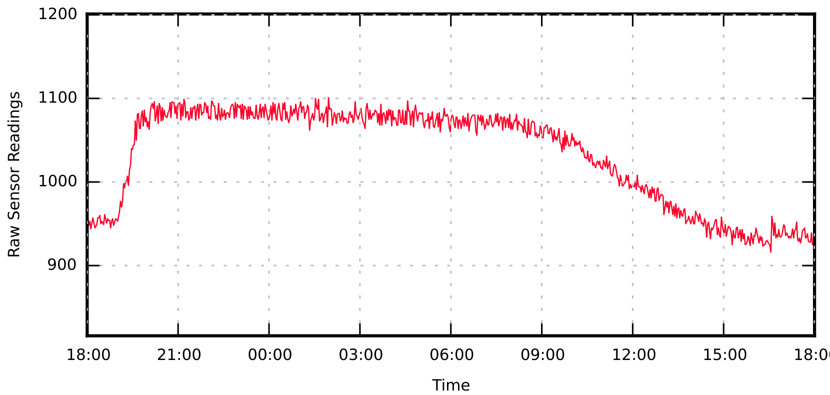


Fig. 14. Daily soil moisture cycle.

with a reported accuracy of ±3%. The Decagon sensors cost roughly $110, but data fidelity in our model verification was paramount. Raw sensor readings collected over a period of one day with a high sampling frequency can be seen in Figure 14. The sensors report the dielectric constant of the soil, an electrical property highly dependent on the volumetric water content. A linear calibration function provided by the sensor manufacturer is used to convert the raw readings to volumetric water content. An attentive reader will notice the sharp increase in the late afternoon due to irrigation, stable readings throughout the night, with sensor fall beginning as the sun comes up at 7:45am. This decrease steepens as the sun rises and faces the deployment directly after 10am. In a sensor-dense environment, a more economical alternative to the EC-5 may be chosen, such as the $35 Watermark sensor [35], which can be calibrated to ±5% accuracy.

6.4 System Comparison

In this study, we compare the operation of the DICTUM system against two baseline control systems. The first baseline, evaluated over two weeks of fine-grained data collection, employs a trial-and-error control strategy used widely in practice (including our campus). In this technique, a greenskeeper will monitor an irrigation system for days or weeks; if an excess of runoff provides evidence of over-watering, or if brown patches provide evidence of under-watering, they will adjust the system accordingly. This irrigation scheduling, often remaining unchanged through entire seasons, leads to a misuse of water, as it does not account for changing weather or soil requirements. We emulate this strategy by matching exactly the amount of water coverage as would be provided by the greenskeepers of our campus.

The second baseline control strategy, evaluated in a two-week deployment, employs a state-of-the-art evapotranspiration (ET) control strategy. As described in Section 2, these systems use weather forecasting to estimate the amount of water lost by the soil due to evaporation and plant transpiration. Irrigation controllers that use ET technology typically irrigate every 1–3 days, replacing water lost over that period. To emulate an ET system, we query a local weather station that computes hourly ET loss and compute the previous day's water losses. With our sprinklers' surface coverage rate, we create daily valve schedules to do exact replacement of these losses.

7 EXPERIMENTAL RESULTS

Over four weeks of fine-grained data collection, we ran two irrigation systems side-by-side on identical patches of turf, periodically collecting soil moisture data from each. In the first deployment, our campus' control strategy was tested and for the second deployment, state-of-the-art evapotranspiration (ET) control was used. In both deployments, these systems were compared against schedules computed by our model-based optimization, actuated using our custom-made independent actuation/sensing platform (DICTUM) nodes. The goal of these case studies was to determine if a system could be made that reduced the amount of water used while maintaining a satisfactory level of moisture in the soil for the turf to remain healthy.

7.1 Quality of Service

The primary objective of an irrigation system is to maintain plant health. A very efficient system that is unable to meet this objective will be replaced with a less-efficient system that provides satisfactory water coverage to the turf. To remain healthy, turf needs proper soil nutrients in the appropriate amounts, adequate solar irradiation to power biological processes, and an adequate level of moisture in the soil near the plant roots. Although the irrigation system has no control over solar exposure or soil nutrients, we have direct control of soil moisture through actuation of the sprinklers. In plant physiology, the level of soil moisture at which plants can no longer extract water from the soil is known as the permanent wilting point (θ_{pwp}) [13, 30, 31]. Long-term exposure to soil with moisture below this level will cause the turf to wilt and die, so we aim to minimize the amount of time spent beneath this threshold, as it will ensure a better opportunity for the plant to thrive.

The permanent wilting point (θ_{pwp}) for loamy soils like that found in our deployment is typically between 10% and 15% [9], so we assume the worst case and assign θ_{pwp} to be 15%. We expect that if DICTUM were to distribute moisture in a smarter way by targeting areas that would otherwise receive inadequate water, the DICTUM sensor readings will spend less time underneath the θ_{pwp} threshold than the compared baseline. The sensor data collected by the distributed DICTUM nodes from the collection period and the minimum healthy saturation θ_{pwp}=.15 (black line) are shown for our first deployment against the Campus control strategy and our second deployment against the ET control strategy in Figures 15 and 16, respectively.

We can see that in both deployments, sensors in the DICTUM system spend significantly less time beneath the minimum moisture threshold than the compared baseline. Interestingly, we can see that seasonal rain was experienced once during each deployment, on day 10 in Figure 15 and on day 11 in Figure 16. In both cases, this resulted in a steep increase in all moisture sensor values that continued into the following two days. In our first deployment, the Campus and DICTUM control strategies spend a total of 56.9h and 16.3h beneath θ_{pwp}, respectively, indicating a 3.5× improvement in quality of service. Similarly, in our second deployment, the evapotranspiration system spends a combined 68.1h beneath the θ_{pwp} across the entire deployment, over 4× more than the 16.7h experienced by the DICTUM system.

We can see in both side-by-side deployments that there were occasional periods of data loss, on days 2, 6, 7, and 9 in Figure 15 and days 2 and 11 in Figure 16. Our system included a basestation Sheeva Plug [12] computer that acted as a bridge between the public internet and our wireless sensor network, with data forwarded over a USB connection to an attached Telosb. Occasionally, this USB connection would fail at night, resulting in corrupted data transmission between the two devices; we noted that this particularly occurred when the basestation was exposed to cold ambient temperatures at night, but further investigation is required to determine the exact cause of this failure. Important to note is that as the data connection would re-establish itself when conditions improved and a snapshot of data is only required as initial conditions to optimization once per day, these lapses in data had no effect on the operation of the DICTUM system.

Although the amount of time spent beneath the minimum moisture threshold gives a good indication of quality of service, it signifies that all time spent beneath the threshold is equal. In reality, the further under the minimum moisture threshold, the worse the health of the plant will

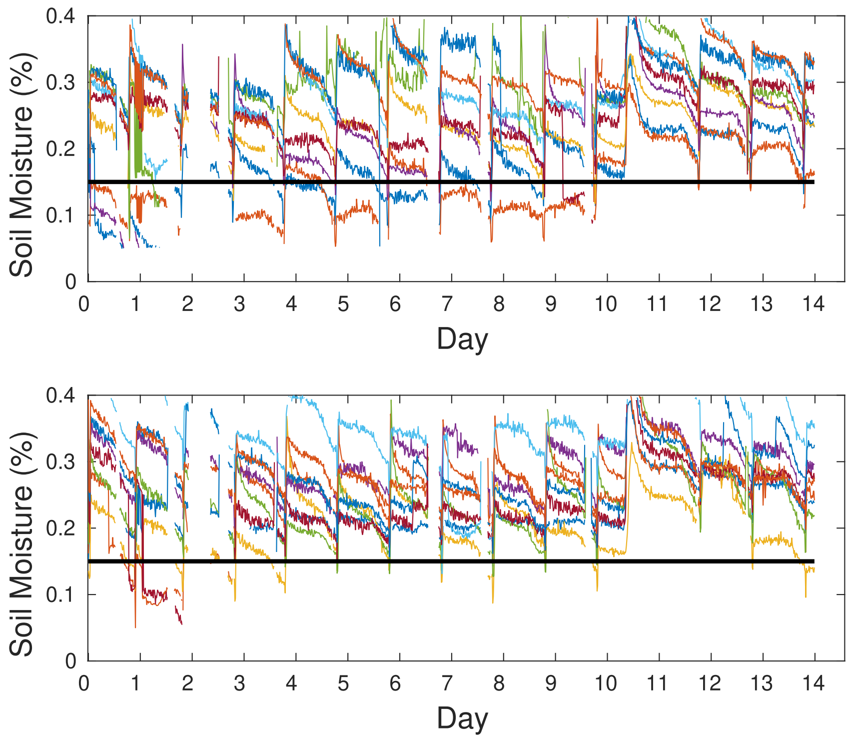


Fig. 15. Deployment sensor trends for all Campus (top) and DICTUM (bottom) systems, as collected by the installed DICTUM nodes.

become. To take this into account, we consider the sum squared amount of time spent beneath θ_{pwp}. In this way, the system will be punished more the further below the minimum moisture threshold. The sum squared time beneath θ_{pwp} can be seen across our first deployment in Figure 17, where the Campus control strategy is significantly worse than the DICTUM system until day 11, when the seasonal rain pushes the soil moisture levels into the acceptable range. In total, the Campus control strategy is 8.17× worse than the DICTUM system in this metric. Likewise, we evaluate our second deployment as shown in Figure 18. On day 3, we can see that the DICTUM system quality of service dips temporarily, but on all other days provides significant improvement in comparison to the evapotranspiration system. We can also see on day 12 that the rain on the previous day pushes the soil moisture levels to acceptable levels across the space. Across this deployment, the evapotranspiration system is 6.28× worse than the DICTUM system with respect to this metric. Although these analyses show that DICTUM is not perfect, it also demonstrates that more precise watering strategies can provide a significantly improved quality of service, despite using less water, as we will see in the following section.

7.2 Water Consumption Analysis

Throughout the deployment, the total on-time for each sprinkler was recorded in both systems. With knowledge of each sprinkler's distribution angle and the regulated water pressure, we can accurately estimate the amount of water consumed under any given schedule.

In comparison to the campus irrigation system, DICTUM is consistently more efficient, as shown in Figure 19, due to the campus irrigation system using a fixed schedule. We can see, however, that

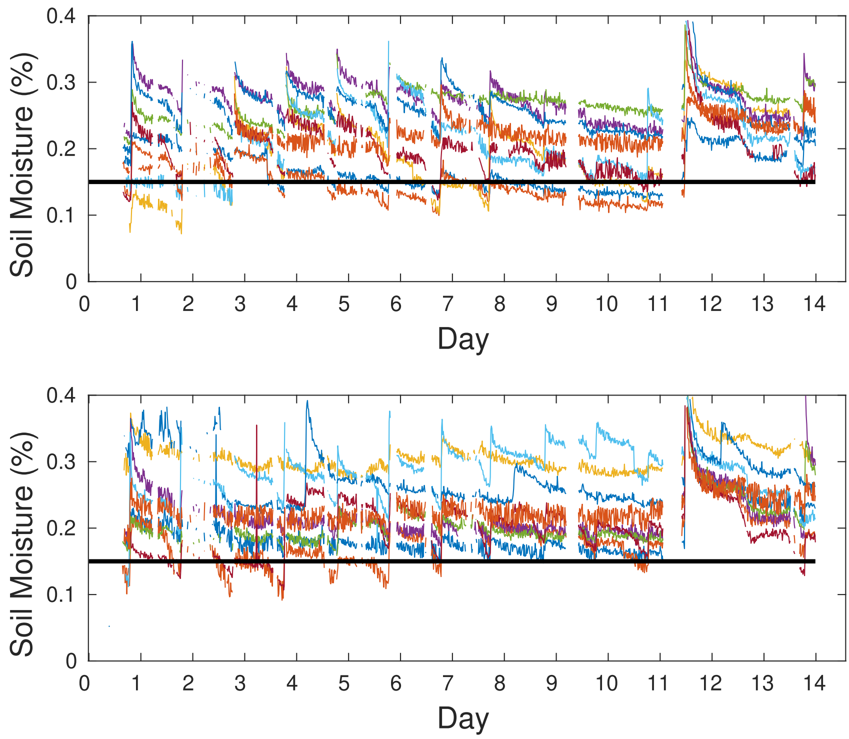


Fig. 16. Deployment sensor trends for all Evapotranspiration (top) and DICTUM (bottom) systems, as collected by the installed DICTUM nodes.

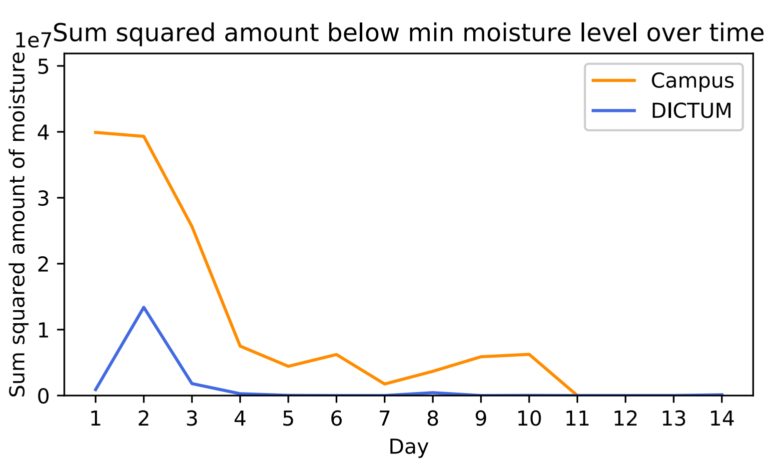


Fig. 17. Daily quality of service vs. Campus strategy (lower is better).

as our campus irrigation system is equipped with a rain detection sensor, the 11th day of irrigation is completely disabled in response to that day's precipitation. Irrigation is also disabled here in the DICTUM system, but this is in response to significantly higher soil moisture levels removing the need for irrigation. We can see that whereas the campus strategy continues providing full irrigation

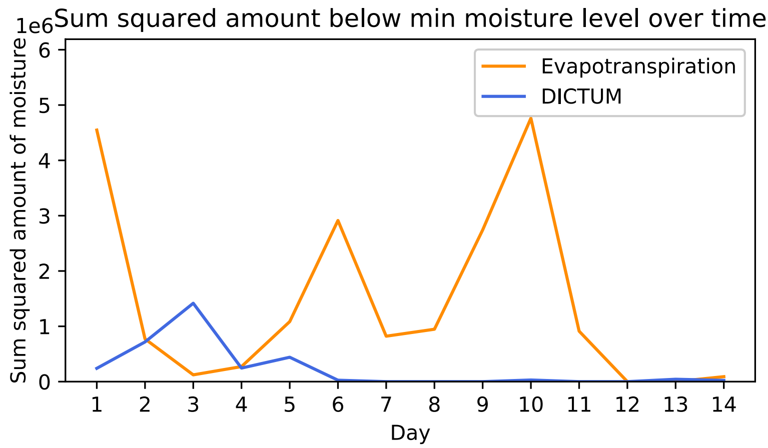


Fig. 18. Daily quality of service vs. evapotranspiration strategy (lower is better).

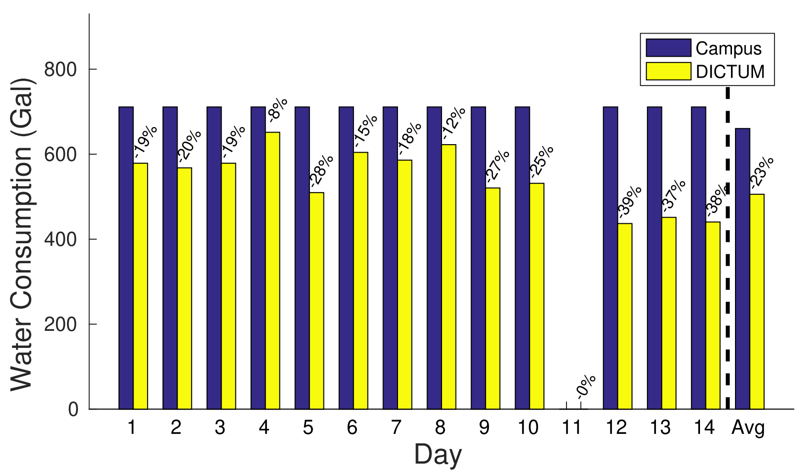


Fig. 19. Daily water consumption of Campus and DICTUM systems.

on the following days, the DICTUM system is able to significantly reduce the amount of irrigation in response to continued elevation in soil moisture levels for the last 3 days of system deployment.

A side-by-side comparison can also be seen for the deployment against the evapotranspiration controller in Figure 20. The variation of water consumption of the evapotranspiration system indicates changes in local weather. Low water consumption of both systems on days 9, 10, and 11 is due to cooler local weather, where the ET side consumed significantly less water. However, we can see in Figures 16 and 18 that these days of reduced water consumption also result in the worst days of quality of service, meaning the ET controller was too aggressive in its water savings on these days. Interestingly, a similar effect can be seen on day 6; the ET controller reduces water use in response to weather conditions, but at the cost of a reduction in quality of service, as shown in Figure 18. Rain on the 12th day caused both systems to cease irrigation until two days later, when the soil became dry enough to require it.

Across the two deployments, the DICTUM system consumed 12.3% less water than the evapotranspiration strategy, and 23.4% less in comparison to our campus' control strategy. Schedules

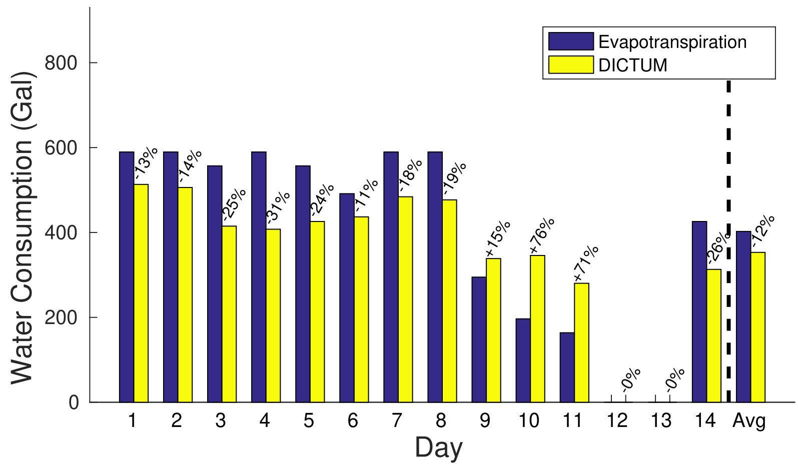


Fig. 20. Daily water consumption of evapotranspiration and DICTUM systems.

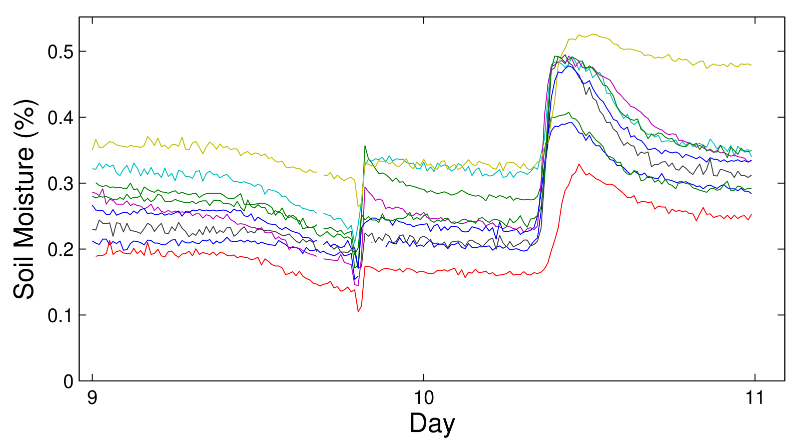


Fig. 21. DICTUM sensor readings through rain.

created by the DICTUM system were shown to consume as much as 651 gallons and as little as 280 gallons, a 371-gallon variation in response to the state of soil moisture. The reason our system is able to save water while providing a higher quality of service to the space is due to our ability to pinpoint regions within the irrigated space with varying moisture requirements. Using our model-driven approach, we can optimize actuation to send more water to areas that would otherwise receive insufficient moisture, while sending less water to the areas that would otherwise be over-watered.

Sensor readings through rain during our first deployment can be seen in Figure 21. To respond to such weather events, the weather stations used for evapotranspiration monitoring are generally equipped with a precipitation sensor. These sensors deactivate the irrigation system when rain is first experienced [6] or when a measured amount of rain has fallen. In the case that the rain is sufficient to irrigate the turf, this is satisfactory, but if the rain is very light, DICTUM might choose to provide additional irrigation.

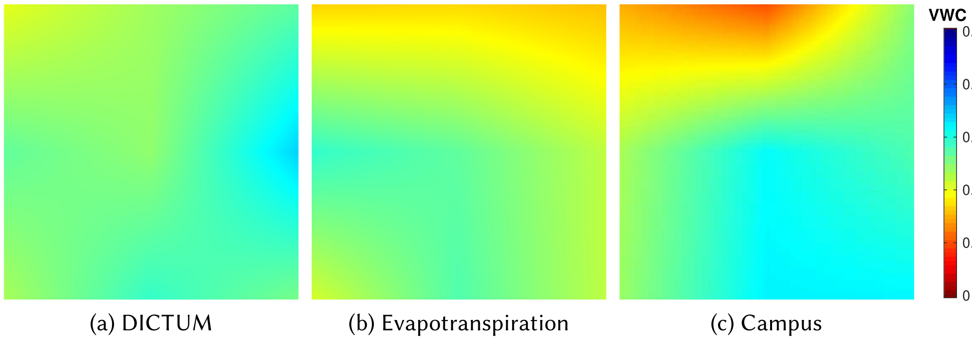


Fig. 22. Average moisture coverage of compared systems.

7.3 Moisture Uniformity

An interesting pattern recognized throughout the two deployments was the emergence of moisture uniformity. The optimization performed by DICTUM will create schedules that compensate for any areas of high or low moisture levels using soil moisture and topographical information. However, due to lack of moisture information and the physical limitations of the irrigation system (inability to actuate individual sprinklers), the ET and our campus' systems are unable to correct for uneven moisture. These artifacts can be seen in Figure 22, which shows a spatially interpolated view of the average moisture level of the three considered systems across our two deployments.

In this figure, the top of the image aligns with the uphill region depicted in Figure 8. We can see that as the irrigated water tends to flow downhill, the Campus and ET systems end up with inadequate soil moisture at the top of the hill; and in the Campus system, we can see extra water building up at the bottom of the hill. In comparison, the topographical effects are accounted for in the schedule creation of the DICTUM system, as more uniform moisture can be seen across the space with the exception of a small spot on the right-hand side where a bit of extra water can be seen to accumulate. Although variation in seasonal weather patterns between these two separate deployments makes a true side-by-side comparison difficult to make, these results demonstrate DICTUM's ability to produce schedules that correct for heterogeneities to provide homogeneous water coverage by taking advantage of distributed actuation.

7.4 Energy Consumption Analysis

From a wireless sensor network standpoint, the ability of a system to operate for a long period of time without user intervention is fundamental. Irrigation control devices are no different, especially if they are meant to be buried in the ground. The first version of the devices used a non-latching solenoid, which requires constant power to allow water to flow at irrigation time, dominating the power profile of the device as shown in Figure 23. As a solenoid can be active for as much as an hour each day for irrigation, we found that, during our first deployment, we had to constantly change batteries, once or twice per week. Our second version greatly improved with the use of a latching solenoid, which required only a short pulse of power to throw the valve between the off and on positions. This allows the device to be truly low-power, as this on-off cycle will only occur a few times per day in the worst case, with a clearly improved power profile, as shown in Figure 24.

For additional energy savings, the radio in each node is duty-cycled, activating for only a 10s period every 10mins. It was computed that using this duty-cycle, the 4 D cell batteries providing power to our nodes could run for over two years without requiring change. However, we note that

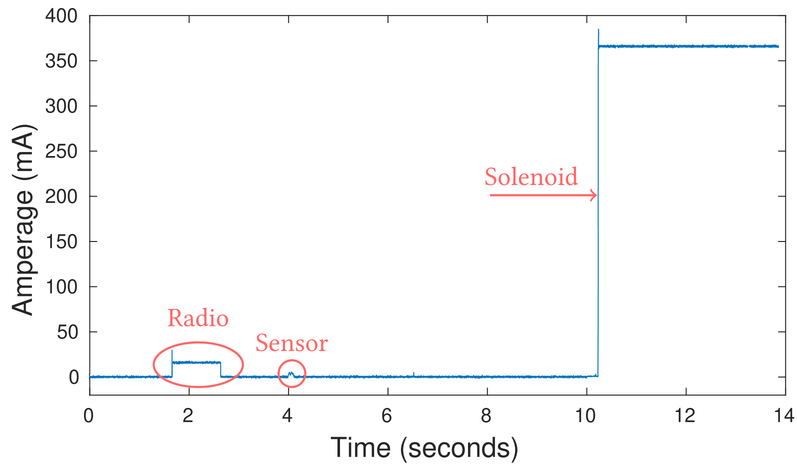


Fig. 23. Example energy trace of the sensing/actuation node with non-latching solenoid as used in the first deployment. Note that the high power consumption associated with solenoid activation will continue until the sprinkler is deactivated, potentially as long as an hour into the future.

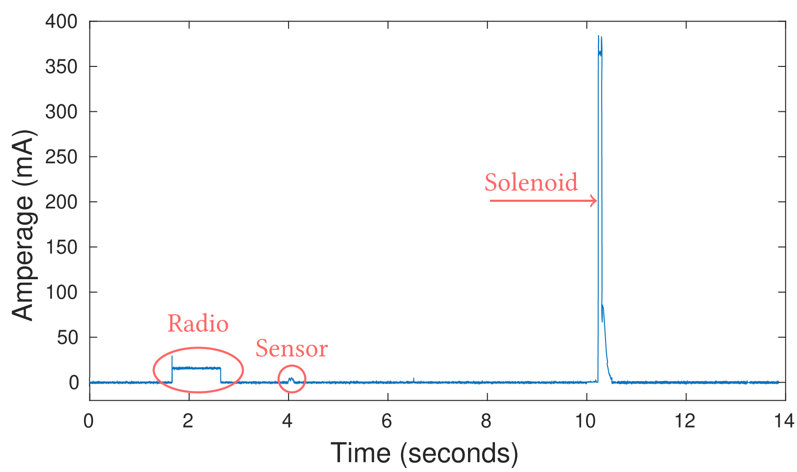


Fig. 24. Example energy trace of the sensing/actuation node with latching solenoid.

collected data is only required in our processing pipeline just before irrigation each day as initial conditions to optimization. In practice, a much more energy-efficient solution would be to continue sampling the onboard sensors but to store all this data on the mote's flash storage and send the entire dataset in batch just before irrigation. As these data samples are small, a day's data can be easily sent within a minute of radio on-time each day, allowing our prototype irrigation system to run uninterrupted in excess of 14 years while still performing its daily irrigation and data collection. The power effect of these peripheral devices on our latest prototype can be seen in Figure 24, where radio, solenoid, and sensor power consumption is shown over background CPU usage.

8 RETURN ON INVESTMENT ANALYSIS

Installing a system such as ours has many health, political, and environmental benefits that are difficult to quantify, but the economic benefits are crucial when deciding whether or not to install

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

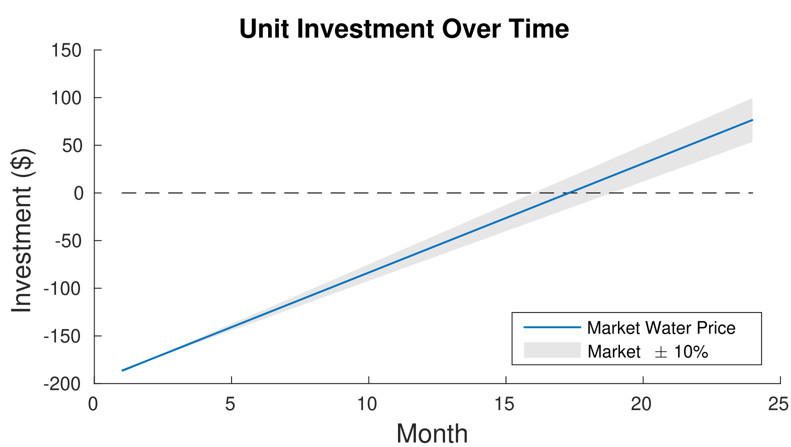


Fig. 25. Return of Investment timeline with varying water pricing.

such a system. In particular, we must consider 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 take into account the initial cost of the replacement system and the monetary savings expected from the increased efficiency of the replacement system.

Here, we consider the cost to develop a single DICTUM node in bulk for return on investment analysis. The primary components can be readily found: the sensor, solenoid, batteries, and waterproof enclosure are all possible to purchase from other manufacturers. In our prototype, the communication module used was a Tmote Sky [14]. However, as our application requires very specific circuitry to provide power to the various modules, commercialization of the DICTUM node would involve the manufacture of a stripped-down communication module, with the inclusion of the additional components described in Section 6.3. The pricing of a bare-bones Tmote Sky replacement, as well as the other various DICTUM components, can be found in Table 3.

To evaluate the expected return of investment, we computed the cost of the system and calculated the net investment as time progresses. The factor that most influences system pay-back is price of water. As this value is constantly changing, we perform the analysis using the current price for our campus, $5.60 per thousand gallons, and incorporate a deviation of 10% to account for changing water-market prices. Although the unit itself carries a high initial cost, return of investment can be expected to occur in 16–18 months, as shown in Figure 25. Considering the immense political pressure for irrigation water to be more expensive [3, 11], it is a good bet that the savings from an irrigation system such as ours will be on the rise in the near future.

It is difficult to directly extend the savings seen in our prototype to all irrigated space on a university's campus due to the heterogeneity of the installed system architectures. However, with slight modifications, the independent actuation control platform can be easily extended to control sprinklers of any type, delivering site-specific actuation for small-large scale systems. For example, on a campus such as ours, the majority of irrigated spaces use rotor sprinklers. As the rotors use substantially more water, independent actuation could provide an even greater positive environmental and financial impact, to be investigated in future work.

9 LIMITATIONS AND FUTURE WORK

To simplify the placement of irrigation infrastructure, sprinklers are almost always installed in a grid pattern. However, it might be possible to use the developed model to compute ideal sprinkler locations to compensate for natural topological characteristics for a system that has not yet been installed. In future work, we hope to evaluate the potential for water savings by also allowing variation in sprinkler positioning.

For DICTUM to estimate water movement, it must understand the spatial characteristics of the irrigated turf. This includes the topography of the terrain and estimates of soil type and depth. Topography measurement of advanced terrain can be performed by various imaging methods by satellite, drone, or other emerging technologies, and soil characteristics may be estimated by the system installer, but they remain a burden. To lessen these requirements, future work may be directed to data-driven system identification, where soil moisture measurements throughout the space and knowledge of fluid movement can be used to build the model over time.

In optimization, we wish for our schedules to guide soil moisture into healthy levels, i.e., above θ_{pwp}. Although we use a value of θ_{pwp} as physically measured and reported in plant physiology literature, in practice it may be better to tune this value by hand. Although θ_{pwp} may have been correct in the original model described in Section 4, it will not be correct in the model we optimize due to the approximations we introduce in Section 4.4. The method of tuning this parameter and the effect of this choice is left for future work.

To correct for PDE model error, it is possible to perform intermediate schedule optimization using fresher soil moisture data in the middle of the irrigation period. However, with the CVX optimization library used during our experiments, the 40mins optimization time would be substantially disruptive in the middle of irrigation, as it would cause a long pause for computation in irrigation that the optimization did not find necessary. Furthermore, due to system pressure limitations, groundskeepers often assign a maximum irrigation time to allow different areas to be scheduled in series, in our system chosen to be three hours. However, with the more recent port to the Julia programming language, subsequent optimizations can be performed in around 1s, making this a very reasonable feature for future implementations of the DICTUM system. By reducing the persistent model error through this re-correction, the DICTUM system will be able to more effectively meet its primary goal of maintaining adequate moisture levels while doing its best to minimize water consumption. Depending on the nature of the model error, it is possible that this re-correction may cause a penalty to the system efficiency to further improve quality of service, but as the DICTUM system already has substantially improved quality of service, we expect any penalty to be slight. As further experiments would be required to quantify these potential improvements by adding this feature, we leave it to future work.

As initial conditions of soil moisture are collected at only a finite number of sensing locations and upsampled by interpolation to the spatial granularity of the PDE grid, it is possible that conditions between the sensed locations may deviate from those expected by the model. Unfortunately, this can be caused by several anomalous conditions such as differing soil depth, soil type, and faulty or incorrectly installed sprinklers, which will cause model disagreement to the true conditions in the space and may be difficult to notice and correct in the model. However, there are several emerging aerial imaging technologies [10, 20] that allow the plant health to be monitored at a significantly higher spatial resolution than our distribution of soil moisture sensors. While they do not allow the soil moisture to be directly measured, necessary for short-term control decisions during irrigation, they will allow the control system to identify regions of the space that have unhealthy plants. If this plant health data shows regions between sensors that are experiencing poor plant health, it can be used to expose a local anomaly, which can then be incorporated into the PDE model for improved control or trigger an alert for local groundskeepers to search for a physical system fault. This will allow our system to provide quality guarantees at all locations throughout the space.

Due to the computational requirements of optimizing the schedule, the model only considers moisture movement that occurs *during* irrigation and as such does not include long-term effects such as leaching and weather that cause the fluid to be lost throughout the rest of the day. This is limiting, as it forces us to assume that all locations in the space lose water in the same way, which may not be true in practice. Future work is necessary to consider these losses across the full 24-hour cycle and include this information when finding irrigation schedules to ensure satisfactory water levels at all times.

10 CONCLUSIONS

Fresh water is a delicate resource, and we must find ways to use it sustainably. The largest irrigated crop by surface area in North America, turf has a demand for an estimated 9B gallons each day. Due to current shortages, there is strong social, environmental, and monetary incentive to shrink this enormous consumer. In this work, we seek to improve the efficiency of turf irrigation systems by analyzing heterogeneous water needs across a span of turf. To this end, we develop a computationally light moisture movement model to be used as constraints in an optimization problem, which is then used to produce optimal valve scheduling within an irrigation system to minimize water consumption while maintaining healthy levels of moisture everywhere. To test its effectiveness, we produce the DICTUM sprinkler node, with the ability to actuate, sense local soil conditions, and communicate wirelessly with sister nodes in the network. Through two separate deployments spanning a total of four weeks, we find that the DICTUM system can reduce system water consumption by 23.4% over our campus' control strategy, and by 12.3% over a state-of-the-art evapotranspiration system. Despite this reduced water usage, DICTUM was also found to reduce turf exposure to unhealthy levels of moisture by a factor of 3.5 over the campus' control, and a factor of 4.08 over the evapotranspiration control. The DICTUM system is expected to return its investment in 16–18 months based on water savings alone.

APPENDICES

A GRASS AS POROUS MEDIA

Darcy's law [22] is applicable to systems where the drag due to the multiple obstructions is the dominant fluid force. To determine when flow through turf may be modeled as flow through a porous medium, we estimate and compare the drag, viscous force, and inertial force per unit volume. Denoting a typical flow velocity scale as U, a typical grass blade size as a, viscosity as μ, density as ρ, the thickness of the liquid layer as h, and the porosity as ϕ, we estimate the drag per volume, based on the low Reynolds number drag of an elongated obstruction of length h and lateral size a, as

$$D \sim \frac{4\pi\mu U h(1-\phi)}{ha^2} = \frac{4\pi\mu U(1-\phi)}{a^2}. \tag{11}$$

Within the liquid, the viscous force per volume is estimated from the viscous term in the Navier-Stokes equations as $F_v \sim \mu U/h^2$. Finally, the inertial force per volume is estimated, also from the Navier-Stokes equations, as $F_i \sim \rho U^2/h$.

As typical values for water and grass blades, we use $\rho = 1\text{g/cm}^3$, $\mu = 0.01\text{g/cm s}$, $a = 0.1\text{cm}$, $h = 1\text{cm}$, and $\phi = 0.9$, and find $\frac{F_v}{D} \sim \frac{a^2}{4\pi h^2}\frac{1}{1-\phi} \sim 1/100$, showing that the drag is dominant over viscous forces for any velocity of the flow. The ratio of inertial forces to drag is $\frac{F_i}{D} \sim \frac{a^2\rho U}{4\pi\mu h}\frac{1}{1-\phi} \sim U$. The applicability of our model is therefore limited to systems where the water flows at velocities $U < 1\text{cm/s}$, which corresponds to most intermittent irrigation regimes.

B DISCRETIZED MODEL FORMULATION

The following six equations hold at each spatiotemporal cell (i, j, t) and represent the discretized, linearized flow motion of Equations (6)–(9) over the variables u and v (velocity of water in soil and surface, respectively, both horizontal components x and y for each), h (surface fluid height), and theta (volumetric moisture content). They also represent the equality constraints in our optimization problem.

$$
\begin{aligned}
\frac{h_{i,j,t+1} - h_{i,j,t}}{\Delta t} &= -\frac{1}{2\Delta x}\Big((\hat{h}v_0^x + h_0\hat{v}^x)_{i+1,j,t} - (\hat{h}v_0^x + h_0\hat{v}^x)_{i-1,j,t} \\
&\quad + (\hat{h}v_0^y + h_0\hat{v}^y)_{i,j+1,t} - (\hat{h}v_0^y + h_0\hat{v}^y)_{i,j-1,t}\Big) + F_s \\
&\quad - \eta\Big(h_0K(\theta_0) + h_0K'(\theta_0)\hat{\theta} + \hat{h}K(\theta_0) + \hat{h}K'(\theta_0)\hat{\theta}\Big), \qquad (12)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\theta_{i,j,t+1} - \theta_{i,j,t}}{\Delta t} &= -\frac{1}{2\Delta x}\Big((\hat{\theta}u_0^x + \theta_0\hat{u}^x)_{i+1,j,t} - (\hat{\theta}u_0^x + \theta_0\hat{u}^x)_{i-1,j,t} \\
&\quad + (\hat{\theta}u_0^y + \theta_0\hat{u}^y)_{i,j+1,t} - (\hat{\theta}u_0^y + \theta_0\hat{u}^y)_{i,j-1,t}\Big) \\
&\quad + \zeta\Big(h_0K(\theta_0) + h_0K'(\theta_0)\hat{\theta} + \hat{h}K(\theta_0) + \hat{h}K'(\theta_0)\hat{\theta}\Big), \qquad (13)
\end{aligned}
$$

$$
\begin{aligned}
\hat{u}^x_{i,j,t} &= -\frac{K(\theta_{0\,i,j,t})}{2\Delta x}(\hat{h}_{i+1,j,t} - \hat{h}_{i-1,j,t} + h_{0\,i+1,j,t} - h_{0\,i-1,j,t}) \\
&\quad - K'(\theta_{0\,i,j,t})\hat{\theta}_{i,j,t}(h_{0\,i+1,j,t} - h_{0\,i-1,j,t}) + \frac{K(\theta_{i,j,t})\vec{\tau}^x}{\rho g} \\
&\quad - \frac{\varphi(\theta_{0\,i,j,t})}{2\Delta x}(\theta_{0\,i+1,j,t} - \theta_{0\,i-1,j,t} + \hat{\theta}_{i+1,j,t} - \hat{\theta}_{i-1,j,t}) \\
&\quad - \varphi'(\theta_{0\,i,j,t})(\theta_{0\,i+1,j,t} - \theta_{0\,i-1,j,t})\hat{\theta}_{i,j,t} - u^x_{0\ i,j,t}, \qquad (14)
\end{aligned}
$$

$$
\begin{aligned}
\hat{u}^y_{i,j,t} &= -\frac{K(\theta_{0\,i,j,t})}{2\Delta x}(\hat{h}_{i,j+1,t} - \hat{h}_{i,j-1,t} + h_{0\,i,j+1,t} - h_{0\,i,j-1,t}) \\
&\quad - K'(\theta_{0\,i,j,t})\hat{\theta}_{i,j,t}(h_{0\,i,j+1,t} - h_{0\,i,j-1,t}) + \frac{K(\theta_{i,j,t})\vec{\tau}^y}{\rho g} \\
&\quad - \frac{\varphi(\theta_{0\,i,j,t})}{2\Delta x}(\theta_{0\,i,j+1,t} - \theta_{0\,i,j-1,t} + \hat{\theta}_{i,j+1,t} - \hat{\theta}_{i,j-1,t}) \\
&\quad - \varphi'(\theta_{0\,i,j,t})(\theta_{0\,i,j+1,t} - \theta_{0\,i,j-1,t})\hat{\theta}_{i,j,t} - u^y_{0\ i,j,t}, \qquad (15)
\end{aligned}
$$

$$
\hat{v}^x_{i,j,t} = -\alpha_h\left(\frac{h_{i+1,j,t} - h_{i-1,j,t}}{2\Delta x}\right) + \frac{\kappa_g}{\eta}\vec{\tau}^x - v^x_{0\ i,j,t}, \qquad (16)
$$

$$
\hat{v}^y_{i,j,t} = -\alpha_h\left(\frac{h_{i,j+1,t} - h_{i,j-1,t}}{2\Delta x}\right) + \frac{\kappa_g}{\eta}\vec{\tau}^y - v^y_{0\ i,j,t}. \qquad (17)
$$

REFERENCES

[1] [n.d.]. Comsol subsurface flow module. Retrieved from: http://www.comsol.com/subsurface-flow-module.
[2] [n.d.]. Decagon devices. Retrieved from: http://www.decagon.com/products/soils/.
[3] [n.d.]. Farming irrigation price increases. Retrieved from: http://www.bloomberg.com/news/articles/2014-07-24/california-water-prices-soar-for-farmers-as-drought-grows.
[4] [n.d.]. Freshwater crisis. Retrieved from: http://environment.nationalgeographic.com/environment/ freshwater/freshwater-crisis/.
[5] [n.d.]. Hunter MP rotator. Retrieved from: http://www.hunterindustries.com/irrigation-product/nozzles/ mp-rotator.
[6] [n.d.]. Hunter rain-clik rain detection. Retrieved from: http://www.hunterindustries.com/irrigation-product/sensors/rain-clikr.
[7] [n.d.]. Hydrus liquid simulator. Retrieved from: http://www.pc-progress.com.
[8] [n.d.]. Looking for lawns. Retrieved from: http://earthobservatory.nasa.gov/Features/Lawn/printall.php.
[9] [n.d.]. Permanent wilting point. Retrieved from: http://nrcca.cals.cornell.edu/soil/CA2/CA0212.1-3.php.
[10] [n.d.]. Photonics—Plant health imaging filters. Retrieved from: https://www.photonics.com/Product.aspx?PRID=60994.
[11] [n.d.]. Price of water is too low. Retrieved from: http://www.nytimes.com/2014/10/15/business/economy/the-price-of-water-is-too-low.html.
[12] [n.d.]. Sheeva plug DevKit. Retrieved from: https://www.globalscaletechnologies.com/t-sheevaplugs.aspx.
[13] [n.d.]. Soil quality indicators. Retrieved from: https://www.nrcs.usda.gov/Internet/FSE_DOCUMENTS/nrcs142p2_053288.pdf.
[14] [n.d.]. Tmote sky. Retrieved from: http://www.snm.ethz.ch/Projects/TmoteSky.
[15] [n.d.]. UGMO irrigation. Retrieved from: http://www.ugmo.com/.
[16] [n.d.]. US outdoor water use. Retrieved from: www3.epa.gov/watersense/pubs/outdoor.html.
[17] 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.
[18] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. 2017. Julia: A fresh approach to numerical computing. *SIAM Rev.* 59, 1 (2017), 65–98. DOI: https://doi.org/10.1137/141000671
[19] Bernard Cardenas-Lailhacar, Michael D. Dukes, and Grady L. Miller. 2008. Sensor-based automation of irrigation on bermudagrass, during wet weather conditions. *J. Irrig. Drain. Eng.* 134, 2 (2008), 120–128.
[20] Laury Chaerle and Dominique Van Der Straeten. 2001. Seeing is believing: Imaging techniques to monitor plant health. *Biochim. Biophys. Acta (BBA)—Gene Struct. Express.* 1519, 3 (2001), 153–166. DOI: https://doi.org/10.1016/S0167-4781(01)00238-X
[21] Antonio Costa. 2006. Permeability-porosity relationship: A reexamination of the Kozeny-Carman equation based on a fractal pore-space geometry assumption. *Geophys. Res. Lett.* 33, 2 (2006). DOI: https://doi.org/10.1029/2005GL025134
[22] Henry Darcy. 1856. *Les Fontaines Publiques de la Ville de Dijon.* Dalmont, Paris, France..
[23] N. A. Dobbs, K. W. Migliaccio, M. D. Dukes, K. T. Morgan, and Y. C. Li. 2013. Interactive irrigation tool for simulating smart irrigation technologies in lawn turf. *J. Irrig. Drain. Eng.* 139, 9 (2013), 747–754. DOI: 10.1061/(ASCE)IR.1943-4774.0000612
[24] Quentin Gallas, Ryan Holman, Toshikazu Nishida, Bruce Carroll, Mark Sheplak, and Louis Cattafesta. 2003. Lumped element modeling of piezoelectric-driven synthetic jet actuators. *AIAA J.* 41, 2 (2003), 240–247.
[25] M. Th. Van Genuchten. 1980. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Sci. Soc. Amer. J.* 44, 5 (1980), 892–898.
[26] Michael Grant, Stephen Boyd, and Yinyu Ye. 2008. CVX: Matlab Software for Disciplined Convex Programming. https://www.researchgate.net/publication/229058225_CVX_MATLAB_software_for_disciplined_convex_programming.
[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. American Society of Civil Engineers.
[29] Eugenia Kalnay. 2003. *Atmospheric Modeling, Data Assimilation, and Predictability.* Cambridge University Press.
[30] M. B. Kirkham. 2004. *Principles of Soil and Plant Water Relations.* Elsevier Science. Retrieved from: https://books.google.com/books?id=NUcoZNPMQEEC.
[31] S. B. McCreath and R. Delgoda. 2017. *Pharmacognosy: Fundamentals, Applications and Strategies.* Elsevier Science. Retrieved from: https://books.google.com/books?id=qEl8DAAAQBAJ.
[32] Enrique Playán, Raquel Salvador, Cristina López, Sergio Lecina, Farida Dechmi, and Nery Zapata. 2013. Solid-set sprinkler irrigation controllers driven by simulation models: Opportunities and bottlenecks. *J. Irrig. Drain. Eng.* 140, 1 (2013), 04013001.

[33] Rolf H. Reichle, Dennis B. McLaughlin, and Dara Entekhabi. 2002. Hydrologic data assimilation with the ensemble Kalman filter. *Month. Weath. Rev.* 130, 1 (2002), 103–114.

[34] E. Rodriguez, F. Giacomelli, and A. Vazquez. 2004. Permeability-porosity relationship in RTM for different fiberglass and natural reinforcements. *J. Comp. Mat.* 38, 3 (2004), 259–268.

[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] K. H. Wesseling, J. R. C. Jansen, J. J. Settels, J. J. Schreuder et al. 1993. Computation of aortic flow from pressure in humans using a nonlinear, three-element model. *J Appl. Phys.* 74 (1993), 2566–2566.

[37] 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

[38] 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 April 2018; revised December 2018; accepted June 2019